

# Tilgængelighed på Internettet

Af Niels Gamburg

Hovedopgave Maj 2008 - Statsbiblioteket



Vejleder: Niels Müller Nielsen  
Multimediedesigneruddannelsen - Århus Købmandsskole

Rapporten må gerne offentliggøres. Tilgængelighed på Internettet - Accessibility on the Internet

## Forord

Min hovedopgave tager specifikt udgangspunkt i Statsbibliotekets hjemmeside <http://www.statsbiblioteket.dk> for at beskrive tilgængelighed på Internettet. Min JavaScript uafhængige kurv kan ses og testes på <http://developer.statsbiblioteket.dk/jsfree>

De tilhørende filer ligger på Cd'en til nærmere gennemsyn og en oversigt over CD'ens filer findes i bilag 3. Den JavaScript uafhængige kurv kan ikke køres lokalt uden for statsbiblioteket, da kurven bygger videre på en lang række webservices, som kun er åbne på Statsbibliotekets interne net. Det drejer sig f.eks. om søgninger i bibliotekssystemer og brugerdata-baser.

## Abstract

*Rapporten behandler tilgængelighed på Internettet for brugere med funktionsnedsættelse. Baggrunden er en praktik på Statsbiblioteket og en række nye krav for offentlige virksomheders hjemmesiders tilgængelighed.*

*En væsentlig del af tilgængelighedsproblematikkerne kan løses ved at arbejde aktivt med et websites generelle brugervenlighed. Det vil medføre store fordele for mange brugere med funktionsnedsættelse, og endvidere vil det også være til gavn for hele brugergruppen.*

*Den anden store del af tilgængelighedsområdet består af tekniske løsninger. Mange af problemerne er meget konkrete og mulige løsninger er velbeskrevne og oftest lette at gå til. Andre krav er dog meget specialiserede og rammer kun en snæver målgruppe. Her må man som offentlig virksomhed analysere hvilke funktioner, der er grundlæggende, og så udarbejde alternative løsninger.*

*Jeg har konkret lavet en kurvfunktion, så brugere der anvender skærmlæsere også kan bestille og reservere materialer på Statsbibliotekets hjemmeside.*

*Incitamentet til at arbejde med tilgængelighed kan virke forholdsvis lille. Hverken blandt udviklere eller designere er der den store prestige i tilgængelighedsarbejdet. Målgruppen kan overfladisk set virke forholdsvis lille og deres generelle indflydelse i samfundet ubetydelig. Heldigvis kan mange tilgængelighedsproblematikker løses med ganske lille indsats, hvis bare området får lidt mere opmærksomhed. Ny lovgivning på området betyder forhåbentligt at fokus på området vil stige, og ellers er konkurrencer som f.eks. bedst på nettet med til at øge fokus på området.*

# Indhold

<b>FORORD</b> .....	<b>2</b>
<b>ABSTRACT</b> .....	<b>2</b>
<b>1 PROBLEMSTILLING</b> .....	<b>5</b>
1.1 PROBLEMBAGGRUND .....	5
1.2 PROBLEMFORMULERING .....	5
1.3 AFGRÆNSNINGER .....	6
<b>2 STATSBIOTEKET</b> .....	<b>7</b>
2.1 HISTORISK RIDS .....	7
2.2 OPGAVER .....	7
2.3 ORGANISATIONSOPBYGNING .....	8
2.4 ORGANISATIONSKULTUR .....	9
<b>3 TILGÆNGELIGHED</b> .....	<b>10</b>
3.1 HVAD ER TILGÆNGELIGHED? .....	10
3.2 KOMMUNIKATIONS TEORI .....	10
3.3 MÅLGRUPPE .....	11
3.3.1 Læseproblemer.....	11
3.3.2 Hørehæmmede .....	12
3.3.3 Farveblindhed.....	12
3.3.4 Blinde/svagtseende.....	12
3.3.5 Motoriske funktionsnedsættelser.....	12
3.4 PRIORITERING .....	12
3.5 FORPLIGTIGELSER SOM OFFENTLIG VIRKSOMHED .....	13
3.5.1 Åbne standarder.....	13
3.6 W3 CONSORTIUMS RETNINGSLINJER FOR TILGÆNGELIGHED.....	14
3.6.1 Oversigt over W3C's 14 krav om tilgængelighed.....	14
3.6.2 Statsbiblioteket og de 14 krav .....	15
3.7 HVORDAN UNDERSØGER MAN TILGÆNGELIGHED?.....	16
3.7.1 Hjælpeværktøjer til undersøgelse af tilgængelighed.....	17
3.8 UDBUDSVÆRKTØJSKASSEN.....	18
3.9 BEDST PÅ NETTET .....	18
3.9.1 Konventioner.....	19
3.9.2 Digital selvbetjening.....	19
3.9.3 Åbenhed og nytteværdi.....	19
3.9.4 Teknisk tilgængelighed.....	19
3.9.5 Debat om Bedst på Nettet .....	19

<b>4 BRUGERVENLIGHED OG TILGÆNGELIGHED .....</b>	<b>21</b>
4.1 BRUGERVENLIGT LAYOUT .....	21
4.1.1 Tydelige hierarkier.....	21
4.1.2 Konventioner på nettet.....	21
4.1.3 Tydeligt definerede områder.....	22
4.1.4 Minimér støj.....	22
4.2 STATSIBLIOTEKETS LAYOUT .....	23
4.3 BRUGERVENLIGE TEKSTER.....	25
4.3.1 Læselighed.....	25
4.3.2 Læsbarhed.....	26
4.3.3 Læsbarhed og Lixtal .....	26
4.4 STATSIBLIOTEKETS TEKSTER .....	27
4.5 BRUGERTESTS.....	29
<b>5 TEKNOLOGI OG TILGÆNGELIGHED .....</b>	<b>30</b>
5.1 JSP OVERSIGT .....	30
5.2 XML OVERSIGT .....	31
5.3 XLST OVERSIGT .....	32
5.4 SESSIONER .....	32
5.5 KURVFUNKTION UDEN JAVASCRIPT. EN OVERSIGT.....	33
5.6 KODEEKSEMPLER FRA JAVASCRIPT OG AJAX UAFHÆNGIG KURV .....	34
5.6.1 Detektere om brugeren har JavaScript installeret.....	34
5.6.2 Lægge materialer i kurven.....	35
5.6.3 Fjerne materialer fra kurven.....	36
5.6.4 Vise materialer i kurven.....	36
5.6.5 Vise materialets status .....	38
5.6.6 Bestille materialer i kurven.....	39
5.6.7 Manipulere datoer i Java.....	41
5.7 STATUS OVER JAVASCRIPT UAFHÆNGIG KURV .....	42
<b>6 TILGÆNGELIGHED – EN AFSLUTTENDE DISKUSSION.....</b>	<b>43</b>
6.1 Hvorfor kan det være svært at opfylde kravene? .....	43
6.2 Hvordan kan man få mere fokus på tilgængeligheden? .....	43
6.3 Tilgængelighed i private virksomheder.....	44
<b>7 KONKLUSION.....</b>	<b>45</b>
<b>8 LITTERATURLISTE .....</b>	<b>47</b>
8.1 BØGER .....	47
8.2 INTERNET RESSOURCER .....	48
<b>9 BILAG .....</b>	<b>50</b>
BILAG 1. DE 14 TILGÆNGELIGHEDSKRAV I TEORI OG PRAKSIS.....	50
BILAG 2. DESIGNDOKUMENT FOR STATSIBLIOTEKET.DK .....	60
BILAG 3 OVERSIGT OVER FILER PÅ CD'EN .....	64
BILAG 4 UDVIKLINGSVÆRKTØJER .....	64

# 1 Problemstilling

## 1.1 Problembaggrund

Tilgængelighed kan defineres som, at et websted er brugbart for brugere med særlige behov, med særligt udstyr og under særlige forhold.

Tilgængelighed er en del af en række krav stillet til alle statslige institutioner hjemmesider fra januar 2008.

Flere af disse krav gælder kun for nye systemer, og derfor ikke p.t. for Statsbiblioteket. Men Statsbiblioteket er som udgangspunkt interesseret i at servicere sine brugere bedst muligt, og er derfor også interesseret i at have et websted, der er så tilgængeligt som muligt.

Den konkrete problembaggrund er implementeringen af et nyt CMS system, Plone, og udviklingen af en søgemaskine, Summa. To elementer, der sammen udgør ryggraden i Statsbibliotekets hjemmeside.

CMS er valgt ud fra en filosofi, om at køre den "out of the box" for at gøre systemet så fleksibelt som muligt i forhold til opgraderingen, og for at minimere den daglige drift og vedligeholdelse for dermed at kunne frigive flere ressourcer til udviklingsopgaver.

Summa er implementeret i Plone, og udviklingsafdelingen arbejder p.t. på at optimere Summa og frigive en udgave som open source.

Statsbiblioteket har arbejdet meget med brugervenlighed og brugerundersøgelser for at optimere deres brugergrænseflade i forhold til deres brugeres specifikke behov.

## 1.2 Problemformulering

Ovenstående giver en række spørgsmål, der er relevante at belyse:

Er alle tilgængelighedskrav relevante for Statsbiblioteket at indfri? Hvad gør Statsbiblioteket for at indfri kravene?

Er der nogen sammenhæng mellem traditionel brugervenlighed og tilgængelighed? Øger den generelle brugervenlighed hjemmesidens tilgængelighed? Kan der være konflikter mellem brugervenlighed og tilgængelighed.

Hvordan håndterer et moderne CMS system som Plone krav til tilgængelighed? Kan man leve op til kravene om tilgængelighed ved at køre et CMS, som Plone out of the box, uden at skulle tilpasse systemet?

Statsbibliotekets mest benyttede side/service er ubetinget deres søgefelt. Søgefunktionen er bygget på Statsbibliotekets egen søgemaskine "Summa". Gennem Summa kan brugeren søge efter materialer og lægge dem i en kurv til senere bestilling. Kurven er p.t. baseret på brug af JavaScript og AJAX. Hvad betyder det for tilgængeligheden? Kan samme funktionalitet opnås uden brug af JavaScript og AJAX?

Hvad kan man gøre for at styrke incitamentet til at arbejde mere med tilgængelighed? Hvad bliver der gjort fra statens side?

### ***1.3 Afgrænsninger***

På grund af et krav om behandlingsdybde og samtidige begrænsninger i tid og sideantal, vil det være nødvendigt at udvælge og prioritere stoffet.

I forhold til de 4 fagområder på multimediedesigner uddannelsen vil jeg i rapporten lægge hovedvægten på design, kommunikation og programmering.

Tilgængelighed handler basalt set om kommunikation under specielle forudsætninger, og derfor er det naturligt at lægge meget fokus på det kommunikative.

En væsentlig del af tilgængeligheden afhænger af den grafiske brugergrænseflade og designet. Jeg vil beskæftige mig en del med overskuelighed og brugervenlighed, men nøjes med at henvise til hvilke designmæssige redskaber, der er egnede til at skabe overskuelighed, frem for at beskrive, hvordan man bruger dem i praksis.

Programmeringen og udviklingen af webapplikationer er vigtig, når det handler om at gøre tingene tilgængelige. Dels fordi det ofte er nye teknologier, der begrænser adgangen for brugere med funktionsnedsættelser, og dels fordi man ofte skal bruge alternative programmeringsløsninger for at afhjælpe en manglende tilgængelighed.

Jeg vil opdele rapporten i to hovedsektioner som afspejler denne vægtning. I den første sektion vil jeg behandle brugervenligheden og den mere brede vinkel på tilgængelighed. Og den sidste sektion vil jeg koncentrere sig om den specialiserede og tekniske løsning af problematikkerne omkring ét helt specifikt område indenfor tilgængelighed.

Jeg vil tage udgangspunkt i Statsbibliotekets hjemmeside, og derfor koncentrere mig om de problematikker, der er relevante for netop Statsbiblioteket websted. Rapporten er koncentreret om de krav der er til offentlige institutioner på området. Det betyder, at jeg vil gå forholdsvis let hen over andre tilgængelighedsproblematikker, som kan være både interessante og relevante, men som kun har mindre betydning i forhold til statsbiblioteket.dk.

## 2 Statsbiblioteket

### **2.1 Historisk rids**

Statsbiblioteket i Århus startede i 1902. Baggrunden var at nyttiggøre det Kongelige biblioteks 100.000 dubletter af danske bøger. Fra jysk side sås initiativet også som en mulighed for at bryde med hovedstadens dominans og den stærke centralisering, der på det tidspunkt prægede dansk politik.

I starten var samlingerne baseret på ovennævnte dubletter, samt forskellige donerede samlinger fra privatpersoner. Og pligtafleveringsloven fra 1902 medførte en hurtigt en voksende samling.

I slutningen af 1920'erne oprettedes Århus Universitet og fra 1934 fik Statsbiblioteket sine forpligtigelse som universitets bibliotek. Det medførte igen en udvidelse af samlingerne, da man også fik ansvaret for at anskaffe nødvendig udenlandsk litteratur.

I starten er 50'erne nåede samlingen op på 560.000 bind, 30.000 musiknumre og flere specialsamlinger af småtryk. F.eks. gemmes der årlig 600 tidsskifter. Og der begynder at være alvorlige problemer med pladsmangel.

I 1963 blev det 45 meter høje bogtårn bygget. Bygningen er en markant del af universitetsparken, og har siden stået som et af Århus vartegn.

I 1996 startede digitalisering af udvalgte materialer. Dette foregår stadig i stigende omfang og drømmen er at få hele samlingen digitaliseret på et tidspunkt.

I 1998 blev pligtafleveringsloven revideret og begyndte at omfatte visse statiske hjemmeside. Man skulle på det tidspunkt selv anmelde sine hjemmesider til Statsbiblioteket.

I 2005 foretog man igen en større revidering af pligtafleveringsloven, og Statsbiblioteket begyndte at opbevare danske radio og tv-programmer og at gemme betydelige danske hjemmesider.

I dag råder Statsbiblioteket over ca. 4 millioner fysiske materialer, hvoraf bøger og andre skriftlige udgivelser udgør ca.3 millioner. Derudover råder biblioteket over 40.000 elektroniske bøger og 20.000 elektroniske tidsskrifter.

Sammen med den øgede digitalisering er web og IT afdelingerne på Statsbiblioteket vokset eksplosivt. Der arbejdes på forskellige EU projekter samt egne projekter, der sigter på digital formidling af forskelligt materiale.

### **2.2 Opgaver**

Statsbibliotekets opgaver kan overordnet dels ind i tre områder

Som Nationalbibliotek har Statsbiblioteket til opgave at indsamle, bevare og tilgængeliggøre dele af den danske kulturarv for offentligheden. Konkret handler det om alle danske publikationer fra bøger og aviser til danske audiovisuelle medier og dansk internetmateriale.

Som overcentral for folkebibliotekerne opbygger Statsbiblioteket samlinger og formidler lån. Samtidig understøtter de folkebibliotekernes virksomhed ved at udvikle relevante serviceydelser.

Som Universitetsbibliotek forsyner Statsbiblioteket Aarhus Universitet med informations- og dokumentationsmateriale til brug for undervisning og forskning. Pt. er statsbiblioteket i færd med at fusionere med universitetets interne bibliotekssamlinger.

## 2.3 Organisationsopbygning

Overordnet kan Statsbiblioteket organisationsform beskrives som et klassisk fagbureaukrati. Der er ansat mange højtuddannede i flere af niveauerne, og strukturen er forholdsvis flad. Der ses også visse karakteristika fra Adhokratiet, som er en organisationsform, der er specielt egnet til organisationer med komplekse arbejdsopgaver og et dynamisk marked.

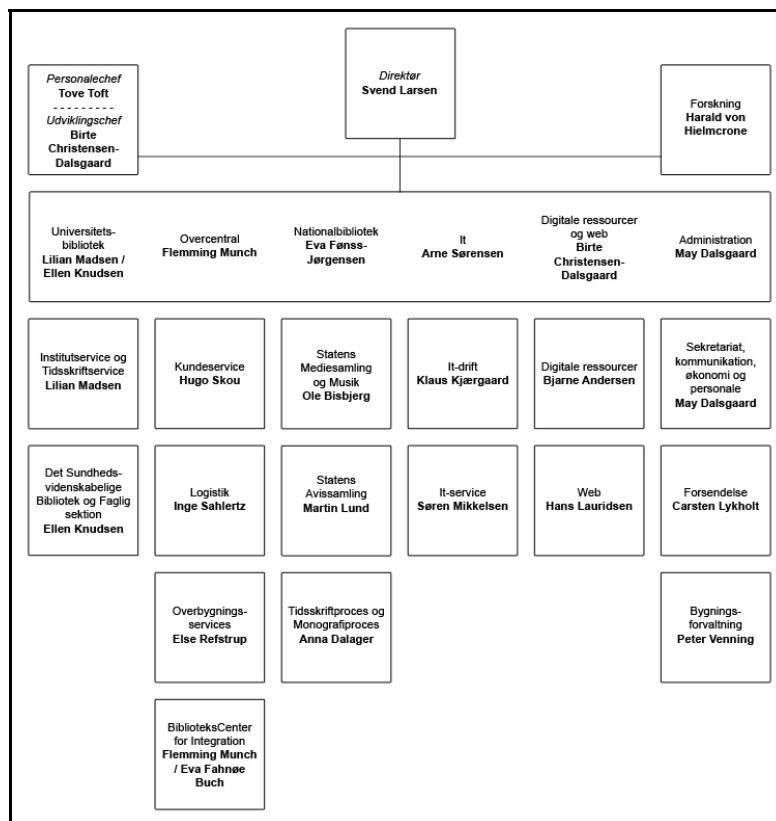
(Christiansen/Kjær/Skriver/Staunstrup, 2005: 105).

Konkret er Statsbiblioteket opbygget med en direktør øverst i hierarkiet. Under ham er der et par støttefunktioner i form af en personalechef, en udviklingschef og en forskningschef.

Direktøren og cheferne under ham danner en ledelsesgruppe, som tager de strategiske beslutninger i organisation, selvfølgelig med direktøren som den ansvarlige.

Laget under direktøren er organiseret som et såkaldt "flydende cheflag". Det indebærer at de forskellige sektionsledere (de nedre kvadrater på organisationsdiagrammet), ikke har en, men flere chefer. I den klassiske organisationsteori vil man kalde en sådan konstruktion for horisontal arbejdsdeling, hvor opgaverne i organisationen er opdelt efter objektprincippet (Christiansen/Kjær/Skriver/Staunstrup, 2005: 58). Et eksempel ses i f.eks. de 4 IT afdelinger, der er opdelt efter specifikke arbejdsområder frem for medarbejdernes IT kompetencer.

Det flydende cheflag giver mulighed for et stærkt samarbejde på tværs af organisation, hvor hver enkelt afdeling bedre kan følge med i, hvad der foregår i de andre afdelinger. Ulempen er at det nogle gange medfører svært gennemskuelige hierarkier og kommandoveje, hvilket kan medføre frustrationer på både sektionsleder og medarbejderniveau.



Organisationsdiagram for Statsbiblioteket



## **2.4 Organisationskultur**

Statsbiblioteket er en organisation med mange faggrupper på arbejdspladsen, spændende fra bygningspersonalet og studentermedhjælpere, over kontorpersonale og bibliotekarer til akademikere og forskere.

Det medfører selvfølgelig en stor diversitet i kulturerne i de forskellige afdelinger. Mine erfaringer stammer fra afdelingerne for Web og Digitale ressourcer. Afdelingerne er typiske udviklingsafdelinger, hvor fokus er udvikling frem for drift og vedligeholdelse.

Kulturen i Web og Digitale ressourcer er hovedsageligt en projektorienteret kultur, hvor forholdet til lederne er kendetegnet ved lighed og ledelsen er orienteret mod opgaverne

(Christiansen/Kjær/Skriver/Staunstrup, 2005: 200).

Dette passer godt sammen med at Digitale ressourcer og webafdelingen i store træk fungerer som en typisk projektorganisation. Der er en høj grad af decentralisering og tværgående kommunikation, og mange af medarbejderne er involveret i flere forskellige projekter, hvor gruppen er sammensat til at udføre specifikke opgaver (Ad Hoc, til dette formål)

(Christiansen/Kjær/Skriver/Staunstrup, 2005: 80).

Digitale ressourcer og webafdelingen arbejder ud fra en filosofi om, at medarbejderne i stor udstrækning er med til selv at definere deres arbejdsopgaver. Der er selvfølgelig overordnede arbejdsopgaver, der skal løftes, men ellers lægges der vægt på, at medarbejdernes idéer er en væsentlig del af drivkraften i udviklingsarbejdet. En sådan personalepolitik medfører et stort engagement i det daglige arbejde og underbygger også det innovative.

Et andet eksempel på en understøttelse af det innovative element, ses også i et begreb, der kaldes "Google dage". Det er dage, hvor de daglige projekter får lov til hvile, og medarbejderne i stedet kaster sig over projekter, de selv definerer. Eneste betingelse er at projekterne skal have en vis relevans for Statsbiblioteket, enten her og nu eller måske i fremtiden. Google dagene ligger fast om fredagen, men kan dog aflyses, hvis der opstår akutte problemer eller der er vigtige deadlines, der skal overholdes.

Set ud fra en teoretisk synsvinkel skaber initiativer som Google dage gode muligheder for innovative tankegang og kreativitet. Google dagene er præget af en eksperimenterende og legende tilgang til tingene. Det giver IT-udviklerne både tid, rum og formålsløshed, der alle er faktorer, der er med til at understøtte den kreative og innovative proces (Wolf-Toft/Poulsen, 2007: 85).

## 3 Tilgængelighed

### 3.1 Hvad er tilgængelighed?

IT og Telestyrelsen har tidligere brugt følgende korte og præcise definition på tilgængelighed på Internettet:

*"Tilgængelighed på Internettet er et spørgsmål om at give flest mulige mennesker i flest mulige situationer og fra flest mulige browserteknologier adgang til webbaseret information."*

Man bruger ikke denne definition mere. Måske fordi den åbner op for at tilgængelighed også dækker adgang fra andre browserteknologier. Dette kan være relevant, hvis der er tale om browserteknologier, der skal afhjælpe funktionsnedsættelser, men måske mindre relevant, hvis det er understøttelse af browsere i mobile enheder som f.eks. mobiltelefoner etc. Adgang fra mobiltelefoner kan også være yderst relevant for at udvikle servicen i den offentlige forvaltning, men i denne forbindelse kan en sådan formulering måske tage fokus fra den tilgængelighed, der vedrører brugere med funktionsnedsættelser.

En anden definition på tilgængelighed gives af Sussie Christensen (Christensen, 2003:11):

*"Tilgængelighed kan defineres som, at et websted er brugbart for brugere med særlige behov, med særligt udstyr og under særlige forhold."*

Tilgængelighed handler altså om at sikre, at alle brugere uanset eventuelle funktionsnedsættelser har adgang til samme informationer og offentlig service på Internettet. Tilgængeligheden på offentlige hjemmesider, bliver vigtigere efterhånden som digitaliseringen tager til, og flere og flere offentlige institutioner tilbyder forskellige former for selvbetjening på nettet.

Den første tanke man ofte får, når talen falder på tilgængelighed er brugere med massive problemer i forhold til brugen af Internettet, som f.eks. blinde. Men flertallet af brugere med funktionsnedsættelse findes i helt andre og mindre alvorlige kategorier, hvor de tilsvarende løsninger ofte vil være langt mindre gennemgribende. Det kan f.eks. være brugere, som har kognitive funktionsnedsættelser, nedsat syn eller ordblindhed.

Anbefalingerne til tilgængelighed udvikler sig lige så hurtigt som Internettet. Så det der var gode regler for få år siden er det ikke nødvendigvis i dag. F.eks. anbefalede man for få år siden ikke at anvende tabeller til layout af websteder, fordi skærmlæserne ikke forstod disse layout tabeller. Moderne skærmlæsere kan derimod godt håndtere simple layout tabeller. At man så af andre grunde anbefaler at undgå tabelbaserede layouts, er en helt anden diskussion.

### 3.2 Kommunikations teori

For at anskueliggøre problematikkerne omkring tilgængelighed er det oplagt at kigge på den klassiske kommunikationsmodel. I denne simple model arbejder man med 3 begreber. Afsender, medie og modtager.

Afsender -> Medie -> Modtager

Modellen foreskriver at enhver form for kommunikation har en afsender og en modtager, og at kommunikationen mellem afsender og modtager foregår i et medie.

Afsenderen er den der vil formidle et budskab eller nogle informationer. I vores tilfælde er det Statsbiblioteket, der er afsenderen. Budskabet eller informationerne er oplysninger omkring deres materialesamlinger.

Modtagerne er Statsbibliotekets brugere, og når vi snakker tilgængelighed er målgruppen defineret som brugere, der har en funktionsnedsættelse.

Mediet er den kanal som budskabet bliver overført igennem. Det kan f.eks. være aviser, radio, telefon eller som i dette tilfælde en hjemmeside.

Mediet vil ofte tilføre budskabet en form for støj. I f.eks. en radio kan begrebet være helt bogstaveligt. Det kan være decideret støj som følge af dårlige modtagerforhold. I avisen kan det være typografien eller kvaliteten af trykket, der afgør hvor meget støj der er. På en hjemmeside, kan det være for lille tekst eller urolige baggrunde, der gør indholdet vanskeligt at læse. Støj gør betingelserne for kommunikationen dårligere, idet budskabet risikerer at drukne i støjen.

Når vi snakker om brugere med funktionsnedsættelse, risikerer man, at støjen bliver så høj at budskabet ikke bliver forstået eller måske slet ikke når frem til modtageren.

I modsætning til den klassiske kommunikationsteori, der overvejende beskæftigede sig med massekommunikation, der kun går fra modtager til afsender, vil kommunikationen i moderne medier, som på et website ofte gå begge veje. Brugere kan kommunikere tilbage til afsenderen. De kan f.eks. skrive mails, oprette indlæg i diskussionsfora eller bestille varer. Dette forhold illustreres ved at lade pilene i modellen pege begge veje.

Afsender <-> Medie <-> Modtager

Denne udvidede model er meget vigtig efterhånden som flere og flere offentlige funktioner rykker ud på Internettet. Så hvis borgere med funktionsnedsættelse ikke bare skal have adgang til samme informationer, men også have adgang til samme grad af selvbetjening på Internettet, er det vigtigt, at minimere støjen på mediet begge veje i kommunikationen.

### **3.3 Målgruppe**

Målgruppen for tilgængelighed er personer, der har en eller anden form for nedsat funktionsevne på et eller flere områder. For at kunne lave en prioritering er det interessant at se på, hvilke funktionsnedsættelser brugerne har og hvor mange, der lider af hvilke funktionsnedsættelser.

Målgruppen er bred, fordi den ikke er defineret ud fra hvad personerne i målgruppen har til fælles med hinanden. Den er tværtimod defineret i forhold til, at den **adskiller** sig fra en gruppe af gennemsnitlige brugere. Så det brugere med funktionsnedsættelse har til fælles er at adskille sig fra noget andet. Dette gør den samlede målgruppe meget bred, og den indeholder brugere med **vidt** forskellige krav.

Det er ikke muligt at sætte fuldstændige tal på det samlede antal personer med fysisk eller psykisk funktionsevnenedsættelse. Dels registreres personer ikke efter en eventuel nedsat funktionsevne, og dels er det ikke muligt at sætte en endegyldig definition for, hvornår en person har en funktionsnedsættelse eller ej. Tilhører man f.eks. gruppen af personer med nedsat funktionsevne hvis man bruger briller, høreapparat eller er dårlig læser?

Men der findes en række overslag over befolkningens funktionsnedsættelser. Så længe man har i baghovedet at tallene ikke er nagelfaste, kan disse estimater være nyttige til at tegne et overordnet billede af situationen.

#### **3.3.1 Læseproblemer**

It- og Telestyrelsen (<http://adgangforalle.dk>) angiver, at der i Danmark er mellem 800.000 og 1,3 mio. voksne personer, der på grund af læseproblemer har store problemer med at tilegne sig viden via tekstbaserede medier. Tallene dækker både mennesker, der har

funktionsnedsættelser f.eks. ordblindhed og personer, der måske på grund af mangelfuld skolegang har problemer med at læse.

### 3.3.2 Hørehæmmede

Man antager, at ca. 500.000 mennesker i Danmark har nedsat hørelse. Af disse bruger ca. 250-300.000 høreapparat. (<http://dagenssundhed.dk/index.php/OErelidelser/Hoerenedsaettelse.html>)

### 3.3.3 Farveblindhed

Der findes en række forskellige former for farveblindhed. Den mest udbredte type er en rød-grøn farveblindhed.

Ca.. 8% af danske mænd og 1% af danske kvinder (ca. 225.000) har en såkaldt afvigende farveopfattelse. (Christensen, 2003: 18)

### 3.3.4 Blinde/svagtseende

Gruppen af blinde er forholdsvis lille, men antallet af svagtseende er stort. Og man antager, at antallet af mennesker med dårligt syn, vil stige i de kommende år. Dels fordi vi bliver ældre og ældre, dels på grund af en øget udbredelse af diabetes 2, hvor synsproblemer er en følgesygdom ([http://www.dkblind.dk/om\\_os/udgivelser/blindesagen/bls06/bls6-06/synstolkning-er-ikke-kun-for-blinde](http://www.dkblind.dk/om_os/udgivelser/blindesagen/bls06/bls6-06/synstolkning-er-ikke-kun-for-blinde)).

### 3.3.5 Motoriske funktionsnedsættelser

Begrebet dækker over alt fra spastisk lammelse til museskader. Ligesom ved de blinde er de alvorlige funktionsnedsættelser forholdsvis ualmindelig, hvor f.eks. museskader er et stadigt stigende problem. (<http://www.erhvervsbladet.dk/article/20061129/news01/111290235/>) .

## 3.4 Prioritering

Prioritering af tilgængelighedsarbejdet kan ikke undgås. Der skal tages stilling til hvor langt man vil gå for at øge tilgængeligheden. For selv om der findes konkrete regler for det meste, er der også vide fortolkningsmuligheder af mange af retningslinjerne. Hvornår er f.eks. et billedes tesktliggørelse beskrivende nok, og hvor stort skal et link være, før det er let at ramme for en bruger med motoriske vanskeligheder.

Det er også vigtigt at prioritere de ressourcer man har til rådighed, så de bliver brugt mest hensigtsmæssigt. Det vil sige der, hvor man får mest tilgængelighed for pengene.

Det kan muligvis opfattes som kontroversielt at diskutere prioritering i forbindelse med tilgængelighed. Men ud fra en generel filosofi om, at det **ikke** at træffe et valg, også er at træffe et valg, kan man ikke undgå at prioritere.

Det vigtige, når man skal lave sine prioriteringer, er at man ikke sætter prioritering lig med fravælgelse. Men i stedet arbejder ud fra et princip om at opnå, den **nødvendige** funktionalitet, og i forståelse af, at det kan være økonomisk uopnåeligt at opnå fuld funktionalitet. Det må så bygge på konkrete skøn, hvad den nødvendige funktionalitet er.

Ud fra tallene tegner der sig klart billede. Brugervenlighed med fokus på læselighed vil være det der gavner flest mulige. Ikke alene gavner det de mange svage læsere i Danmark, men er gavnligt for langt de fleste brugere. Et område som farveblindhed vil det også være naturligt at prioritere højt. Igen er der tale om en stor gruppe mennesker som bliver hjulpet, og problemerne omkring farveblindhed er både simple og billige at løse eller forebygge. Mest af alt handler det om, at webdesigneren er opmærksom på problematikken.

Ifølge en stor undersøgelse om tilgængelighed foretaget af Forrester Research i 2003 for Microsoft (<http://www.microsoft.com/enable/research/phase1.aspx>), vil der blandt amerikanske computerbrugere mellem 18 og 64 år være 57 % som ville have gavn af overholdelse af regler for tilgængelighed. I samme undersøgelse skønner man at ca. 20 % af den amerikanske befolkning har en decideret funktionsnedsættelse.

Man kan med rimelighed også argumentere for at en høj grad af generel brugervenlighed også vil gavne brugere med funktionsnedsættelser. Der er ingen grund til at tro, at brugere med funktionsnedsættelser skulle blive mindre forvirrede end "almindelige" brugere af uoverskuelige og rodede brugergrænseflader. For visse typer af funktionsnedsættelser kan man endog antage, at det modsatte er tilfældet. Nemlig at brugere med kognitive funktionsnedsættelser, vil have endnu større problemer med at manøvrere og overskue en kaotisk webside, og dermed stor gavn af en generel brugervenlighed, der skaber overblik over hjemmesiden.

Opprioritering af nogle områder, medfører så naturligvis nedprioritering af andre. I dette tilfælde tegner der sig et billede af at brugerne af skærmlæsere er den mindste gruppe. Her er det vigtigt igen at understrege, at løsninger **ikke** skal fravælges, men at man bliver nødt til at lave en analyse af grundfunktionaliteten, og ud fra denne analyse skal man så lave sine endelige løsninger.

Jeg vil i resten af rapporten arbejde ud fra disse overordnede prioriteringer med eksempler fra begge sider i spekteret, altså på den ene side den brede almene løsning, der gavner flertallet af brugere og på den anden side den højt specialiserede løsning, som tilbyder nødvendig funktionalitet til en snæver målgruppe.

## ***3.5 Forpligtigelser som offentlig virksomhed***

### **3.5.1 Åbne standarder**

Fra den 1. januar 2008 er en række åbne standarder blevet obligatoriske at anvende for alle offentlige myndigheder. Det betyder, at alle nye it-løsninger inden for det offentlige skal kunne anvende og overholde disse obligatoriske, åbne standarder.

De syv sæt af obligatoriske, åbne standarder træder i kraft fra den 1. januar 2008.

Standarderne omfatter blandt andet:

- Standarder for dataudveksling mellem offentlige myndigheder (OIOXML)
- Standarder til elektronisk sags- og dokumenthåndtering (FESD)
- Standarder til elektroniske indkøb i det offentlige (OIOUBL)
- Standarder for digital signatur (OCES)
- **Standarder for offentlige netsteder / hjemmesider og tilgængelighed**
- Standarder for it-sikkerhed (DS484 - kun for staten)
- Standarder for dokumentudveksling (ODF/OOXML)

Kravet om anvendelse af de åbne standarder gælder kun for nye it-løsninger

Fokus i aftalen synes at være på dokumentudveksling, da det er det eneste område, der selvstændigt bliver diskuteret i aftalen. Det hænger sandsynligvis sammen med, at det også har afstedkommet store diskussioner i offentligheden om det konkrete valg af standarder. Tilgængelighed bliver kun nævnt som et punkt og ikke diskuteret yderligere i dokumentet.

Ud over de lovmæssige forpligtigelser mener jeg også, at det offentlige har en række moralske forpligtigelser overfor alle borgere uanset funktionsniveau. Det er en del af velfærdssamfundets filosofi, at vi også tager hånd om svagere medborgere. Disse moralske forpligtigelser er selvfølgelig svære at lave tilgængelighedsarbejde ud fra, da grænserne er flydende og den

samfundsmæssige udvikling afgørende. Desuden har det politiske klima en afgørende indflydelse på, hvordan begreber som velfærd, tilgængelighed og service bliver defineret i praksis. En spændende diskussion, men den ligger udenfor denne opgaves rammer.

### **3.6 W3 Consotiums retningslinjer for tilgængelighed**

World Wide Web Consotiums (W3C) er den organisation, der har stået med ansvaret for at udvikle standarder og specifikationer for world wide web. W3C har også udviklet et sæt retningslinjer for tilgængelighed på Internettet. Reglerne kaldes Web Content Accessibility Guidelines (WCAG), og er det nærmeste man kommer på en international standard for tilgængelighed. Retningslinjerne er omfattende og grundige og kan findes i deres originale udformning her på W3C's hjemmeside (<http://www.w3.org/TR/WAI-WEBCONTENT/>).

W3C's retningslinjer som er blevet bearbejdet til danske forhold af IT og Telestyrelsen (<http://www.itst.dk/kommunikation-og-tilgaengelighed/tilgengelighed/tilgengelighed-i-praksis/vejledning-wcag>), er at regne for de officielle danske retningslinjer, der skal efterleves jf. de obligatoriske åbne standarder, og retningslinjerne kaldes i den danske oversættelse for krav.

#### **3.6.1 Oversigt over W3C's 14 krav om tilgængelighed**

Reglerne er opdelt i 14 overordnede områder.

- 9 af dem omhandler tekniske aspekter omkring websider (krav 1 og krav 3-11)  
Kravene er meget konkrete og omhandler emner som, validering, flash, multimedier, tabeller, frames etc. Kravene er entydige, og det er derfor let at afgøre om man overholder det enkelte krav.
- 1 omhandler farver/grafisk design (krav 2).  
Det handler om farveblindhed, og ikke-grafiske skærmteknologier. Kravene er entydige og med de rigtige værktøjer, er det let at afgøre, om man lever op til kravene.
- De sidste 3 krav handler om brugervenlighed i form navigation, overblik og læsevenligt sprog (krav 12-14)  
Kravene om brugervenlighed er i vid udstrækning åbne for fortolkninger, og jeg vil gøre en del ud af at beskrive brugervenlighed i forhold til tilgængelighed senere i rapporten.

### De 14 krav til Tilgængelighed

1. Billede og lyd  
Alternativ tekster til visuelt og auditivt indhold
2. Farver  
Kan siden bruges uafhængigt af farvevalg og farvesyn
3. Opmærk korrekt  
Valid html-kode. Style sheets. Relative størrelser. Overskrifter. Lister. Citater.
4. Angiv benyttet sprog.  
Hvilket sprog er hjemmesidens hovedsprog
5. Tabeller.  
Data-tabeller. Layout tabeller.
6. Nye teknologier  
Teknologi neutral. Uden StyleSheets. Alternativ til dynamisk indhold. AJAX.
7. Tidsafhængige ændringer  
Blink. Bevægelse.
8. Indlejrede objekter  
Programmerede elementer.
9. Hardware uafhængighed  
Image-maps. Enhedsuafhængighed. Logisk script.
10. Midlertidige løsninger  
Pop-up. Label. Fieldset og legend.
11. W3C teknologier  
Anvend W3C teknologier. Alternativ side.
12. Kontekst og orientering  
Frames. Overblik
13. Navigation  
Links. Metadata. Sitemap. Konsistens.
14. Forståeligt og enkelt  
Klart sprog

### 3.6.2 Statsbiblioteket og de 14 krav

Jeg har undersøgt om statsbiblioteket.dk opfylder de 14 krav til tilgængelighed. Her følger en kort og koncentreret gennemgang. Noterne fra den fuldstændige gennemgang kan ses i bilag 1.

Overordnet lever statsbiblioteket op til langt de fleste krav. Dels i kraft af at en lang række af kravene ikke er aktuelle, da hjemmesiden hovedsageligt er tekstbaseret og ikke indeholder betydende billeder eller multimediepræsentationer. En anden del af kravene bliver opfyldt fordi Plone tilbyder en del af de nødvendige teknologier som f.eks. acces-taster til skærmlæsere og præ-udfyldte alt-beskrivelser på billeder.

Endelig har man på statsbiblioteket arbejdet meget med brugerinvolvering og har foretaget mange forskellige former for brugertest og workshops inden for de sidste par år. Dette afspejler sig i brugergrænsefladen, som er rippet for overflødige funktioner og forstyrrende elementer. I stedet er fokus på materialesøgning, som flertallet af brugere har peget på, som den altoverskyggende vigtigste funktion på hjemmesiden. Dette forhold, samt en generel utilfredshed med langsommeligheden og den manglende fleksibilitet i det eksisterende bibliotekssøgesystem, medførte at statsbibliotekets udviklingsafdeling kastede sig ud i at udvikle egen søgemaskine.

Alle brugerundersøgelserne er suppleret af løbende studier af logfiler, der har peget på det samme resultat. Brugere besøger hjemmesiden for at søge på bibliotekets materialer og ikke ret meget andet. Det har medført at søgefeltet har fået en fremtrædende plads, og er det vigtigste element på hjemmesiden.

Derudover har man på Statsbiblioteket en tydeligt defineret politik om at benytte sig af Internettets konventioner og undgå overflødig pynt.

Så overordnet set lever Statsbiblioteket op til kravene om brugervenlighed. Men som tidligere nævnt er brugervenlighedskravene mindre konkrete end de teknologiske krav. Derfor vil jeg i afsnittet om "Brugervenlighed og tilgængelighed", tage både regler for brugervenlighed og Statsbibliotekets status i forhold til disse op til en mere præcis gennemgang.

Der er også område, hvor det kniber med at overholde kravene fra W3C. Det drejer sig hovedsageligt om krav 6, der handler om nye teknologier og om at gøre siden teknologi-neutral. Problemerne her koncentrerer sig om Summas brugergrænseflade, som er afhængig af avanceret JavaScript og AJAX til en lang række essentielle funktioner.

AJAX udgør i dag stadig et reelt problem for skærmlæsere. Skærmlæsere bliver ikke oplyst om, hvor på siden der sker en opdatering, derfor kan brugere ikke orientere sig på siden. Der arbejdes fra international side på højtryk med at løse disse problemer, f.eks. ved at forbedre enten skærmlæsere eller AJAX opmærkningen, så AJAX bliver gjort mere tilgængelig. Dette arbejde foregår selvfølgelig, fordi AJAX teknologien generelt er meget brugervenlig, og derfor optræder på flere og flere websites.

Jeg vil behandle en løsning på denne problematik i rapportens sidste sektion "Teknologi og tilgængelighed", hvor jeg vil beskrive programmering af en JavaScript og AJAX uafhængig kurv.

Derudover er der på hjemmesiden et par småproblemer, som kan løses på ganske kort tid. Det handler f.eks. om:

- Manglende alt-tags på de omslagsillustrationer som Summa henter ind (krav 1). Problemet kan løses med et par linjer XSLT, hvor der sættes et standard alt-tag til alle omslagene. F.eks. med teksten "omslag".
- Manglende validering på grund af tilpasninger (krav 3). Problemerne er meget små og skyldes f.eks. et par uafsluttede tags.
- Manglende angivelse af sprog i på hjemmesiden pga. en mindre fejl i Summa (krav 4).
- Enkelte steder glipper det med at bruge sigende linktekster (krav 13). Dette er desværre nok uundgåeligt i en større organisation som Statsbiblioteket, som benytter sig af mange webredaktører.

### ***3.7 Hvordan undersøger man tilgængelighed?***

Når man skal til at teste for tilgængelighed er der ingen lette veje. Man bliver nødt gennemgå tilgængelighedskravene et for et.

Der findes godt nok en række automatiske online tjenester, der tester en hjemmesides tilgængelighed. De fungerer ved, at man taster hjemmesidens adresse ind på servicens side, og et script vil så løbe kildekoden igennem. Men resultaterne er for upræcise til at kunne bruges til andet end at skaffe sig et hurtigt overblik. Det siger sig selv at f.eks. alle brugervenlighedsaspekterne ikke kan efterprøves ved hjælp af et program.

Efter at man har sikret sig, at siden lever op til de formelle krav, er det derefter vigtigt at brugerteste hjemmesiden. Brugertest er altid nødvendige for at afsløre de u hensigtsmæssigheder, der altid vil optræde på trods af at siden overholder kravene.



Læs mere om brugertest i afsnittet om "Brugervenlighed og tilgængelighed".

### 3.7.1 Hjælpeværktøjer til undersøgelse af tilgængelighed

Heldigvis findes der en række forskellige hjælpeværktøjer, som kan gøre arbejdet med både at undersøge et websites tilgængelighed og udvikle nye webløsninger meget lettere. Først og fremmest findes der en række forskellige toolbars til både Internet Explorer og Firefox browsere.

Til Internet Explorer findes der AIS Web Accessibility Toolbar (<http://www.visionaustralia.org.au/ais/toolbar/>). Toolbaren er oversat til dansk og der findes også en omfattende dansk dokumentation. Et alternativ er Web Accessibility Toolbar (<http://www.paciellogroup.com/resources/wat-ie-about.html>).

Til Firefox findes Firefox Accessibility Extension (<https://addons.mozilla.org/da/firefox/addon/5809>). Og hvis man kombinerer den med Firebug (<https://addons.mozilla.org/da/firefox/addon/1843>) og/eller Webdeveloper (<https://addons.mozilla.org/da/firefox/addon/60>), har man et meget kraftigt og gratis webudviklingsværktøj.

Fælles for disse toolbars, er at de indeholder et væld af nyttige funktioner, når en hjemmesides tekniske tilgængelighed skal testes.

Af de mange funktioner kan nævnes:

- HTML validering med et enkelt klik.
- Check af siden ved forskellige skærmstørrelser.
- CSS-funktioner, der viser siden med CSS slået fra, samt overblik over CSS.
- Billedfunktioner, hvor man blandt andet kan tjekke ALT tekster til image-tags.
- Farvefunktioner, hvor man f.eks. kan se websiden i gråtoner eller finde værktøjer til kontrol af kontraster
- Strukturfunktioner, hvor man kan se overskriftsniveauer, tabel, liste og ramme elementer, og få et overblik over DOM modellen.
- Scriptfunktioner, hvor man kan se JavaScript og eventhandlers
- Dokumentinfo, hvor man blandt andet kan se dokumentdeklarationer og hjemmesidens sprog
- Browserindstillinger, hvor man kan slå JavaScript, ActiveX, billeder og CSS til og fra

Der er lidt variationer i antallet og kvaliteten af værktøjerne i de forskellige toolbars, så det endelig valg afhænger dels af personlige præferencer og dels, hvilke tilgængelighedsproblematikker, der er mest relevante i forhold til den side man udvikler eller tester.

Et andet godt værktøj kan man finde hos Vischeck.com (<http://vischeck.com/vischeck/vischeckURL.php>). Her kan man få en idé om, hvordan webstedet ser ud med en farveblinds øjne. Vischeck.com simulerer tre forskellige typer af farveblindhed.

Hos Vischeck kan man også downloade plugins til blandt andet Photoshop, der også kan simulere forskellige former for farveblindhed. Resultatet bliver noget mere præcist med dette plugin, da Vischecks online service kan svinge noget i kvalitet afhængig hvilken typer billeder, der skal behandles.

Hvis man skal have hjælp til at lave tabeller, der lever op til tilgængelighedsreglerne kan man få hjælp på: <http://www.accessify.com/tools-and-wizards/accessibility-tools/table-builder/>. Man starter med at vælge om tabellen er simpel eller kompliceret, og vælger derefter antallet af koloner, rækker, overskrifter etc. Værktøjet giver både XHTML og CSS, som man enten kan bruge direkte eller arbejde videre med.

### **3.8 Udbudsværktøjskassen**

Hvis man ikke selv laver sine webløsninger fra bunden, men bruger en udefrakommende leverandør, er det vigtigt at man har nogle konkrete kravspecifikationer, så leverandørerne har et konkret og ensartet udgangspunkt at give tilbud på, og senere lave en tilgængelig løsning efter.

Derfor har IT- og Telestyrelsen lavet en hjemmeside, der hedder "Udbudsværktøjskassen" (<http://vkassen.oio.dk/omudbudsvaerktoejskassen/>).

Med Udbudsværktøjskassen kan man nemt specificere de krav man har til det færdige produkt. Udbudsværktøjskassen er bygget op som en guide, hvor man gennem en række valg, når frem til de konkrete krav til applikationen.

1. Først vælges hvilken type applikation der er tale om.  
F.eks. Er det en web-baseret løsning, en software løsning eller en hardware løsning?  
Skal der laves et nyt websted, eller ændres et eksisterende?
2. Så besvares en række ja/nej spørgsmål om applikationen  
F.eks. Er det et CMS? Benyttes der billedrepræsentation af programelementer?  
Benyttes der tabeller til layout? Skal brugeren kunne benytte formularer?
3. Så vælges hvilket niveau kravene skal opfyldes på.  
Guiden indeholder 3 niveauer, hvor det mindst omfattende ikke sikrer adgang for alle brugere. Derfor anbefales at vælge niveau 2. Dermed opfyldes kravene på niveau 1 og 2 og man er sikret en tilgængelig løsning. Kravene på niveau 3 er de skrappeste og ikke nødvendige at leve op til i forhold til lovgivningen.
4. Til sidst gives en liste over konkrete krav til opfyldelse af tilgængelighed for den valgte applikation på det valgte tilgængelighedsniveau. Kravene kan nu copy-pastes og bruges i udbudsmaterialer og kravspecifikationer, for at sikre at løsningen lever op til reglerne.

### **3.9 Bedst på nettet**

Bedst på nettet er en alternativ indgangsvinkel til borgerservice, brugervenlighed og tilgængelighed. Det er en konkurrence som offentlige institutioner kan bruge til at måle blandt andet deres tilgængelighed i forhold til andre offentlige institutioner i Danmark. Der konkurreres i forskellige kategorier, og konkurrencen virker til at være et kraftigt incitament for visse websteder til at gøre noget ved tilgængeligheden.

Bedst på nettet blev startet i 2001 og skulle løbe som et 3-årigt projekt med det formål at forbedre kvaliteten af det offentlige netsteder. Resultaterne var dog meget positive i form af forbedrede websteder, hvor evalueringer viste at Bedst på Nettet ifølge Videnskabsministeriet har været en væsentlig drivkraft i denne udvikling. Derfor har videnskabsministeriet besluttet at fortsætte projektet

Selvom tilgængelighed kun tæller for 25 % af vurderingsgrundlaget, mener jeg det er relevant at se på Bedst på Nettet. Dels fordi at de øvrige vurderingskategorier lægger sig tæt op af nogle de brugervenlighedsaspekter, som jeg vil diskutere i næste sektion. Desuden er Bedst på Nettet med til at sætte fokus på tilgængelighed, og også med til at rejse debat omkring tilgængelighed. Det har jeg f.eks. mærket i min praktikperiode, hvor de fleste har haft en holdning til Bedst på nettet og deres vurderingskriterier.

Udover Bedst **på** Nettet findes også konkurrencen Bedst **til** Nettet, som henvender sig til leverandørerne. Dette kan være en hjælp for offentlige institutioner, der skal ud og finde en ny leverandør af systemer.

Bedst på Nettet opererer med følgende overordnede kategorier:

### 3.9.1 Konventioner

Konventioner fokuserer på struktur og formidling. Hvis ikke man kan finde rundt på en hjemmeside og forstå teksten, så nytter det jo ikke meget, at den ligger på nettet. Derfor handler kategorien Konventioner om navigation, sprog og om at tage udgangspunkt i brugerens behov.

### 3.9.2 Digital selvbetjening

Digital selvbetjening ser på omfanget og karakteren af hjemmesidens selvbetjeningsløsninger samt brugervenligheden af dem.

### 3.9.3 Åbenhed og nytteværdi

Åbenhed og nytteværdi forholder sig til, om afsenderen af hjemmesiden er åben og imødekommende over for brugerens behov, og om indholdet er pålideligt og nyttigt.

### 3.9.4 Teknisk tilgængelighed

Kategorien Teknisk tilgængelighed dækker i store træk over de samme krav som W3.org.

Selvom kategorien teknisk tilgængelighed ikke bliver vægtet højere end de øvrige kategorier, er den en forudsætning for, at alle kan få adgang til indholdet på en hjemmeside. Derfor skal en hjemmeside opnå et minimumsresultat på 75 % af mulige point i denne kategori for at kunne blive nomineret og dermed vinde i Bedst på Nettet.

### 3.9.5 Debat om Bedst på Nettet

Som tidligere nævnt har Bedst på Nettet skabt både mange resultater og megen debat gennem tiderne. Debatten har været god fordi, den har skabt en generel fokus på tilgængelighed, og hvilke krav, der har været relevante og hvilke, der har været mindre relevante.

I takt med at fokus og viden på området er vokset ude på institutionerne, er kritikken også vokset. Derfor var det også glædeligt, da Bedst på Nettet i marts 2008 åbnede mulighed for at debattere vurderingskriterierne for 2008 på deres hjemmeside.

(<http://www.bedstpaanettet.dk/vurderingsgrundlag%202007/diskussion-af-vurderingsgrundlaget-2007>).

På trods af meget både berettiget og konstruktiv kritik, fik diskussionen desværre kun lov til at løbe i 2 måneder. Det er min fornemmelse, at mange slet ikke opdagede at muligheden var til stede og med mængden af kritik, virker det besynderligt, at man ikke lod debatten løbe. Debat og brugerinvolvering er jo ellers helt i tråd med Bedst på Nettets anbefalinger.

Nogle af kritikpunkter har gået på urimelig pointtildeling. Er det f.eks. rimeligt at man kan få point, udelukkende ved at lægge et spørgeskema ud på hjemmesiden, uden der følger nogen forpligtigelse med til at reagere på de indkomne resultater. Eller ved bevidstløst at linke til sider som Bedst på Nettet anbefaler man linker til.

Andre kritikpunkter har været at nogle vurderingskriterier ikke er relevante for lige præcis ens eget websted, at vurderingskriterierne er for gamle og trænger til fornyelse eller omvendt at vurderingerne ligger under for forskellige web 2.0 modeluner.

Og flere af kritikpunkterne er ganske vist berettigede, især fordi mange websteder er meget forskellige i både målgrupper, form og indhold. Derfor kan krav, der er relevante for nogle hjemmesider, virke meningsløse eller måske endda direkte misvisende for andre.

Omvendt kan man argumentere at fordelene ved at have meget konkrete krav, er at vurderingsgrundlaget bliver meget gennemskueligt.

For mig at se, er det vigtigste dog ikke om det ene kriterium skal med eller ej. Det væsentlige er hele den debat som Bedst på Nettet rejser. Debatten viser, at det er vedkommende for folk ude på institutionerne, og at man i forbindelse med Bedst på Nettet konkurrencen tager stilling til både egne og konkurrenternes hjemmesider. Alt andet lige, vil den øgede opmærksomhed give en bedre kvalitet på hjemmesiderne.

## 4 Brugervenlighed og tilgængelighed

Groft sagt kan man dele brugervenligheden op i to områder, der er særligt vigtige, når vi vil hjælpe brugere med funktionsnedsættelser, nemlig forhold omkring layout og opsætning, og brugervenlige tekster.

### 4.1 Brugervenligt layout

Hvis man ser på brugervenligt layout i forhold til tilgængelighed er de vigtigste punkter:

- Sørg for tydelige hierarkier
- Benyt konventioner, der er opstået omkring Internettet.
- Del siden op i tydeligt definerede områder
- Minimér støj

Punkterne beskæftiger sig med krav 12 og 13, der omhandler navigation og overblik.

#### 4.1.1 Tydelige hierarkier

Ved at vise, hvad der er vigtigt og mindre vigtigt, gør man det meget lettere for brugeren at overskue siden. Brugeren skal hverken bruge energi på at **finde** det væsentlige, eller på at afgøre **hvad**, det væsentlige på siden er. Han kan gå direkte i gang med at undersøge, om det siden tilbyder, er det han leder efter.

Et tydeligt hierarki vil gøre brugeroplevelsen bedre og mere flydende, og typisk efterlade brugeren med mere energi og fokus til de væsentlige opgaver på siden.

Krug, 2006: *"A good visual hierarchy saves us work by preprocessing the page for us, organizing and prioritizing its contents in a way that we can grasp almost instantly."*

Hierarkier skaber vi f.eks. med bevidst brug af grafiske og typografiske kontraster. F.eks. størrelseskontrasten, som er den mest åbenlyse. Men også farvekontraster er gode til at skabe hierarkier med (Birkvig, 2004:41).

For at kontrasterne kan opfylde opgaven er det vigtigt at man bruger tydelige kontraster. Brug af subtile kontraster kan være gode til signalere eksklusivitet og seriøsitet, men i forhold til at skabe tydelige hierarkier er de ikke meget bevendt.

#### 4.1.2 Konventioner på nettet

I Internettets korte historie er der allerede opstået en række konventioner eller standarder for, hvordan en typisk hjemmeside er opbygget. Konventionerne er opstået ved, at de effektive og gode idéer er blevet gentaget af andre designere.

En af de mest udbredte konventioner er at links altid er understreget, og at tekst, der er understreget, er links. Dette skal vi selvfølgelig udnytte ved altid at understrege vores links og aldrig at bruge understregninger til andet end links.

En anden veldefineret konvention er at vandrette lister i toppen, og lodrette lister i venstre side er menuer.

Der er ikke meget prestige for webdesignere i at følge konventionerne, da det grafiske udtryk kan blive opfattet som kedeligt. Det designeren laver i dag ligner det han designede i går. Ofte har virksomhederne også en forventning om, at de professionelle designere leverer noget, der er anderledes og iøjnefaldende. Men som brugere er vi som regel glade for traditionelle løsninger, da de er lette at afkode, funktionerne lette at finde og vi føler os generelt trygge ved et websted

med få overraskelser. Den grafiske oplevelse er oftest underordnet for brugeren. Flot grafisk design kan være lækkert, men det væsentligste er, at det ikke står i vejen for brugeroplevelsen på siden.

Ifølge Jakob Nielsen (<http://www.useit.com/alertbox/20040913.html>) sikrer brug af konventioner og standarder at brugerne:

- ved hvilke funktioner de kan forvente
- ved hvordan disse funktioner vil se ud på grænsefladen
- ved hvor man kan finde funktionerne på siden og på sitet
- ved hvordan man skal bruge funktionerne for at nå sine mål
- behøver ikke at bruge tid på at gennemskue ukendte designelementer
- overser ikke vigtige funktioner, fordi de ikke ligner standard design elementer

Disse fordele øger brugerens muligheder for at opnå deres mål med besøget. Og det giver dem en fornemmelse af at beherske hjemmesiden og øger dermed deres tilfredshed med websitet.

#### 4.1.3 Tydeligt definerede områder

En side bliver mere overskuelig, hvis siden er opdelt i flere mindre og veldefinerede områder. Inden for hvert område anbringer man så beslægtede former for information. Det gør det lettere for brugeren hvis vi kan opdele en sides mange elementer i 3-5 overordnede kategorier.

Eyetracking studier viser, at brugerne meget hurtigt bestemmer hvilke sektioner på en webside, der indeholder de brugbare informationer. De øvrige områder bliver ofte ignoreret (Krug, 2006: 37). Et godt eksempel fra vores hverdag er måden hvorpå vi ignorerer bannerreklamer i toppen af en side. Den slags "ubevidste" valg træffer vi mange af, især hvis siden samtidig følger konventionerne.

Når man skal inddele websiden er gestaltlovene et godt redskab at bruge (Birkvig, 2004:41; Pedersen 2002: 55 ).

#### 4.1.4 Minimér støj

Jo flere forskellige elementer, der er på siden, jo højere er risikoen for støj. Støjen kan komme fra flere typer af elementer. Det kan f.eks. dreje sig om:

- Grafisk støj.  
Undgå unødigt pynt. Grafik skal have en mening.
- Multimedie-elementer.  
Det kan være musik, flash eller film, der starter op uden brugeren har klikket på dem.
- For mange informationer.  
Der kan simpelt være for mange informationer på for lidt plads. Hvis et website er informationstungt er det vigtigt at finde nogle gode overskuelige strukturer. Hvis for meget information bliver præsenteret for brugeren på engang risikere vi, at han mister overblikket.
- For mange funktioner.  
Undgå funktioner for funktionernes egen eller webudviklernes skyld.  
*"... mine samtaler med typiske brugere viser, at brugere typisk kun udnytter 20-30 % af funktionaliteten. Resten er set fra brugerens synspunkt overflødig – eller direkte forstyrrende for den enkle betjening og det hurtige overblik."* (Molich, 2006: 19).

I nogle situationer kan en overdreven optimering til brugere med funktionsnedsættelse måske være generende for den generelle brugervenlighed. F.eks. vil link/grafik der henviser til skærmpoplæsere eller specielle handicapsider bidrage til støjen på siden, og gøre det vanskeligere at finde de væsentlige informationer og funktioner, både for de almindelige brugere og brugerne med funktionsnedsættelser.

## 4.2 Statsbibliotekets layout

Statsbibliotekets hjemmeside følger de fleste anbefalinger til brugervenligt layout.

- Tydeligt hierarki. Søgefeltet er det vigtigste
- Vigtig navigation tilknyttet søgning og materialer i toppen
- Kun få elementer på hjemmesiden
- Benytter konventioner. Menu i toppen, understregede links, kontaktoplysninger i bunden
- Tydeligt adskilte elementer. Sidespalte, top, indholdsområde

The screenshot shows the homepage of Statsbiblioteket. At the top left is the logo 'STATSBIBLIOTEKET'. To the right are links for 'Stemmap' and 'Dine lån'. Below the logo is a navigation bar with 'Du er her: Forside'. A search bar with a 'Søg' button and links for 'Dine lån', 'Ret profil', and 'Ny bruger' is prominent. Below the search bar is a horizontal menu with categories like 'E-tidsskrifter', 'Tidsskrifter', 'Databaser', 'Tv og radio', 'Aviser', 'Musik', and 'Udvidet søgning'. The main content area is divided into several sections: 'Kontakt og dialog' with a link to 'Spørg Statsbiblioteket'; 'Kurser' with a link to 'Program for foråret 2008'; 'Nyttige oplysninger' with multiple links for 'Åbningstider', 'Ny bruger', 'Ekspeditionstid og levering', 'Lånetid og gebyrer', 'Betaling for print', 'Fysiske faciliteter', 'Lån fra andre biblioteker', and 'Hjemmeadgang til e-ressourcer netmusik.dk'; and 'Om Statsbiblioteket' with links for 'Opgaver og organisation', 'Job og karriere', 'Fakturering og betaling', 'Presse', 'Udgivelser', 'Pligtaflevering', 'Om netstedet', and 'Find en medarbejder'. On the right side, there are two vertical sections: 'Meddelelser' with links for 'Nyhedsbrev til Humaniora', 'Grupperum på 1. sal kan reserveres', 'Ændring i elektronisk levering af artikelkopier', and 'Statsbiblioteket justerer afhentning af materialer'; and 'Videncentre' with links for 'For biblioteker' and 'BiblioteksCenter for Integration'. At the bottom, there is contact information: 'Statsbiblioteket, Universitetsparken, 8000 Århus C, 8946 2022, sb@statsbiblioteket.dk'.

Statsbibliotekets forsider.

Et sted, hvor der er problemer er med menuerne på forsiden. Disse er lavet som brødtekst, og er meget vanskelige at skimme og overskue. Man bliver derfor nødt til at læse hele tekstblokken for at finde et specifikt link.

This image shows a close-up of the 'Nyttige oplysninger' and 'Om Statsbiblioteket' sections from the homepage. The 'Nyttige oplysninger' section contains a list of links: 'Åbningstider', 'Ny bruger', 'Ekspeditionstid og levering', 'Lånetid og gebyrer', 'Betaling for print', 'Fysiske faciliteter', 'Lån fra andre biblioteker', and 'Hjemmeadgang til e-ressourcer netmusik.dk'. The 'Om Statsbiblioteket' section contains a list of links: 'Opgaver og organisation', 'Job og karriere', 'Fakturering og betaling', 'Presse', 'Udgivelser', 'Pligtaflevering', 'Om netstedet', and 'Find en medarbejder'.

Forsidemenu fra statsbibliotekt.dk

Ved i stedet at benytte sig af en listeopstilling vil man opnå flere fordele. Lister er meget hurtigere at overskue. Det er langt lettere at finde starten på linket, da listen har fast forkant. Oftest behøver brugerne kun at læse de første 1-2 ord i listen, for at få en idé om hvor linket fører hen.

**Nyttige oplysninger**

- [Åbningstider](#)
- [Ny bruger](#)
- [Ekspeditionstid og levering](#)
- [Betaling for print](#)
- [Hjemmeadgang til e-ressourcer](#)
- [Fysiske faciliteter](#)
- [Lån fra andre biblioteker](#)
- [netmusik.dk](#)

*Den øverste af menuerne i listeform.*

Undersøgelse foretaget af Jakob Nielsen (<http://www.useit.com/alertbox/navigation-menu-alignment.html>), viser at brugerne skanner lister ved hurtigt at lade øjnene glide ned over den venstre forkant. Menuer og lister, der er højrestillede er derfor meget vanskeligere at skanne. Og helt galt bliver det, hvis links ikke er i listeform.

Ulempen ved listen er at den fylder mere end den nuværende løsning. Men det kunne måske være en lejlighed til at få sorteret i linkene, hvilket også ville øge overskueligheden. Tiloversblevne links kunne eventuelt placeres ude i sidespalten, så der stadig er adgang fra forsiden.

Som et kompromis mellem en traditionel liste visning og Statsbibliotekets nuværende, kan man sætte et skilletegn ind foran hver link, for at markere her starter et nyt link. Det vil gøre det lettere at finde starten på linket og dermed lettere at skimme de første ord i hvert link.

**Nyttige oplysninger**

- [Åbningstider](#) ● [Ny bruger](#) ● [Ekspeditionstid og levering](#) ● [Lånetid og gebyrer](#)
- [Fysiske faciliteter](#) ● [Lån fra andre biblioteker](#) ● [Hjemmeadgang til e-ressourcer](#)
- [netmusik.dk](#) ● [Betaling for kopi](#)

**Om Statsbiblioteket**

- [Opgaver og organisation](#) ● [Job og karriere](#) ● [Fakturering og betaling](#) ● [Presse](#)
- [Pligtanfærelse](#) ● [Om netstedet](#) ● [Find en medarbejder](#) ● [Udgivelser](#)

*Skilletegn sat ind i menuerne.*



Et andet reelt problem optræder i søgeresultaterne, hvor der i højre spalter er en linkliste med 20 muligheder for at begrænse sin søgen. Hver kategori indeholder op til ca. 20 links med nye muligheder. Udviklingsafdelingen er dog opmærksom på problematikken, og holder øje med log statistikkerne for at afgøre hvilke kategorier, der bruges mest.

Som man kan forvente, er det kun et fåtal af kategorierne, der reelt set anvendes af brugerne. Både fordi antallet i sig selv virker afskrækkende, men også fordi mange af kategorierne er ekstremt specialiserede. Denne service i Summa er blandt andet lavet ud fra professionelle brugeres ønsker, altså bibliotekarer og forskere, og har derfor kun meget lille interesse for den ordinære bruger. Sådanne specialiserede informationer hører selvfølgelig ikke hjemme i en brugervenlig grænseflade. Derfor må der forventes en mere overskuelig opdeling med næste version af Summa.

### **4.3 Brugervenlige tekster**

W3c's krav 14 om et klart og forståeligt sprog er et af de vigtigste krav. Det er til gavn for alle brugere. Men især hjælper det den store gruppe af brugere med læsevanskeligheder eller med problemer på det kognitive område.

Læsevenlighed kan opdeles i 2 begreber. Læselighed. Er teksten fysisk til at læse? Og læsbarhed. Er teksten til at forstå?

#### **4.3.1 Læselighed**

God læselighed handler i vid udstrækning om typografi, og der er mange konkrete regler man kan følge.

- Skriftstørrelsen afhænger af skrifttypen, men 10 pixels er et absolut minimum. Til ældre læsere anbefaler man mindst 12 pixels ved f.eks. Verdana. Men som minimum skal skriftstørrelsen kunne ændres via browseren. I de seneste års har vi heldigvis set en tendens til at store skrifttyper er blevet moderne. Endog i sådan en grad at stor skrift er blevet et kendetegn for mange web 2.0 fænomener.
- Skrifttypen skal være egnet til skærmbrug. Traditionelt anbefaler man Sans Serif skrifter, som f.eks. Verdana (Pedersen, 2002: 100). Men med bedre skærme og skrifttyper, specielt udviklet til skærmbrug, er det ikke længere en betingelse. Man kan sagtens bruge Serif skrifter, som f.eks. Georgia til skærmen, så længe skrifterne er åbne og har en stor x-højde.
- Der skal være en passende kontrast mellem tekst og baggrund. Det siger sig selv at kontrasten kan blive for lille, men det omvendte kan også være tilfældet. F.eks. kan helt sort tekst på ren hvid baggrund optræde flimrende på skærmen. Man kan med fordel enten gøre teksten meget mørk grå eller baggrunden off-white, for en mere behagelig læseoplevelse.
- Undgå kapitæler. Det er svært at læse tekster, der udelukkende består af store bogstaver. Det gælder også korte passager, så hvis noget skal fremhæves skal man i stedet bruge fed tekst.
- Antallet af tegn per linje bør ikke overstige 65-70 tegn, ellers bliver det for svært for øjet at finde starten på den næste linje.
- Det er vigtigt at lave mange afsnit og bruge luft mellem afsnittene. Det får teksten til at virke lettere og mere venlig. Store blokke af tekster på nettet kan tage pusten fra selv gode læsere.

- Brug altid venstrejusterede margener på brødtekster og menuer (Birkvig, 2007:23). Det gør det lettere for øjet at finde næste linje.

### 4.3.2 Læsbarhed

For at sikre en god læsbarhed:

- Fremhæv nøgleord med f.eks. Fed eller Link  
Fremhævede ord hjælper skimming af tekster, men man skal være opmærksom på, at det samtidig gør læsningen vanskeligere (Molich, 2006: 102). Dette kan være en væsentlig pointe at have i baghovedet da meget svage læsere har svært ved at skimme tekster, men bliver nødt til at læse teksten i sin helhed.
- Brug mange og beskrivende overskrifter. Det gør det lettere hurtigere at finde det man leder efter.
- Brug forskellige niveauer i overskrifterne. Det gør det tydeligere, hvilke hovedområder de enkelte afsnit beskæftiger sig med.
- Brug punktopstillinger. Gør det igen lettere at finde de informationer, man leder efter, da lister er hurtigere at skimme. Endvidere kan lister bryde tunge brødtekster, og få dem til at fremtræde lettere.
- Skriv det vigtigste først. Brugere læser kun ca. 20-30 % af indholdet, så derfor skal det vigtigste skrives i starten af teksten. (Jakob Nielsen: <http://www.useit.com/alertbox/percent-text-read.html>). Jakob Nielsen nyhedsbreve kan også bruges som et lysende eksempel på hvordan en 2-3 linjers konklusion i starten af artiklerne, gør at brugeren lynhurtigt, kan afgøre om artiklen indeholder de informationer han leder efter.
- Skriv kort og præcist. Tekster på skærmen er meget vanskeligere at læse, derfor skal alt fyld skæres væk. Brug korte afsnit og korte sætninger (Krug, 2006:45). Det kan være svært at overholde dette punkt, især hvis man har noget på hjertet. En god løsning er at få udenforstående til at kigge kritisk på ens tekster.
- Brug talesprog. Det handler om at gøre tekster personlige og sprogbruget hverdagsagtigt. Skriv hellere "du" end "man", og skriv hellere "jobmuligheder" end "beskæftigelsesprognoser".
- Brug aktiv form i stedet for passiv form (Molich, 2006: 106). Passiv form har den svaghed, at den ikke fortæller, **hvem** der skal handle. Så skriv hellere "Du skal aflevere bøgerne inden" end "Bøgerne afleveres senest".

### 4.3.3 Læsbarhed og Lixtal

Lix står for "Læsbarhedsindex", og dækker over en beregningsmetode man benytter til at afgøre en teksts læsbarhed.

**Formel for beregning af Lixtal:**

$$\text{LIX} = \text{O}/\text{P} + (\text{L} * 100)/\text{O}$$

O = antal ord i teksten

P = antal punktummer i teksten

L = antal lange ord (over 6 bogstaver lange)

>55 Meget svær, faglitteratur på akademisk niveau, lovttekster.

45-54 Svær, f.eks. saglige bøger, populærvidenskabelige værker, akademiske udgivelser.

35-44 Middel, f.eks. dagblade og tidsskrifter.

25-34 Let for øvede læsere, f.eks. ugebladslitteratur og skønlitteratur for voksne.

<24 Let tekst for alle læsere, f.eks. børnelitteratur

Beregninger kan dog ikke bruges på alle tekster, men er mest egnede til lange ensartede tekster. Endvidere beskriver Lixtallet også udelukkende tekstens læsbarhed og tager ikke hensyn til teksten læselighed. Det betyder at lixtal kan bruges til at give indikationer, men absolut ikke sandheder om teksten.

It-og Telestyrelse skriver om Lixtal: *Man kan diskutere, hvorvidt meget faglige tekster kan gøres letlæselige, men generelle ikke-faglige tekster skal altid skrives i korte klare sætninger med et lavt lixtal.*

Som en kuriositet kan nævnes at ovenstående sætning er taget ud et afsnit, der omhandler Lixtal. Afsnittet indeholder: 3 sætninger og 72 ord, og har et Lixtal på 50. Eksemplet viser også at det kan være rigtigt svært at få Lixtallet ned. Det paradoksale er, at hvis man bruger flere fyldord som f.eks. jo, så og da, vil Lixtallet falde, fordi andelen af lange ord i teksten vil falde.

Som et lille sideprojekt har jeg lavet en online Lixtalsberegner, som kan findes her:

(<http://www.nielsgamborg.dk/lix>)

#### **4.4 Statsbibliotekets tekster**

Jeg har som **eksempel** udvalgt en side, der beskriver, hvordan nye lånere skal forholde sig. Siden er udvalgt dels fordi det er en side, som de fleste af brugerne vil se på et tidspunkt, dels fordi den får mange hits og endelig fordi den indeholder en del brødtekst.

## Ny bruger

For at låne materialer skal du være registreret som [bruger](#).

Sygesikringsbeviset/sundhedskortet fungerer som lånerkort. Mister du dit sygesikringsbevis/sundhedskort, skal Statsbiblioteket straks informeres, da du hæfter for brugen.

### Pinkode

Når du er oprettet som ny bruger, vil du modtage en e-mail med pinkode og et link, du skal aktivere. Pinkoden udstedes af Statsbiblioteket og kan ændres i dine personlige brugeroplysninger. [Ret profil](#). Her kan du også få tilsendt din pinkode, hvis du har glemt den.

Cpr-nummeret (uden bindestreg) fungerer som bruger-id.

### Manglende e-mail

Har du ingen e-mail adresse, skal du møde personligt op i Statsbibliotekets Information i forhallen for at blive registreret.

### Brugerregistrering for biblioteker og institutioner

Biblioteker, firmaer og institutioner, der ikke er registreret, bedes kontakte Statsbiblioteket på [udlaan@statsbiblioteket.dk](mailto:udlaan@statsbiblioteket.dk).

## Adresseændring

Adresseændring, ændring af e-mailadresse eller pinkode foretages under dine personlige oplysninger. [Ret profil](#).

*Toppen af eksempelside fra Statsbiblioteket.dk*

På eksemplet kan man se, at Statsbiblioteket lever op til mange af reglerne for brugervenlige tekster.

- Teksten er opdelt med mange overskrifter i forskellige niveauer. Størrelseskontrasten mellem øverste niveau og mellemrubrikkerne er tydelige. Der er ingen tvivl om, hvor et nyt emne starter.
- Overskrifter er sigende for indholdet
- Link tekster er understregede og sigende.
- Afsnit og sætninger er korte
- Sproget er hovedsageligt personligt, kort og klart.

En nærmere analyse afslører dog også lidt vanskeligheder. Sproget er nogle steder aktivt i sin form, men har en tendens til at glide over i passiv form. ”Mister du dit sygesikringsbevis skal Statsbibliotekets straks **informeres**” eller ”ændring **foretages** under dine...”

Lixtallet for brødteksten på ovenstående eksempel er 45 (13 sætninger, 126 ord og 9,6 ord pr. sætning). Det placerer teksten mellem middel og svær, og er i overkanten i forhold til anbefalingerne.

Men overordnet ser Statsbibliotekets tekster brugervenlige ud. Som tidligere nævnt er rådene om brugervenlige tekster måske også det sværeste at overholde i praksis. Både fordi rådene er åbne for fortolkninger, men også fordi, der på Statsbiblioteket findes en lang række skribenter og teksterne er dynamiske og hele tiden udvikler sig. For at sikre den skriftlige kvalitet har Statsbiblioteket udnævnt en række overordnede webredaktører, der har ansvaret for at alle tekster lever op til Statsbibliotekets politikker på området.

Jeg har i min praktik periode været med til at formulere et kort design dokument, der har skullet formidle Statsbibliotekets principper for design og brugervenlighed. Dokumentet skal give retningslinjer for statsbibliotekets ca. 20 webredaktører. Et af hovedpunkterne i dette dokument er opskrift på, hvordan man formulerer sig brugervenligt, når man skriver til nettet. (se Bilag 2)

## **4.5 Brugertests**

Ligegyldig hvor godt vi efterlever reglerne, krav og retningslinjer kommer vi ikke uden om at brugerteste vores produkter.

Rolf Molich (Molich, 2006:17): *"Erfaringer viser, at selv en nok så dygtig webansvarlig ikke kan gøre sig håb om at konstruere en brugervenlig grænseflade uden et godt og nært samarbejde med typiske brugere"*

Rolf Molich siger også, at hver gang der opstår den mindste tvivl eller diskussion om brugervenlighed i en udviklingsgruppe, er det bevis på at ens viden på området er for lille, og derfor skal der testes.

Jakob Nielsen (<http://www.useit.com/alertbox/accessibility.html>): *"A strict focus on accessibility as a scorecard item doesn't help users with disabilities. To help these users accomplish critical tasks, you must adopt a usability perspective."*

Jakob Niensens pointe er at ligegyldigt, hvor godt du følger de forskellige tekniske specifikationer til tilgængelighed, vil disse ofte være så detaljerede og løsreven fra sammenhængen, at de ikke give et fuldstændigt billede af tilgængeligheden på hele websitet.

Ifølge Jakob Nielsen skal der testes på realistiske opgaver for websitet, for at lave en endelig dom. Og det behøver ikke at være store forkromede brugertests. Ofte vil det være nok at involvere 3-4 ansatte i virksomheden, der har forskellige former for funktionsnedsættelse. Det vil give et godt billede af om tilgængeligheden for websitet også virker i praksis.

Steven Krug: *"Testing one user is 100 % better than testing none."*

Steven Krug følger alle de andre brugervenlighedseksperter. Brugertests er essentielle i alt brugervenlighedsarbejde. Han anbefaler at teste med kun 3 testpersoner (Krug, 2006: 139), og så gentage testen, når de første problemer er løst. Man finder på den måde langt flere uhensigtsmæssigheder end ved at teste kun en gang med f.eks. 8 brugere.

Når man tester tilgængelighed skal man være opmærksom på, at målgruppen er meget bred og har vidt forskellige behov. F.eks. vil ordblinde brugere ofte foretrække mange ikoner og visuelle forklaringer, hvor blinde foretrækker så lidt visuelt materiale som muligt.

Ofte vil det kun være muligt at finde ganske få brugere til brugertests på tilgængelighedsområdet, og derfor må man som regel vægte kvalitative tests frem for kvantitative. F.eks. vil "tænke højt tests" være særdeles egnede (Molich, 2006: 134).

## 5 Teknologi og tilgængelighed

Jeg vil i denne sektion beskrive den JavaScript uafhængige kurv funktion, som jeg har lavet i min praktik. Kurven skal ses som et eksempel på, hvordan man angriber et specifikt tilgængelighedsområde med en teknologisk indgangsvinkel.

Jeg vil starte med en række overordnede beskrivelser af nogle af de anvendte sprog, der er nye i forhold til hvad vi har lært på uddannelsen. Det drejer sig om:

- JSP
- XML
- XSLT

Dernæst vil jeg beskrive en række essentielle funktioner i forhold til en kurvfunktion og give en overordnet beskrivelse af kurven, inden jeg dykker helt ned i detaljerne og vil kommentere på nogle af de konkrete kodeeksempler i min løsning.

Endelig vil jeg afslutte sektionen med en opsamling på, hvor den JavaScript uafhængige kurv er lige nu, og hvilke udviklingsmuligheder der er for den.

### 5.1 JSP oversigt

JSP (Java Server Pages) er et serverbaseret scriptsprog, der kort fortalt bruges til at generere den webside, der bliver sendt til brugerens browser. JSP afvikles på serveren, og er derfor i modsætning til f.eks. JavaScript, som er et klientbaseret scriptsprog, uafhængig af hvilken browserteknologi brugeren benytter.

Der findes en række forskellige serverside scriptsprog udover JSP, blandt andet PHP, som er det sprog vi har lært på uddannelsen, og som jeg derfor vil bruge som sammenligningsgrundlag. De to sprog har rigtig mange lighedspunkter og kan i store træk det samme. De serverbaserede scriptsprog bruges hovedsageligt til at skabe og vedligeholde dynamisk indhold på websider.

JSP er som navnet antyder baseret på Java, der er et objektorienteret sprog. Objektorienteret betyder lidt simplificeret at alle programstumper opfattes som klart definerede og adskilte objekter. Dette har flere fordele. F.eks. bliver programstumperne meget lette at genbruge, da de hver for sig løser klart definerede opgaver. En anden fordel er at vores omverden også ofte består af tydeligt afgrænsede objekter, og dermed kan det være lettere at se sammenhængen mellem omverdenen og de programmer man udvikler. Ulempen ved den objektorienterede tankegang, er at den kræver en større tilvænning end dens modsætning, procedural programmering, hvor programmeringen ses som sekvenser eller procedurer, hvor man løser én problemstilling af gangen.

Java stiller en mængde foruddefinerede objekter og klasser, samlet i biblioteker til rådighed for programmøren. Udover disse kan man også selv skrive sine klasser, og det har udviklerne på Statsbiblioteket gjort i det omfang, de har haft brug for særlige funktioner, som standardbibliotekerne ikke har stillet til rådighed. Fælles for alle disse objekter og klasser er, at man først skal lave en reference til bibliotekerne fra de filer, hvor man skal bruge de pågældende funktioner.

Java er meget stringent i sine krav til variabler. Alle variabler skal typebestemmes og typerne kan ikke efterfølgende ændres. Kun indholdet af variablerne kan ændres. Det gør JSP noget vanskeligere at arbejde med end f.eks. PHP, som er et relativt tilgivende sprog. Derfor bliver PHP også ofte brugt som et indgangssprog af nye programmører og webudviklere. Efter sigende skulle JSP, dog være meget let at lære, for programmører med baggrund i Java. Sådan en baggrund finder man ofte på universitetsuddannelser som f.eks.

datalogi og informationsvidenskab. Dette er også en medvirkende årsag til at Statsbiblioteket i benytter sig af både Java og JSP i deres udviklingsarbejde, da flertallet af medarbejderne som tidligere nævnt har en universitetsuddannelse som baggrund.

Java bliver betragtet som et langt mere sikkert sprog end PHP, og bliver derfor ofte brugt til sikkerhedskrævende webbrug, som online-banking etc. Denne sikkerhed hænger i høj grad sammen med de strengere krav til syntaks og variabel typer. Det er sværere for programmøren uforvarende at komme galt af sted. Det er lettere at lukke dørene efter sig, da en type ikke uden videre kan laves om og bruges til noget andet, end det den oprindeligt var tiltænkt.

Da JSP er meget mindre ubredt blandt små projekter end f.eks. PHP er det sværere at finde hjælp og tips på Internettet. Og når man er forvænt med php.net og hele communitiet omkring php, kan det være svært at vænne sig til den mere sparsomme dokumentation. Heldigvis ligner PHPs og JSPs grundliggende tankegang hinanden så meget, at det er i mange tilfælde er muligt at lade sig inspirere af PHP kode, og så efterfølgende prøve at lave det samme i JSP.

## 5.2 XML oversigt

XML står for eXtensible Markup Language. Som navnet antyder, er XML et opmærkningsprog ligesom HTML, og det ligner da også HTML på mange måder. En af de væsentligste forskelle er, at man i XML, selv kan navngive sine tags efter behov, og at XML ikke indeholder præsentationselementer, men udelukkende data. Endvidere har XML en noget strammere syntaks end HTML.

Et opmærkningsprog er et sæt af koder ofte kaldet "tags", som omslutter indhold/data, og fortæller personer eller programmer, hvad data indeholder. Et opmærkningsprog er ikke et programmeringsprog, og kan derfor ikke i sig selv, beregne eller analysere på data.

### XML eksempel. Data om bog

```
<shortrecord>
  <rdf:RDF xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
  syntax-ns#">
    <rdf:Description rdf:about="http://www.ilrt.bristol.ac.uk/people/cmdjb/">
      <dc:title>Java : An introduction to computer science and programming</dc:title>
      <dc:creator>Walter Savitch</dc:creator>
      <dc:type xml:lang="da">bog</dc:type>
      <dc:type xml:lang="en">book</dc:type>
      <dc:date>2001</dc:date>
      <dc:identifier>ISBN 0130316970</dc:identifier>
    </rdf:Description>
  </rdf:RDF>
</shortrecord>
```

XML bruges især til informations- og dataudveksling mellem forskellige systemer, teknologier og programmeringsprog. Formatet er meget velegnet til dette, blandt andet på grund af den store fleksibilitet, der følger med når man selv kan navngive tags efter behov, og fordi XML formatet kan overskues og læses af mennesker. Det gør XML meget brugbart til vidt forskellige formål spændende fra præsentationer f.eks. i kendte browserteknologier, over mobile enheder til opbevaringer i forskellige databasesystemer.

For at opnå denne store fleksibilitet er det nødvendigt at XML følger nogle ret stramme syntaktiske regler, for hvordan man f.eks. starter og afslutter tags, hvordan forskellige typer af data f.eks. data selv og metadata skrives og hvordan strukturen i et XML dokument er bygget op. Stram syntaks medfører mindre tvivl om selve dataindholdet, og gør det derfor lettere at tilpasse XML data til de specifikke behov, man måtte have.

På statsbiblioteket benyttes XML i mange sammenhænge. Blandt andet til at lave indeksering og til at levere søgeresultatet i et ensartet format til den del af webgruppen, som skal tage sig af den endelige præsentation af søgeresultatet på hjemmesiden.

XML "familien" er efterhånden udvidet med flere forskellige sprog og teknologier, som varetager forskellige funktioner i forhold til XML. Nogle af de mest benyttede er XSLT og XPath, som jeg også har benyttet i dette projekt.

### **5.3 XSLT oversigt**

XSLT står for eXtended Stylesheet Transformations. Det er et sprog der er udviklet til at transformere XML til andre formater som f.eks. HTML, XHTML eller nye XML dokumenter. XSLT er et redskab i kraftig udvikling og har allerede været udråbt som en af fremtidens teknologier i en årrække. Dette hænger selvfølgelig sammen med XMLs store udbredelse og systemuafhængighed, og det faktum, at man ved hjælp af XSLT kan bruge sine XML dokumenter på lige præcis den måde man ønsker det. Sproget er også rimelig let både at lære og bruge for lægmand, hvilket også er med til at sikre en stor udbredelse. På statsbiblioteket sidder f.eks. flere bibliotekarer og skriver XSLT.

XSLT filer er selv XML dokumenter, men kan i modsætning til XML foretage behandling/processering af data, hvilket bruges, når et XML dokument skal transformeres til en anden type dokument.

Rent praktisk bruges XSLT på den måde, at man skriver et stylesheet, som foretager transformationen. Dette stylesheet kan tilknyttes XML'en på flere måder. F.eks. enten ved at XML'en selv henviser til den tilhørende XSLT i starten af dokumentet (ligesom vi kender fra HTML, hvor man fra selve dokumentet peger på det tilhørende stylesheet). Eller man kan sende XML og XSLT dokumenterne gennem en "parser". En parser er en applikation, der undersøger og behandler data efter nogle opstillede regler eller en opskrift. På Statsbiblioteket bruger man en XML-parser fra et af standard Java bibliotekerne. Parseren tager XML'en, den tilhørende XSLT og en række parametre, og laver den nødvendige transformation.

XSLT benytter sig af et par andre teknologier, som er vigtige, når man skal transformere XML:

- XPath, som bruges til at angive placeringer af elementerne i XML dokumentets træstruktur, så man kan tage fat i de rigtige elementer. I praksis angives placering som en sti meget lignende den man bruger i filsystemer og i Internet adresser. Derudover findes en række definerede jokertegn, som gør at man kan udelade et eller flere elementer i stiangivelsen. Hvis man bruger XML eksemplet fra forrige side, vil en sti, der fører ned til bogens titel se sådan ud: **shortrecord/RDF/Description/title**
- Namespaces, som bruges til at definere XML tags, der umiddelbart ligner hinanden. Det klassiske eksempel er tagget table. Det kan både være navnet på en HTML tabel og et møbel. Løsningen er at specificere navnet med et namespace f.eks. <html:table> eller <furniture:table>

På Statsbiblioteket benyttes XSLT f.eks. til at præsentere søgeresultatet for brugerne. Relevante data fra XML'en bliver ved hjælp af XSLT omformet til HTML, som derefter skrives ud på skærmen. I mit projekt bruger jeg XSLT til at generere den HTML, der præsenterer kurven for brugeren.

### **5.4 Sessioner**

HTTP(den protokol der almindeligvis bruges til at kommunikere mellem servere og klienter på Internettet) er en såkaldt tilstandsløs protokol. Det vil sige, at hver anmodning, som afsendes, bliver behandlet uafhængigt af tidligere anmodninger. Der er altså intet 'forløb', alt glemmes fra

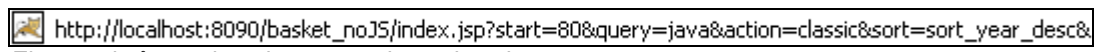


anmodning til anmodning. En JSP-side "glemmer" også alle data fra gang til gang, hvis man ikke gør noget for at gemme dem. Her kommer sessioner ind.

Sessioner bruges i alle serverside sprogene, og giver mulighed for at gemme oplysningerne/data, når brugeren bevæger sig rundt på et website. En session kan betragtes som en beholder, der kan opbevare data på serveren.

Man kan også opbevare de samme data på klienten, men så har brugeren mulighed for at manipulere med disse data, og der er større risiko for at miste data til tredje personer.

Endelig kan man også sende data med som parametre i URL'en (en url er en Internet adresse. En parameter som sendes med url'en starter med et spørgsmålstegn. F.eks. ?bogid=27. Browseren vil gå til den adresse, der står foran spørgsmålstegnet og ellers ignorere parameteren).

 `http://localhost:8090/basket_noJS/index.jsp?start=80&query=java&action=classic&sort=sort_year_desc&`  
*Eksempel på en url med en mængde medsendte parametre.*

Men URL 'en er også ubeskyttet og desuden tilgængelig for manipulation og dermed meget usikker. Derudover er pladsen i URL'en begrænset, og den kan ikke indeholde ret mange informationer. Pladsen i sessionen er dermed (i teorien) ubegrænset.

Ved at opbevare data på serveren er de altså mere sikre. Derfor bruges sessioner også altid, når personfølsomme data er på spil, f.eks. ved login, køb på nettet, kurvfunktioner etc.

I praksis foregår brugen af sessioner ved at brugeren tildeles en unik ID, som f.eks. gemmes i en cookie på klienten. Denne ID bruges til at verificere overfor serveren, hvilke brugere, der har tilknyttet hvilke data i sessionen.

Sessioner bliver gemt til brugeren lukker sin browser eller man kan fastsætte en udløbstid på sessionen, så sessionen bliver afsluttet hvis f.eks. brugeren efterlader sin computer inaktiv et vist tidsrum.

I JSP gemmes sessioner som et objekt i modsætning til i PHP, hvor de gemmes som et array. Det betyder i praksis, at det er en smule mere krævende at hente oplysningerne ud af dette sessions-objekt. Hvis man f.eks. gemmer en String i sessionsvariablen skal den hentes ud af objektet på en særlig måde. Den skal "castes" (tildeles type på ny) for at blive til en String igen.

## ***5.5 Kurvfunktion uden JavaScript. En oversigt.***

Her følger en beskrivelse af de funktioner en JavaScript-uafhængig kurv skal indeholde for, at man kan gemme bøger fra ens søgninger og senere bestille dem.

Først kunne det være smart, hvis applikationen selv kan afgøre om brugerens browser er i stand til benytte standard kurven, og hvis det ikke er tilfældet så automatisk starte den JavaScript uafhængige kurv.

Når man ikke kan bruge AJAX til at opdatere kurven i baggrunden, skal hele siden opdateres, hver gang brugeren foretager sig et valg. Det vil f.eks. sige, at når brugeren tilføjer eller fjerner noget fra sin kurv, skal man sende en lang række parametre med. Parametrene angiver f.eks., hvad der er søgt på, hvad der er sorteret på og hvilken side af søgeresultater, brugeren er på. Hvis man ikke gør det, vil brugeren opleve at blive smidt væk fra den side, han er på, og vil blive nødt til at starte sin søgning forfra.

Lad os kigge på kurvens grundlæggende funktioner med ovenstående i baghovedet.

- Når søgningen er foretaget, skal der ud for hver bog vises et link, hvor man kan tilføje bogen til kurven. Med linket skal bogens unikke id sendes med, for at man vide hvilken bog, der skal gemmes i kurven. Id'en sendes med som en parameter i url'en.
- Når der klikkes på dette link skal det først undersøges, om der eksisterer en kurv. Derfor undersøges det, om en session allerede eksisterer, hvis ikke oprettes (instancieres) en ny.
- Så tjekkes der for om bog-id'et allerede findes i kurven. Hvis det er tilfældet skal brugeren have en tilbagemelding om dette. Hvis ikke, så tilføjes bogens ID til kurven.
- Når det første materiale er tilføjet kurven, skal kurven vises på siden. I kurven skal man kunne se oplysninger om bogen, som titel, forfatter, årstal etc. I selve kurven/sessionen gemmer vi kun bogens id, så de øvrige data skal hentes frem ved hjælp af id'et.
- Når data er hentet skal kurvens udseende genereres. Dette gøres med XSLT fil, som omdanner XML data til HTML, der præsenterer kurven for brugeren.
- Fra kurven skal det være muligt at fjerne bogen fra kurven igen.
- Og til sidst skal det være muligt at bestille bøgerne i kurven. Både enkeltvis og alle på en gang.

## ***5.6 Kodeeksempler fra JavaScript og AJAX uafhængig kurv***

Afsnittene vil være bygget op så jeg eventuelt først vil komme med nogle generelle kommentarer. Derefter kommer, selve kodeeksemplet efterfulgt af forklaringen på koden.

### **5.6.1 Detektere om brugeren har JavaScript installeret**

Dette første skridt i hele processen byder på en række problemer. For det første findes der ingen funktioner i JSP, der kan undersøge om brugeren har JavaScript slået til. Men man kan bruge JavaScript til at undersøge sig selv. Hvis man får JavaScript til at udføre en bestemt handling og så derefter tjekker med JSP om handlingen rent faktisk er udført, kan man på den måde undersøge om brugeren har JavaScript aktiveret. Herefter redirecter man brugeren til enten den normale eller JavaScript uafhængige side.

Dette giver dog en række ulemper. F.eks. at 99,99 % af brugerne skal ledes igennem en unødigt JSP redirect. Dette vil alt andet lige give længere svartider og øget pres på serverne, hvilket igen har betydning for svartiderne.

Et andet problem i metoden er, at de fleste skærmlæsere rent faktisk vil have JavaScript slået til, da moderne skærmlæsere godt kan håndtere JavaScript. Det de oftest har problemer med er AJAX og det forholdsvis komplicerede JavaScript, der ligger til grund for den originale kurv og søgefunktion.

Alternativt kan man bruge HTML tagget "noscript", som man kan bruge til at skrive en besked til brugere, hvis browsere ikke kan håndtere scripts. Men dette vil så heller ikke fange de skærmlæsere, der har JavaScript slået til.

En tredje mulig løsning er at lave et link til den JavaScript uafhængige funktion på søgesiden. Ulempen ved denne metode er, at den øger det generelle støjniveau på siden.

En fjerde mulighed er at ligge linket som en skjult element, der kun vil være "synligt" for skærmlæsere. Dette er nok den mest "skånsomme" metode, forstået på den måde, at den kun vil berøre brugere med skærmlæsere.

## 5.6.2 Lægge materialer i kurven

**Filnavn: show\_short\_record.xsl. Linje: 309**

```
1 <a href="?start=${start}&query=${query}&id=${record_id}&sort=${sort}"
2   class="addRemoveLink" id="addRemove_{record_id}"> Læg i kurv</a>
```

Første skridt er at tage den traditionelle visning af søgeresultatet, og oprette et link, som man kan bruge til at lægge materialet i kurven. Linket peger på siden selv, men der bliver tilføjet en række parametre til URL'en, nemlig:

- Start: Hvilken post visningen skal starte ved
- Query: Søgeordet
- Id: Materialets unikke id.
- Sort: Parameter som søgningen er sorteret på

Parametrene hentes fra en række variabler, der allerede er i brug i den eksisterende XSLT.

Ved klik på linket bliver siden opdateret. I index filen bliver "basket.jsp" inkluderet, og nedenstående script tager imod de medsendte parametre.

**Filnavn: basket.jsp. Linje: 60**

```
1   if (session.getAttribute("kurv") == null) {
2     kurvindhold = new ArrayList<String>(); //kurvindhold instancieres
3   }else{
4     kurvindhold = (ArrayList)session.getAttribute("kurv");
5   }
```

Først undersøges det, om der er oprettet et sessionsobjekt med navnet kurv. Hvis ikke, instancieres en ny ArrayList med navnet "kurvindhold". Hvis sessionen allerede findes, hentes indholdet fra sessionen og gemmes i ArrayListen "kurvindhold".

**Filnavn: basket.jsp. Linje: 67**

```
1 if(bogId != null){
2   for(int i=0; i < kurvindhold.size() ; i++){ //array løbes igennem
3     if(bogId.equals(kurvindhold.get(i))){ //Der checkes om elementet allerede er i
kurven
4       out.println("Materialet er allerede i kurven");
5       ikurv = true; //Boolean sættes til true, hvis bogen allerede er i kurven
6     }
7   }
8   if(!ikurv){
9     kurvindhold.add(bogId); //hvis bogen ikke er i kurven tilføjes den
10  }
11  session.setAttribute("kurv", kurvindhold); //det nye Kurvindholdet føjes til sessionen
```

I line 1 undersøges om der er sat en bogId (er der klikket på "tilføj til kurv").

Så loopes indholdet af arrayet/kurven igennem.

I linje 3 undersøges det, om bogId er lig nogle af de eksisterende elementer i arrayet.

Hvis dette er tilfældet, gives tilbagemelding til brugeren om at materialet allerede er i kurven, og boolean variabelen "ikurv", sættes til true (Denne variabel er sat til false i starten af scriptet): I linje 8 tjekkes for om variabelen ikurv er false. Hvis det er tilfældet tilføjes materialet til arrayet, kurvindhold.

Endelig gemmes kurvindholdet i sessionsobjektet "kurv".

### 5.6.3 Fjerne materialer fra kurven

I kurvevisningen bliver der ud for hvert materiale lavet et link, der fjerner materialet fra kurven. Med linket sendes bogId som parameter.

**Filnavn: basket.jsp. Linje: 80**

```
1  if(fjernId != null){
2      for(int i=0; i < kurvindhold.size() ; i++){ //indholdet af kurv loopes igennem
3          if(fjernId.equals(kurvindhold.get(i))){ //Tjekkes efter "FjernID"
4              kurvindhold.remove(i);          //Indhold fjernes fra sessionen
5          }
6      }
7  }
8  session.setAttribute("kurv", kurvindhold); //det nye Kurvindholdet føjes til sessionen
```

For at fjerne materialet fra kurven tjekkes først om parameteren "fjernId" er sat. Derefter loopes arrayet kurvindhold igennem, og når fjernID matcher en bogID fjernes det aktuelle materiale fra arrayet. Sluttelig gemmes arrayet igen i sessionsobjektet.

### 5.6.4 Vise materialer i kurven

Her er hele den XSLT, der formaterer XML'en til det HTML, der vises på siden. Bemærk at man i XSLT kan blande HTML og XSLT tags næsten som man har lyst til, hvilket gør den meget simpel at arbejde med.

**Filnavn: basket\_show.xsl. linje 1**

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3     xmlns:xs="http://www.w3.org/2001/XMLSchema"
4
5 xmlns:message="http://xml.apache.org/xalan/java/dk.statsbiblioteket.summa.Util">
6 <xsl:output method="html" encoding="UTF-8"/>
7 <xsl:param name="query" />
8 <xsl:param name="start" />
9 <xsl:param name="sort" />
10 <xsl:param name="bogid" />
11 <xsl:template match="/">
12     <xsl:for-each select="shortrecord/RDF/Description">
13         <tr>
14             <td><xsl:value-of select="title"/></td>
15             <td><xsl:for-each select="creator">
16                 <xsl:value-of select="."/>
17             </xsl:for-each></td>
18             <td><xsl:value-of select="date"/></td>
19             <td><xsl:value-of select="type"/></td>
20             <td><a
21 href="?start={\$start}&amp;query={\$query}&amp;fjernid={\$bogid}&amp;sort={\$sort}"
22 style=" cursor: pointer; text-decoration: underline;">Fjern fra kurv</a></td>
23             <td><a
24 href="?start={\$start}&amp;query={\$query}&amp;fjernid={\$bogid}&amp;sort={\$sort}"
25 style=" cursor: pointer; text-decoration: underline;">(Bestil)</a></td>
26             <form action="/.no_jsrequest/do_request.jsp" method="get">
27                 <select name="interestdate">
28                     <option value="{\$dato3}" selected="selected"><xsl:value-of
29 select="{\$dato3}"></xsl:value-of></option>
30                     <option value="{\$dato2}"><xsl:value-of select="{\$dato2}"></xsl:value-
31 of></option>
32                     <option value="{\$dato1}"><xsl:value-of select="{\$dato1}"></xsl:value-
33 of></option>
34                 </select>
35             </td>
36             <td>
37                 <input type="hidden" name="id" value="{\$bogid}"/>
38                 <input type="submit" value="Bestil"/>
39             </td>
40             </form>
41         </tr>
42     </xsl:for-each>
43 </xsl:template>
44 </xsl:stylesheet>

```

Filen startes med de sædvanlige metadata, så som dokumentdeklarationer, og henvisninger til forskellige standarder.

Dernæst sættes en række parametre med `<xsl:param name="minParameter">` elementer som giver mulighed for at sende parametre til en template. Parametrene kan så kaldes med `minParameter` og bruges i XSLT outputtet. Parametrene fungerer altså som en form for variable. Det specielle ved variable i XSLT er, at de er konstante. Når de først har fået tildelt en værdi kan denne ikke ændres igen.

Så er vi klar til at gå i gang med hjertet af XSLT. Først laves en match (linje 10), på hvilke dele af XML dokumentet, der skal transformeres. Her kommer teknologien XPath ind i billedet. "skråstreg" (slash) i matching betyder at hele XML'en matches.

I linje 11 sker den endelige match sted. Her startes et loop, der løber for hvert XML tag/element, der ligger inden i "stien" (angivet med XPath) shortrecord/RDF/Description.

#### XML eksempel. De tags som XPath viser stien ned igennem

```
<shortrecord>
  <rdf:RDF xmlns:dc="http://purl.org/dc/elements/1.1/" >
    <rdf:Description rdf:about="http://www.ilt.bristol.ac.uk/people/cmdjb/">
      <dc:title>Java : An introduction to computer science and programming</dc:title>
```

Hvis ikke XML er opbygget præcist som beskrivelsen angiver, vil det ikke blive fundet. Det repræsenterer en af de væsentligste fejlkilder, når man arbejder med XSLT. Derfor er det vigtigt at have styr på XPath, for at få sine XSLT-filer til at virke.

#### 5.6.5 Vise materialets status

Det er væsentlig for en bestillingsfunktion, at vise om materialet er hjemme, om det skal bestilles eller findes på en anden location. Min JavaScript uafhængige version bygger på den allerede eksisterende funktion. Jeg har kun ændret ved selve præsentationen. På den oprindelige side blive materialernes status vist med farvekoder og status bliver hentet ind med AJAX, og opdateret efterhånden som resultaterne løber ind.

Når status skal hentes ind på siden uden mulighed for løbende opdateringer, tager det længere tid at loade siden. Men funktionen er så væsentlig, at den er værd at vente på.

Denne funktion er altså et godt eksempel, hvorfor det er fornuftigt at bruge AJAX på den oprindelige side i stedet for at vælge en "lavteknologisk" løsning. Opdateringer med AJAX i baggrunden giver brugerne en langt hurtigere og mere flydende oplevelse af søgefunktionen på siden.

#### Filnavn: short\_record\_show.xsl. linje: 212

```
1  <xsl:choose>
2    <xsl:when test="message:getAvailabilityCode($record_id) =
3      'availableForRequest'" >
4      Hjemme
5    </xsl:when>
6    <xsl:when test="message:getAvailabilityCode($record_id) = 'fjernlaan'" >
7      Fjernlån
8    </xsl:when>
9    <xsl:when test="message:getAvailabilityCode($record_id) =
10     'needReservation'" >
11     Skal bestilles
12   </xsl:when>
13   ....
14   <xsl:otherwise>
15     Status ukendt
16   </xsl:otherwise>
17 </xsl:choose>
```

Ovenstående XSLT eksempel viser en simpel kontrol struktur, hvor jeg tjekker på det navn for status, som den underliggende service leverer. Navnet bliver så oversat til dansk. Til dette bruges XSLT-syntaksen `xsl:choose`. Denne tillader i modsætning til `xsl-if`, at man kan definere en default værdi, hvilket svarer til en "else" i de fleste andre programmeringssprog.

Ideelt set burde sådanne oversættelser foregå i en "property fil", hvor man så henter oversættelsen ind afhængig af browseren landekode. Det vil gøre koden mere overskuelig og lettere at vedligeholde.

Jeg har udelukkende valgt at gøre det i XSLT'en udelukkende for at få lov til at eksperimentere med nogle af de kontrolstrukturer, som XSLT stiller til rådighed.

### 5.6.6 Bestille materialer i kurven

Selve bestillingen og login foregår gennem allerede definerede funktioner fra den originale side. Det som jeg har skullet tage højde for, er at få fat i de rigtige parametre, og sende dem med. Samt at sørge for at brugerne er loggede ind før de foretager en bestilling, og så at fjerne de bestilte materialer fra kurven efter en vellykket bestilling.

En af de parametre, der skal sendes med er i formatet JSON. JSON står for JavaScript Object Notation.

JSON er et letvægts (The Fat-free Alternative to XML) data udvekslingsformat. Det er tekstbaseret og programmeringssprogsuafhængigt, men bruger konventioner som man kender fra programmering i den såkaldte C-familie (bl.a. C, Java, JavaScript, Python). Og så er det let umiddelbart at læse for mennesker. Det bruges hovedsageligt, hvor det er for besværligt at sætte en XML op.

JSON formattet er kompatibelt med en lang række programmeringssprog som f.eks. JavaScript, Java, Python, C, C#, C++, PHP 5.2, Ruby etc.

**Fil: do\_request.jsp. Linje: 1**

```
1 SummaUser user = (SummaUser) session.getAttribute("summa_user");
2 String userId;
3 if (user == null){
4     %>
5     <a
href="https://www.statsbiblioteket.dk/casext/?service=http%3A%2F%2Flocalhost%3A8090%
2Fbasket_noJS%2Flogin.jsp">Log ind for at bestille materialerne</a>
6     <%
7 }else{
8     userId = user.getUsername();
9     if(fjernId != null){
10        json = "{\"id\": [" + bogId + "], 'position': [1], 'extra': ['extras_text']}";
11        Thread thread = new DoRequestThread(userId,location,ip,usertype,json);
12        thread.setDaemon(true);
13        thread.start();

14        response.sendRedirect("http://localhost:8090/basket_noJS/index.jsp?fjernid="+
fjernId);
15    }
16    if(bestil != null){
17        for(int i=0; i < kurvindhold.size() ; i++){ // kurven loopes igennem
18            id = kurvindhold.get(i); //bogid hentes ud fra kurven
19            json = "{\"id\": [" + id + "], 'position': [1], 'extra': ['extras_text']}";
20            Thread thread = new DoRequestThread(userId,location,ip,usertype,json);
21            thread.setDaemon(true);
22            thread.start();
23            out.print("json" + i + ": " + json + "<br/>");
24        }
25        response.sendRedirect("http://localhost:8090/basket_noJS/index.jsp?bestil="+
bestil);
26    }
27 }
```

I de første 6 linjer undersøges det, om brugeren er logget ind. Hvis det ikke er tilfældet skrives et link, der fører til login siden. Bemærk at man i JSP ligesom i PHP kan springe ud og ind af scriptet og skrive html tags inde i de opstillede kontrolstrukturer.

Hvis brugeren er logget ind hentes brugernavn. I linje 9 undersøges om parameteren "fjernId" findes. Hvis det er tilfældet, betyder det at bestillingen er en enkeltbestilling. En JSON streng bygges op med bogens unikke ID, samt et par andre data, som p.t. er konstante. Der er bare gjort plads til ekstra parametre af hensyn til den fremtidige udvikling, hvor man regner med at skulle bruge dem.

I linje 11-13 foretages selve bestillingen, før brugeren i linje 14 bliver sendt tilbage til kurven, hvor parameteren "fjernId" angiver, at materialet skal fjernes fra kurven.

I linje 16 undersøges om parameteren "bestil" er sat. Hvis det er tilfældet betyder det, at alle materialerne i kurven er bestilt på samme tid. Her foretages samme bestilling og oprettelse af JSON strenge. Forskellen er at denne gang loopes kurven igennem, for at få fat i de enkelte materialer. Parameteren "bestil" sendes med, når man returnere til kurven, for at angive at alt indholdet i kurven, nu skal slettes.



### 5.6.7 Manipulere datoer i Java

For at give brugeren mulighed for at angive, hvor lang tid han vil være interesseret i at vente på en reservering, er det nødvendigt at finde dags dato og ændre den til en dato i fremtiden. I projektet har jeg lavet 3 foruddefinerede valg, på 3 måneder, et halvt eller et helt år. Her er et eksempel på hvordan, jeg lægger 3 måneder til dags dato.

**Filnavn: basket.jsp. Linje: 37**

```

1 Date dagsdato = new Date();
2 SimpleDateFormat datoFormat = new SimpleDateFormat("dd-MM-yyyy");
3 Calendar kalender = Calendar.getInstance();
4 kalender.setTime(dagsdato);           // sætter dato ind i kalender objektet
5 kalender.add(Calendar.DATE, 90);     // lægger 90 dage til
6 Date udloeb = kalender.getTime();    // henter den nye dato
7 String dato = datoFormat.format(udloeb);


```

Først sættes dags dato med Date objektet. Dernæst sættes en regel op for formatering af datoen til det ønskede format. Til dette bruges SimpleDateFormat objektet.

I linje 3 oprettes en instans af kalender objektet, og derefter sættes dagsdato ind i kalender instansen og i linje 5 lægges der 90 dage til dags dato.

Dernæst hentes den nye dato ud af kalenderen, og formateres efter den opskrift, som jeg lavede i linje 2. Datoen gemmes i en streng, klar til at blive sendt videre.

På den eksisterende side kan brugeren, ud over de foruddefinerede datoer, også selv vælge en dato ud fra en kalender. Dette kan være en god service, hvis man f.eks. ønsker at fastlægge interesse datoer ud fra f.eks. en specifik eksamensdato. Omvendt komplicerer det også brugergrænsefladen. Det vil tage længere tid for brugeren at vælge en bestemt dato fra en kalender, frem for fra en liste af muligheder.

 **STATSBIBLIOTEKET**
Sitemap

---

Du er her: Forside → Søgning

Bestillinger (0) [Vis](#)
Niels Gamborg

Materialer i kurven (3)

Titel	Forfatter	År	Type	Fjern fra kurv	Ikke interesseret efter	Afhentningssted	Bestil
Java : an Object First Approach	Fintan Culwin	1998	bog	<a href="#">Fjern fra kurv</a>	16-05-2009	Statsbiblioteket	<input type="button" value="Bestil"/>
Java : An introduction to computer science and programming	Walter Savitch	2001	bog	<a href="#">Fjern fra kurv</a>	16-05-2009	Statsbiblioteket	<input type="button" value="Bestil"/>
Java : the garden of the East	Eliza Ruhamah Scidmore	1985	bog	<a href="#">Fjern fra kurv</a>	16-05-2009	Statsbiblioteket	<input type="button" value="Bestil"/>

**Søgning på: java**

1 til 20 af 10898. Forrige [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) ... [545](#) [Næste](#) >

Sortér efter: [År](#) | [Relevans](#) | [Titel](#)

Titel	Forfatter	År	Type	Bestil
Hjemme <a href="#">Krama</a> : Billede fra Java	Hans Theilade	1942	bog	<a href="#">Tilføj</a>
Fjernlån <a href="#">Java</a> : An introduction to computer science and programming	Walter Savitch	2001	bog	<a href="#">Tilføj</a>
Fjernlån <a href="#">Java</a> : an Object First Approach	Fintan Culwin	1998	bog	<a href="#">Tilføj</a>
Hjemme <a href="#">Java</a> : the garden of the East	Eliza Ruhamah Scidmore	1985	bog	<a href="#">Tilføj</a>
Lån for <a href="#">Java</a> : buddhistische und brahmanische institutter. Architektur und Plastik auf Java	Karl With	1922	bog	<a href="#">Tilføj</a>

**Afgræns din søgning**

Forfattere

- [the pennsylvania state university citeseer archives \(7992\)](#)
- [ebrary inc \(59\)](#)
- [massachusetts institute of technology dept of electrical engineering and computer science \(40\)](#)
- [safari tech books online \(22\)](#)
- [melvill van cambee o pieter baron 1815 1856 algemeene atlas van nederlandsch indie \(22\)](#)

Den færdige kurv.

### ***5.7 Status over JavaScript uafhængig kurv***

For at kunne sætte kurven i drift mangler projektet stadig noget finpudsning. F.eks. mangler der overordnet at, der bliver taget nogle beslutninger om, under hvilke forudsætninger man kan bruge en sådan kurv i en større sammenhæng. Det er nødvendigt at finde den rigtige løsning til, hvordan man får denne alternative side implementeret på Statsbibliotekets hjemmeside, uden at tilføre de "almindelige" brugere ekstra gener.

Endvidere er kurven bygget oven på den eksisterende Summa udvikling. Derfor slæber den en mængde ubenyttede funktioner og hjørner med sig, som gør den uhensigtsmæssig stor og uoverskuelig.

Samtidig arbejdes der stadig på højtryk med at færdiggøre Summa, og i denne proces har hverken Summa eller præsentationen og funktionalitet af kurv og søgeresultater, fundet sin endelige form endnu.

Endelig mangler kurven at blive brugertestet. Holder det som jeg kalder den nødvendige funktionalitet også i virkeligheden, når den bliver udsat for virkelige brugere.

## 6 Tilgængelighed – en afsluttende diskussion

### 6.1 Hvorfor kan det være svært at opfylde kravene?

På trods af, at Statsbiblioteket egentlig er rigtig godt med i forhold til langt de fleste krav, sidder jeg alligevel med en fornemmelse af, at det i mange tilfælde kan være vanskeligt at opfylde de krav, der er til tilgængelighed. Det udspringer nok af oplevelsen af, at den viden der er omkring tilgængelighedsproblematikkerne både hos mig selv og mine kolleger virker lidt overfladisk eller tilfældig. Samtidig viser resultaterne fra Bedst på Nettet, at der bliver ved med at være problemer med visse af tilgængelighedskravene.

Selvom mange af kravene er kendt af de fleste udviklere, så fylder de ikke så meget i det daglige udviklingsarbejde. Der er et væld af andre krav at tage hensyn til. Der er mange interessenter i udviklingsarbejdet, der er deadlines, der skal overholdes og der er altid knaphed på ressourcer.

Derudover er der et væld af tekniske problemer, som skal løses. Der er kravspecifikationer, som skal overholdes, der er forskellige systemer og platforme, som skal kunne tale sammen og der er pt. tre meget forskellige browsere, som grænsefladen skal fungere i.

Så tilgængelighedskravene kan overfladisk betragtet virke som en mængde ekstra krav, der kun tilfredsstiller et behov hos en ganske snæver brugergruppe, og samtidig er tidskrævende at opfylde.

Oven i alt dette er der ikke meget prestige i arbejdet med tilgængelighed. Målgruppen virker som nævnt smal, og har kun begrænset indflydelse.

Det er interessant, at når man dykker ned i problematikkerne, viser det sig, at man kan løse rigtig mange af problemerne med ganske få midler. Det handler ofte bare om at træffe de rigtige beslutninger på de rigtige tidspunkter. Når det så alligevel nogle gange glipper, tror jeg, at det skyldes, at tilgængeligheden ligger langt tilbage i rækken af hensyn, der skal tages. Det bliver med andre ord meget let overset.

Et andet spørgsmål, der rejser sig er, hvorfor man ikke bare lave lavteknologiske løsninger fra starten? Så ville vi have sikret tilgængeligheden for en masse mennesker på en gang. Den argumentation holder også et stykke af vejen, men hvis vi med de nyeste teknologier kan give øget funktionalitet og bedre brugervenlighed til hovedparten af brugerne, skal det udnyttes. Den generelle udvikling trækker også teknologien på tilgængelighedsområdet med sig, så brugere med funktionsnedsættelser også får glæde af udviklingen med en vis forsinkelse.

### 6.2 Hvordan kan man få mere fokus på tilgængeligheden?

Som tidligere nævnt har Bedst på Nettet allerede bidraget væsentligt til at få sat fokus på tilgængeligheden. Offentliggørelse af resultaterne, har haft stor betydning for især dem der har klaret sig godt. De har kunnet bruge det til at profilere sig overfor både borgere og andre offentlige virksomheder.

Man kan også håbe på at lovgivningen omkring de åbne standarder, vil øge fokus på tilgængelighedsarbejdet. Det bliver nødvendigt at have tilgængeligheden med i målsætningen fra starten, og medfører forhåbentligt, at der vil blive afsat den nødvendige tid og ressourcer til arbejdet. Med lovgivning følger også helt specifikke krav, som gør arbejdet mere håndgribeligt for udviklerne.

Et alternativt initiativ jeg kunne forestille mig er et udvidet samarbejde med brugere med funktionsnedsættelser. Der er i webudviklingskredse efterhånden en god forståelse, af at brugertests er nødvendige. Hvis man kan få samme erkendelse i forhold til test af brugere med

funktionsnedsættelser, vil det være med til at konkretisere kravene yderligere for både udviklerne og beslutningstagere. Jeg vil postulere, at forståelsen, for de behov brugerne med funktionsnedsættelse har, vil vokse, når man med egne øjne har set, hvilke problemer de reelt set kæmper med.

### **6.3 Tilgængelighed i private virksomheder**

Jeg har i rapporten udelukkende beskæftiget med tilgængeligheden i offentlige virksomheder. Men tilgængelighedsarbejde har selvfølgelig også relevans for den private sektor.

Dels er visse af målgrupperne meget store, som tidligere beskrevet. Derfor ligger der et stort forretningspotentiale i disse grupper. Og jo dårligere ens konkurrenter håndterer tilgængeligheden, jo mere rentabelt vil det være for virksomheden at gøre noget ud af området. Virksomheden må så konkret afveje, hvor mange potentielle kunder, der kan hjælpes med forskellige former for tilgængelighedstiltag.

Som minimum bør virksomhederne benytte sig af de brugervenlighedsaspekter, der også øger tilgængeligheden, netop fordi det gavner alle brugere bredt, og man dermed slår to fluer med et smæk.

Et andet aspekt i forbindelse med tilgængelighed, der kunne være interessant for virksomhederne, er deres image. Virksomheder, der gerne vil fremstå som socialt ansvarlige og som værende i besiddelse af medmenneskelige værdier, kan med fordel gøre noget ud af tilgængeligheden, og så diskret gøre opmærksomhed på, at de tager tilgængelighed alvorligt.

## 7 Konklusion

*Er alle tilgængelighedskrav relevante for statsbiblioteket at indfri? Hvad gør Statsbiblioteket for at indfri kravene?*

Mange af tilgængelighedskravene er meget specifikke i forhold til bestemte teknologier, som Statsbiblioteket ikke benytter og har selvfølgelig derfor ingen relevans for Statsbiblioteket at indfri.

Den del af kravene der vedrører brugervenlighed, har allerede stort fokus i Statsbibliotekets arbejde. De har gennemført flere grundige brugerundersøgelser for at afdække, hvad deres brugere har behov for.

De mere specifikke tekniske krav har ikke den store fokus i det daglige arbejde. CMS systemet Plone løser en del af problematikkerne og andre løses "på rygraden". Egentligt fokus har området ikke. Der bliver ikke foretaget brugertest med brugere med funktionsnedsættelser.

*Er der nogen sammenhæng mellem traditionel brugervenlighed og tilgængelighed? Øger den generelle brugervenlighed hjemmesidens tilgængelighed? Kan der være konflikter mellem brugervenlighed og tilgængelighed.*

Det viser sig, at der er en stor sammenhæng mellem tilgængelighed og brugervenlighed. En del af de formelle krav til tilgængelighed beskæftiger sig med brugervenlighedsrelaterede områder som overskuelighed, navigation, konsistens og læsevenlighed. Endvidere er brugervenlighed isoleret set det område, der gavner flest brugere med funktionsnedsættelser.

Der kan dog også findes konflikter mellem brugervenlighed og tilgængelighed. F.eks. vil for mange links, der henviser til tilgængelighedsvenlige funktioner, være med til øge støjen og mindske overskueligheden på siden. Der findes ingen lette løsninger på sådanne problematikker, men første skridt på vejen er at erkende dem, så man kan foretage den fornødne vægtning af argumenterne.

*Hvordan håndterer et moderne CMS system som Plone krav til tilgængelighed? Kan man leve op til kravene om tilgængelighed ved at køre et CMS, som Plone out of the box, uden at skulle tilpasse systemet?*

Plone lever op til rigtig mange af kravene til tilgængelighed. Plone skulle efter sigende også være et af de bedre CMS på området, men der er i de seneste år sket en kraftig forbedring af CMS'ernes tilgængelighed. Det er sket i takt med at systemerne er blevet mere avancerede, og grundfunktionaliteterne er kommet på plads, så CMS'en evne til at levere tilgængelighed er blevet en konkurrenceparameter.

Det er dog vigtigt at have i baghovedet, at CMS selvfølgelig kun beskæftiger med den tekniske tilgængelighed. Ansvar for hele brugervenlighedsområdet ligger stadig hos designere, redaktører og udviklere.

Endvidere er en væsentlig pointer i eksperternes udsagn, at både tilgængelighed og brugervenlighed skal testes. Kravspecifikationer og brugervenlighedsregler kan løse mange problemer, men vil ikke kunne forhindre alle u hensigtsmæssigheder.

*Statsbibliotekets kurv er baseret på brug af JavaScript og AJAX. Hvad betyder det for tilgængeligheden? Kan samme funktionalitet opnås uden brug af JavaScript?*

Avanceret JavaScript og AJAX udgør et reelt problem for tilgængeligheden. Heldigvis har det vist sig, at det har været muligt at opnå næsten samme funktionalitet uden brug af disse teknologier. I denne konkrete sag vil jeg vurdere, at en rutineret udvikler på 14 dage vil kunne lave et fuldt funktionsdygtigt modul, der er uafhængigt af JavaScript. Dette må siges at være

ganske lille investering, set i lyset af at udviklingen af hele Summa har taget 4-5 mand omkring 2 år.

På et mere overordnet plan forventes skærmlæsernes problemer med avanceret JavaScript og AJAX, at blive løst i løbet af et til to år. Der arbejdes fra forskellige udvikleres side på at løse disse problemer, og W3 er på vej med en række standarder på områder.

Dette specifikke problem er et godt eksempel på, at teknologi udviklet til brugere med funktionsnedsættelse altid vil halte efter den almindelige teknologiske udvikling. Det betyder, at når de nuværende problemer med JavaScript og AJAX bliver løst, vil der sandsynligvis dukke en række nye problemer op.

*Hvad kan man gøre for at styrke incitamentet til at arbejde mere med tilgængelighed? Hvad bliver der gjort fra statens side?*

For mig at se handler det om at få en større fokus på tilgængeligheden. Første skridt kunne meget vel være den nye lovgivning på området, som er med til sætter fokus på problemerne på ledelsesniveau. Næste skridt er så at formidle de strategiske overvejelser ned på udviklerniveau. Gode initiativer kunne være en øget brugerinddragelse. Det giver alt andet lige et større engagement og forståelse for problematikkerne, når man har set konkrete brugeres konkrete problemer på nært hold.

Bedst på nettet er et godt eksempel på et statsligt initiativ, der har højnet tilgængeligheden på et væld af danske hjemmeside. Konkurrencemomentet har gjort at mange forskellige offentlige institutioner har brugt flere ressourcer arbejdet med tilgængelighed. Succesen skyldes formentlig også en samfundsudvikling, hvor begreber som branding og profilering har bredt sig til den offentlige sektor. Bedst på nettet har givet de enkelte institutioner mulighed for at promovere sig selv.

Set i et større perspektiv, vil de ressourcer, der bruges på tilgængelighedsområdet være afhængige af samfundskonjekturene. I et overskudssamfund er der altid større vilje til at tage sig af svagere borgere.

## 8 Litteraturliste

### **8.1 Bøger**

- Birkvig, Henrik  
2004, 'Sådan laver du smukkere grafisk design' (Grafisk Litteratur)
- Birkvig, Henrik  
2007, 'Grundbog i Typografi' (Grafisk Litteratur)
- Christensen, Susie  
2003, 'Handicapvenligt webdesign' (Libris)
- Christiansen, Poul Erik  
Kjær, Henrik  
Skriver, Hans Jørgen  
Staubstrup, Erik  
2005, 'Organisation' (Trojka)
- Grandahl, Benny Christensen  
Simon, Martin  
2003, 'Webkommunikation' (Kroghs Forlag)
- Hanna, Phil  
2003, 'JSP 2.0 The Complete Reference' (Osborne)
- Krug, Steve  
2006, 'Don't make me think' (New Riders Publishing)
- Molich, Rolf  
2006, 'Brugervenligt Webdesign' (Nyt Teknisk Forlag)
- Krug, Steve  
2006, 'Don't make me think' (New Riders Publishing)
- Nordfalk, Jacob  
2004, 'Webprogrammering med JSP' (Forlaget Globe)
- Nordfalk, Jacob  
2005, 'Objektorienteret programmering i Java' (Forlaget Globe)
- Pedersen, Kim  
2002, 'Sttram op på dit Webdesign' (Forlaget Grafisk Litteratur)
- Sebesta, Robert W.  
2005, 'Programming the World Wide Web', (Pearson/Addison Wesley)
- Tidwell, Doug  
200, 'XSLT' (O'Reilly)
- Wolff-Toft, Morten  
Poulsen, Ole Vestergaard  
2007, 'Innovationsrecepten(side - )' (Jyllands Postens Forlag)

## **8.2 Internet ressourcer**

### **Om statsbiblioteket**

Statsbibliotekets historie:

<http://www.bibliotekshistorie.dk/sektion02.tekst.lasso?readthis=13&-token.skiprecords=3>

Statsbibliotekets årsberetning 2007

<http://www.statsbiblioteket.dk/om-statsbiblioteket/resolveuid/265a99abd4998e45f7d8c8ee354c04c8>

### **Åbne standarder**

Vejledning om åbne standarder:

[http://www.itst.dk/regeringens-it-og-telepolitik/abne-standarder/Vejledning\\_om\\_abne\\_standarder.pdf](http://www.itst.dk/regeringens-it-og-telepolitik/abne-standarder/Vejledning_om_abne_standarder.pdf)

Aftale om anvendelse af åbne standarder for software i det offentlige:

<http://videnskabsministeriet.dk/site/forside/nyheder/Aktuelle-temaer/aabne-standarder/bilag/bilagsfil/aftale.htm> og den medfølgende vejledning fra IT-og Telestyrelsen: <http://www.itst.dk/regeringens-it-og-telepolitik/abne-standarder>

### **Retningslinjer for tilgængelighed**

Indgang til Web Accessibility Guidelines (WAI)

<http://www.itst.dk/kommunikation-og-tilgaengelighed/tilgengelighed/tilgengelighed-i-praksis/kom-videre-med-tilgengelighed/indgang-til-wai>

W3's anbefalinger: <http://www.w3.org/TR/WAI-WEBCONTENT/>

Udbudsværktøjskassen:

<http://vkassen.oio.dk/omudbudsvaerktoejskassen/>

Bedst på nettet:

<http://www.bedstpaanettet.dk/vurderingsgrundlag%202007>

W3.org portal til anbefalinger til webtilgængelighed (Web Accessibility Initiative WAI)).

<http://www.w3.org/WAI/> [W3.org](http://www.w3.org)

Beregning af farve og styrkekontrast jvf. W3.org:

<http://www.w3.org/TR/AERT#color-contrast>

### **Brugervenlighed**

Jakob Niensens Alertbox, November 21, 2005: Accessibility Is Not Enough

<http://www.useit.com/alertbox/accessibility.html>

Jakob Niensens Alertbox, March 14, 2005: Lower-Literacy Users

<http://www.useit.com/alertbox/20050314.html>

"Når svage læsere bruger hjemmesiden." En masterafhandling af Peter Iversen

[http://rudar.ruc.dk/bitstream/1800/2634/1/Hjemmesider\\_og\\_svage\\_laesere.pdf](http://rudar.ruc.dk/bitstream/1800/2634/1/Hjemmesider_og_svage_laesere.pdf)



"Flere kunder og bedre service med et tilgængeligt website". Stig Møller Carlsen  
IT Rådgiver, Dansk Center for Tilgængelighed  
<http://www.cmforum.dk/artikler/artikel5>

Brugbarhedsundersøgelser fra Statsbiblioteket  
<http://www.statsbiblioteket.dk/om-statsbiblioteket/brugbarhed>

### **Programmering**

Hjælp, tutorials og referencer til webudvikling generelt  
<http://www.w3schools.com/>

Sun Forums. Hjælp og tips til Java og JSP  
<http://forum.java.sun.com/index.jspa>

XSLT og XPath tutorial  
<http://www.topxml.com/xsl/tutorials/intro/default.asp>

Om JSON formatet  
<http://json.org/>

## 9 Bilag

### ***Bilag 1. De 14 tilgængelighedskrav i teori og praksis.***

#### **Krav 1. Billede og lyd**

##### **Alternativ tekst**

Angivelse af en "ensbetydende" tekst til samtlige ikke tekstlige elementer. Det betyder, at man med tekst skal angive det meningsmæssige indhold, der er i det ikke tekstlige indhold. F.eks. kan en pils "ensbetydende" indhold være "gå til ...".

I praksis gøres det ved altid at angive alt-attributter i sin mark-up af f.eks. billeder.

Hvis det ikke tekstlige element ikke er meningsbærende skal alt-attributten stadig angives men den skal efterlades tom. Dvs. at siden validerer XHTML standarden, men de forskellige oplæsere vil ignorere billedet. Hvis alt teksten ikke er angivet vil skærmoplæseren f.eks. selv forsøge at finde en mening med billedet. Og dette kan give mere forvirring end afklaring.

Alternative tekster skal også angives f.eks. til frames/lframes eller klint-side imagemap. Dette er vigtigt for at kunne overskue webstedet organisation.

##### **Video og Audio**

Endelig bør audio og video tekstliggøres. Jo større betydningsmæssigt indhold de har, jo vigtigere er tekstliggørelsen selvfølgelig.

##### **Statsbiblioteket**

Brugen af grafikker på hjemmesiden er meget begrænset. Man har bevidst fravalgt brug af f.eks. ikoner, for at formindske mængden af støj. Når grafik bruges er det i form af billeder.

Plone sætter default billedets navn som alt-attribut. Brugeren skal så selv ændre alt-attributten til det ønskede. Hvis man sletter indholdet af alt-feltet, vil alt-attributten blive fjernet fra img-tagget. Det er ikke hensigtsmæssigt. Løsning: Redaktører skal gøres opmærksomme på altid at udfylde alt-attributten.

Problemer: Ved søgninger hentes der nogle gange et omslagsbillede fra???. Disse img-tag indeholder ingen alt-tags.

#### **Krav 2. Farver**

##### **Farver**

Farver bør ikke bruges som de eneste betydende elementer i designet. Endvidere skal beskrivelser som "udfyld den røde boks" og lignende. Links skal være understregede, så man uafhængigt af sit farvesyn kan se, at der er tale om links.

##### **Kontrast**

Endvidere skal der bruges nok styrke kontrast til at tekster kan læses og vigtige informationer stadig vil være fremtrædende på trods af evt. nedsat farvesyn.

##### **Statsbiblioteket**

Links på siden er bevidst holdt i Internettets standard farver. Dette er primært gjort for at øge brugervenlighed ved at udnytte de konventioner, der er på Internettet.

The screenshot shows the top part of the Statsbiblioteket website. At the top left is the logo 'STATSBIBLIOTEKET'. To the right are links for 'Sitemap' and 'Dine lån'. Below this is a large, dark, blurred banner image. Underneath the banner is a navigation bar with the text ': Forside'. Below that is a search bar with a 'Søg' button and links for 'Dine lån', 'Ret profil', and 'Ny bruger'. A secondary navigation bar contains links for 'E-bøger', 'Tidsskrifter', 'Databaser', 'Tv og radio', 'Aviser', 'Musik', and 'Udvidet søgning'. On the left side, there is a section titled 'Kontakt og dialog' with links to 'Statsbiblioteket' and 'Mest benyttede numre og adresser'. Below this is a section for 'Nyheder' with a link to 'Nyheder for foråret 2008'. Further down are 'Andre oplysninger' with links for 'Låstider', 'Ny bruger', 'Ekspeditionstid og levering', 'Lånetid og gebyrer', 'Betaling for print', 'Nærdgang til e-ressourcer', 'Fysiske faciliteter', and 'Lån fra andre biblioteker'. On the right side, there is a 'Meddelelse' box containing two items: 'Statsbiblioteket 21. december 2008' and 'Nyhedsbrev til H 19. december 2008'.

Udsnit af Statsbibliotekets forside i gråskala:

Som det ses på billedet lever statsbiblioteket helt op til kravene om tilstrækkelige kontraster. Der hvor der brugt farver i farver er styrke kontrasten så stor at ønsket om at fremhæve teksten stadig bliver opfyldt.

### Krav 3. Opmærk korrekt

#### Valid html

Websidens mark-up, bør følge W3's standarder, og sidens struktur, udseende og funktionalitet skal adskilles ved den rette brug af CSS, XHTML og scripting. Som minimum skal standarden for XHTML 1.0 strict overholdes. (forskellen mellem XHTML 1.0 strict og XHTML 1.1 er minimal):

Ved at bruge korrekt opmærkning opnår man maksimal fleksibilitet og dermed den størst mulige tilgængelighed på flere forskellige platforme.

Med valid kode sikrer man, at flest mulige nuværende og fremtidige hjælpemidler kan bruge dit indhold. Har du adskilt layout, struktur og funktionalitet, giver du brugeren de bedste forudsætninger for at tilpasse indholdet til egne behov, og dig de bedste forudsætninger for at tilpasse til forskellige medier.

Endvidere giver du udviklerne af nye hjælpemidler de optimale muligheder for at skabe nye applikationer.

#### Stylesheets

Stylesheets'ene skal være så simple som muligt og benytte sig af de nedarvede styles, som er en CSS' store styrker. Det giver lettere overskuelig kode og mindre tunge sider, med deraf følgende hurtigere load tider.

#### Relative størrelser

Dimensioner som skal så vidt muligt angives i relative enheder. Dimensioner på siden i % og skriftstørrelser i em. Det giver brugeren mulighed for selv at skrue op og ned for skriftstørrelsen. Op til 200% og ned til 50% bør teksten kunne ændres uden tab af indhold eller funktionalitet. Endvidere vil et fleksibelt layout give bedre mulighed for at bruge andre klienter end traditionelle browsere.

### **Overskrifter**

Overskrifter skal bruges til at give strukturer og niveauer på siden. Man skal følge XHTML logiske opsætninger så øverste overskrift er h1 og derefter lavere niveauer. Overskrifter skal ikke kun bruges på de enkelte sider, men på strukturen som helhed. F.eks. kan menuer opmærkes under et niveau og kontaktoplysninger under et andet. Det er med til at angive de enkelte elementers betydning for skærmlæsere, hvor den visuelle opbygning af et hierarki, ikke kan ses. Overskrifter man bruger på denne specifikke måde, kan skjules for almindelige brugere. (<http://www.borger.dk> eksempel på tab navigation med usynlige elementer)

### **Citater**

Citater er ikke brugt så ofte, men når de optræder skal man være opmærksom på at opmærke dem korrekt. Dvs med q-tags til korte citater og blockquote-tags til længere.

### **Statsbiblioteket**

Statsbibliotekets hjemmeside valider ikke. Plone skulle validere XHTML 1.0 transitional "out of the box". Men på grund af tilpasninger har der indsneget sig nogle fejl.

De første fejl skyldes noget kode der skal hente det aktuelle sprog ind fra browserens header informationer. Dette glipper tilsyneladende, og den satte default værdi på engelsk overskrives af "ingenting". Problemet er i småtingsafdelingen, og et eksempel, hvordan nogle

De sidste fejl handler om et forkert afsluttet ul-tag og en ID, der optræder to steder på den samme side.

Stylesheets'ene er store, ukommenterede og i mange tilfælde fyldte med redundans. Den manglende kommentering har ikke direkte indflydelse på tilgængeligheden, men betyder at optimering af stylesheets'ene bliver meget tidskrævende og derfor vil der være mindre tilbøjelighed til at optimere disse.

### **Dimensioner**

Der bliver i Plone og dermed på statsbiblioteket brugt 2-3 overskriftsniveauer. Dette ser ud til at fungere i forhold til anbefalingerne på sideniveauet. Anderledes ser det ud på det overordnede strukturelle niveau. Der er ikke anvendt skjulte overskrifter til at angive menuer etc placering i hierarkiet, og der er enkelte overskrifter, der ikke giver mening for den udefra kommende bruger. Nogle af overskrifter (Her er der nogle problematikker der er relevante at diskutere. Overskrifthevede kontra brugervenlighed for alm brugere og webredaktører)

## **Krav 4. Angiv benyttet sprog**

Det angivne sprog skal angives. Både for at validere sine sider, men også for at give brugere med skærmlæsere mulighed for at kende hovedsproget, men også for at gøre opmærksom på, hvis hoved sproget skifter f.eks. i en enkelt sektion eller på en side.

Når hovedsproget ændres i et dokument, kan talesynthesizere og/eller punktskriftenheder automatisk skifte til det nye sprog og dermed gøre dokumentet bedre tilgængeligt for flersprogede brugere.

### **Statsbiblioteket**

Statsbiblioteket angiver hovedsproget på deres website. Men på grund af nogle scripting problemer, bliver sproget fjernet igen, fra den udgave brugeren ser.

Dette er problematisk i forhold til ovenstående, men også i forhold til validering af hjemmesiden.

Primært for at betjene udenlandske undervisere og studerende, har statsbibliotek.dk en lille engelsksproget sektion med praktiske oplysninger. Her opleves de samme problemer som beskrevet ovenfor.

## Krav 5. Tabeller

### Data-tabeller

Tabeller udgør ofte et stort problem for brugere af skærmlæsere. Især større og komplicerede tabeller med mange data udgør et problem. Dette skyldes blandt andet at tabeller visuelt præsenterer store mængder data for brugeren. Det overblik, som tabellen giver den almindelige bruger, forsvinder, når indholdet af de enkelte tabelceller bliver læst op, uden at man hurtigt kan orientere sig visuelt om hvor man er i tabellen, og hvordan datene hænger sammen

For at bøde på denne naturlige begrænsning er det vigtig at man bygger sine tabeller op efter W3's standarder. F.eks. kan man med simple tabeller benytte tags som thead, tbody og tfoot for, at angive tabellens overordnede struktur. Benytte en caption eller summary til at angive tabellens indhold.

Ved mere komplicerede tabeller, hvor f.eks. overskrifter dækker over andre overskrifter og over flere søjler eller rækker, kan man arbejde med headers og id. Det foregår på den måde, at hvert th-tag får en id, og hver td får en header. Så kan skærmlæseren relatere indholdet af en celle til en overskrift.

Ved komplicerede tabeller er det ekstra vigtig at benytter sig af at lave nogle gode og grundige beskrivelser med enten caption eller summary.

Det kan være et meget stort arbejde, at gøre manuelt, men man kan benytte sig af diverse online værktøjer til det, f.eks. <http://www.accessify.com/tools-and-wizards/accessibility-tools/table-builder/> som kan hjælpe med at bygge både simple og komplicerede tabeller efter forskrifterne.

### Layout tabeller

Layout tabeller udgør i dag kun et mindre problem for skærmlæsere, som er blevet bedre til ikke at springe mellem indholdet i forskellige spalter. Men af andre årsager er det optimale at benytte sig af CSS til positionering i stedet for tabeller.

### Statsbiblioteket

Statsbiblioteket benytter sig kun i mindre grad af tabeller. Tabellerne er små og simple og lette at overskue. Plone sætte automatisk tags som tbody, men thead benyttes ikke. Caption eller summary benyttes ikke af Plone.

Det eneste sted, hvor statsbiblioteket pt viser en større mængde data er i søgningerne. Men Summa, som er statsbibliotekets egen udviklede søgemaskine viser ikke resultatet af søgninger i tabelformat.

Kan man evt programmere en pakke i Python, som sætter en thead ind i tabellen, hvis table headings er afkrydset i Plone CMS'en?

Plone benytter sig i en hvis udstrækning af layout tabeller. Idet sites 2 spalter er lavet i tabeller. Dette kan ikke ændres, da der ligger en overordnet beslutning om at kører Plone så vidt muligt "out of the box". Dette skulle dog heller ikke udgøre et problem, da moderne skærmlæsere alle understøtter simple lay-out tabeller.

## Krav 6. Nye teknologier

### Uden CSS

Skærmlæsere bruger ikke de positioneringer, der er sat i CSS'en, men læser op fra det flow, der er i HTML'en. Derfor kan det give problemer, hvis man på et websted placerer "fremmede elementer" inde i flow'et. Det kunne f.eks. Være en fakta box eller lignende, som er placeret på siden, med en float. En sådan boks skal i stedet placeres sidst i HTML flowet. Lignende hensyn kan man tage ved placering af øvrige elementer, så man f.eks. placere de væsentligste elementer først i flow'et og de mindre vigtige til sidst på siden.

Plone løser denne problematik ved at have nogle "accesskeys" lagt ind på siden, så brugere med skærmlæsere, kan springe direkte til det indhold, som de leder efter. F.eks. kan de springe menuer over.

En let måde at kontrollere på om webstedet lever op til ovenstående er ved at slå CSS'en i visningen.

### Alternativer til dynamisk indhold

Alternativt indhold til dynamisk opdateret indhold, skal udskiftes i samme takt, som det dynamiske indhold bliver opdateret. F.eks. ved en automatisk opdateret billedserie, skal det alternative indhold opdateres i samme takt, som billederne skifter.

Diskussion om rimeligheden i dette. Vil det ikke være mere brugervenligt for skærmlæserbrugeren at angive i starten, at nu starter et lysbilledshow, som indeholder følgende elementer. Der er måske ingen grund til at brugeren i denne situation skal vente f.eks. 10 sekunder på en opdatering af indholdet. En seende bruger vil selvfølgelig kunne bruge de 10 sekunder på at se detaljer i billedet, hvor skærmlæserbrugeren ikke vil have samme behov for at fordøje en oplysning om, at der nu vises et billede af en fiskekutter i Esbjerg havn.

### Teknologi neutral

Begrebet teknologineutral dækker over flere ting. F.eks. at siden kan bruges selv når teknologier som f.eks. JavaScript ikke virker. JavaScript er fint at bruge, da det kan bidrage med forskellige lækre effekter for den almindelige bruger, men det er vigtigt at navigationen fungerer uafhængigt af det. Det klassiske eksempel er de tidligere så populære dropdown menuer styret af JavaScript. Hvis man stadig bruger denne form for navigation, er det vigtigt, at de enkelte menupunkter også kan tilgås på en anden måde. F.eks. gennem tastaturet.

I det hele taget er det vigtigt at siden kan navigeres med f.eks. tastaturet. Mange brugere har i dag museskader og benytter tastaturet til navigation.

### AJAX

AJAX udgør i dag et reelt problem for skærmlæsere. Skærmlæserne bliver ikke oplyst om, hvor på siden der sker en opdatering, derfor kan brugerne ikke orientere sig på siden. Der arbejdes på højtryk på at løse disse problemer, f.eks. ved at forbedre enten skærmlæserne eller AJAX opmærkningen, så AJAX bliver gjort mere tilgængelig. Dette arbejde foregår selvfølgelig, fordi AJAX teknologien generelt er meget brugervenlig, og derfor optræder på flere og flere websites.

Statens anbefalinger er, at man bør vente med at bruge AJAX til, der kommer nogle løsninger på området, eller at man på sine sider tilbyder alternative sider, der fungerer uden AJAX opdateringer.

### Statsbiblioteket

Statsbibliotekets sider fungerer også hensigtsmæssigt med stylesheetet slået fra. En af de store fælder nemlig indsatte faktabokse, er undgået ved at disse ligger i såkaldte portletter i højre spalte, der er positioneret ved layout tabeller, som beskrevet under tabeller. Opbygningen er sådan at det væsentlige indhold ("need to know" informationerne) ligger i den centrale spalte, og det mindre væsentlige ("nice to know") ligger i højrespalten.

Det eneste u hensigtsmæssige på statsbibliotekt.dk er, at alle siderne starter med en lang række menuer. Denne u hensigtsmæssighed er dog løst af en indbygget funktion i Plone, der ved hjælp af en "accesskey" sender skærmlæseren videre til indholdet eller til menuen.

Statsbibliotekt.dk bruger ikke dynamisk opdateret indhold, hvor der er for dynamisk opdaterede alternative tekster.

Statsbiblioteket bruger AJAX til at håndtere bestillinger af materiale, der er søgt på. Det betyder at brugere, der ikke har slået JavaScript til i deres browsere, eller brugere der bruger skærmlæsere, ikke kan se eller bestille materialer. Det er et af de større problemer i forhold til tilgængeligheden på hjemmesiden.

W3 initiativer: WAI-ARIA, the Accessible Rich Internet Applications Suite

<http://www.w3.org/TR/wai-aria-roadmap/>

<http://www.w3.org/WAI/intro/aria.php>

## **Krav 7. Tidsafhængige ændringer**

### **Bevægelse**

Tekst kan være vanskeligt at læse. Dette gør sig især gældende for mennesker med læseproblemer eller syns handicap. Brugere med kognitive handicaps kan blive så distraherede af bevægelig tekst/elementer, at de ikke er i stand til at læse resten af siden.

Skærmlæsere er ofte ikke i stand til at læse bevægende tekst.

Og endelig kan mennesker med fysiske handicaps have svært ved at bevæge sig hurtigt nok til at kunne anvende bevægelige objekter.

### **Blink**

Kraftigt blinkende elementer kan være med til at udløse epileptiske anfald hos epileptikere.

### **Statsbiblioteket**

Statsbiblioteket bruger hverken bevægelige eller blinkende elementer på deres hjemmeside

## **Krav 8. Indlejrede objekter**

Indlejrede objekter på en hjemmeside kan være, video, flash eller lydafspilninger. Disse indlejrede objekter skal have samme tilgængelighed som resten af siden. Det vil f.eks. sige at tekster på start og stop knapper skal kunne betjenes ved hjælp af tastaturet, og også kunne aflæses af brugerens hjælpemidler.

Hvis dette ikke kan opnås skal der som minimum stilles et alternativ til rådighed, som lever op til kravene om tilgængelighed, og indeholder tilsvarende informationer.

### **Statsbiblioteket**

Statsbiblioteket bruger ikke indlejrede elementer på deres hjemmeside.

## **Krav 9. Hardware uafhængighed**

### **Image-maps**

Hvis man bruger imagemaps, skal de være klient baserede frem for server baserede, da de sidste generelt skaber problemer.

### **Enhedsuafhængighed**

Det er vigtigt, at alle elementer på brugergrænsefladen kan manipuleres med flere forskellige enheder. Det kan være mus, tastatur, hovedmus, talebaserede browsere.

I praksis er det tilstrækkeligt at elementerne kan tilgås gennem tastaturet, da de så vil være tilgængelige for alle de øvrige enheder.

### **Logisk script**

Scripts skal undgå at bruge enhedsspecifikke eventhandlers, så som on mouse over etc. De fleste moderne browsere understøtter dog også onclick events gennem keyboardet.

### **Statsbiblioteket**

Statsbibliotekets hjemmeside er fuldt navigerbar med tastaturet, inklusive udfyldning af formularer.

## **Krav 10. Midlertidige løsninger**

En række løsninger, der var tænkt som midlertidige, men stadig anbefales af IT og Telestyrelsen for at løse nogle tilgængelighedsproblemer.

It og telestyrelsen citerer WCAG for følgende:

"De følgende kontrolpunkter gælder indtil brugeragenter (herunder hjælpemidlerne) har adresseret disse problemer. Kontrolpunkterne er klassificerede som "midlertidige," hvilket betyder, at Web Content Guidelines arbejdsgruppen finder dem valide og nødvendige for at sikre web-tilgængelighed på tidspunktet for publiceringen af dette dokument."

### **Pop-up**

Pop-up skal helst undgås af flere forskellige årsager. Mange brugere har installeret pop-up blockere, så derfor er det kritisk hvis man placerer væsentlige informationer i pop-up vinduer. For brugere af skærmlæsere er det forvirrende, hvis der pludselig popper nye vindue op.

Hvis man benytter sig af at åbne links i nye vinduer, er det vigtigt, at man tydeligt fortæller brugeren det, og at sikre sig, at brugeren kan lukke vinduet igen ved hjælp af tastaturet.

### **Label**

Det er vigtigt at bruge labels på sine formularer. Det kan være svært for brugere af skærmlæsere sammenhængen mellem f.eks. knapper og inputfelter. Står der f.eks. Søg, efterfulgt af et input felt og derefter af en knap, er sammenhængen tydelig for en normal bruger. Denne sammenhæng kan være svær at se for en blind.

### **Fieldset og legend**

Med fieldset kan man gruppere store og avancerede formularer. Især hvis der er grupper af selects kan det være påkrævet at lave grupperinger. Med legend kan man give grupperingerne navne. Begge dele er til stor fordel for brugere af skærmlæsere.

### **Statsbiblioteket**

Statsbiblioteket har en politik om aldrig at åbne links i nye vinduer. Dette er for at følge W3C anbefalinger og også for at sikre at brugerne selv bevare fuld kontrol over deres browser.

Statsbiblioteket bruger labels på deres brugeroprettelses sider. De bruger det ikke på søgefelterne. Formularerne er små, så der er ikke behov for at gruppere dem med fieldset.

## **Krav 11. W3C teknologier**

### **Anvend W3C teknologier**

Man skal så vidt muligt bruge W3C teknologier, som html og css. Hvis man har tungvejende grunde kan man benytte sig af andre teknologier, men det kræver så at man opretter alternativer i f.eks. html.

Oversigt over W3 teknologier: <http://www.w3.org/TR/WAI-WEBCONTENT/#References>



Et godt eksempel på en udbredt ikke W3C teknologi, er dokumenter i PDF format. Dette er ikke umiddelbart tilgængeligt for skærmlæsere. Denne tilgængelighed kan opnås ved at gøre selve PDF dokumentet, men ellers er det vigtigt at oplysningerne findes andre steder end i PDF dokumentet.

Et andet udbredt eksempel er brugen af Flash, som heller ikke er en W3C teknologi. Her henvises til blandt andet krav 8 - indlejrede objekter og krav 1 - billede og lyd.

### **Alternativ side**

Som nævnt kan man oprette alternative sider, hvis man bruger "ikke-W3C teknologier". Denne udvej er dog det sidste alternativ, da al erfaring viser, at alternative sider meget sjældent benyttes. Og det er vigtigt, at kun enkelte sider ikke hele websteder oprettes som alternativer.

### **Statsbiblioteket**

Statsbiblioteket tilbyder visse oplysninger på PDF format. Disse informationer er ikke tilgængelige i HTML format.

Tjek om PDF'erne er gjort tilgængelige. Spørg Ane Line

## **Krav 12. Kontekst og orientering**

Dette krav handler om overskuelighed og gruppering af sammenhængende elementer på hjemmesiden. En god og overskuelig hjemmeside hjælper folk med læseproblemer og kognitive problemer. Det vil sige, at dette krav egentlig ligger i forlængelse til et brugervenligt websted generelt. Hvis det er velstruktureret er det ofte også brugervenligt og dermed til gavn både for almindelige brugere og brugere med handicap.

### **Frames**

Hver enkelt frame skal navngives, så brugere af skærmlæsere kan se hvor de er henne og hvilken frame der bliver læst op.

### **Overblik**

Udover bruge af kendte design tricks som gestaltlovene og brug af kontraster til at etablere og understrege hierarkier, skal, man benytte HTML's eget indbyggede hierarki. F.eks. gennem en logisk brug af overskrift niveauer. Se endvidere krav 3 – korrekt opmærkning.

### **Statsbiblioteket**

Statsbibliotekets hjemmeside er holdt i et simpelt lay-out uden unødvendig pynt og effekter for at sikre mindst muligt støj. De forskellige elementer er grupperet sammen i logiske blokke, der tydeligt angiver hvad, der hører sammen med hvad. Dette er til gavn for såvel, den almindelige bruger, som brugere med funktionsnedsættelse.

## **Krav 13. Navigation**

Et væsentligt element i den generelle brugervenlighed er overskuelige navigationsmuligheder. Dette gælder selvfølgelig også for tilgængeligheden.

### **Link**

Det er vigtigt at have sigende tekster på sine. Dette er et godt gammelt råd både i forhold til brugervenlighed og søgemaskineoptimering. I forhold til f.eks. skærmlæsere er det ekstra vigtigt, da mange skærmlæsere lister alle links på siden, for at skabe et overblik over navigationsmulighederne på siden. Derfor skal linktekster som: "læs mere her" undgås. Når man f.eks. i forbindelse med nyhedslisters, viser en snippet med tekst fra nyheden, bliver der ofte

linket nedenunder med "læs mere" eller lignende. En måde at undgå det på er at gøre selve overskrifterne til links. Dermed sparer man også tekst på siden, og gør den mere overskuelig.

Et andet godt råd i forhold til at øge tilgængeligheden er at sætte en beskrivende title attribut på links.

### **Metadata**

Et sigende title tag er vigtigt. Både for skærmlæsere og den generelle brugervenlighed.

### **Sitemap**

Et sitemap er nyttigt til at skaffe sig et overblik over både webstedets opbygning og indhold. I forbindelse med beskrivelsen af strukturen bør man også fremhæve og forklare særlige forhold omkring tilgængeligheden.

### **Konsistens**

Det er vigtigt at navigationselementer etc. er konsistente. Det vil sige, at menuer og søgefelter har samme placering og udformning på alle sider.

### **Statsbiblioteket**

Dårlige linktekster optræder nogle enkelte få steder. Der er opfordret i det tidligere nævnte designdokument, til at udelukkende benytte sig af sigende linktekster. Men med mange webredaktører findes der enkelte smuttere.

Plone sætter automatisk en title attribut på interne links. Titlen bliver taget fra den side eller mappe, som linket peger på. Plone giver ikke mulighed for at sætte title attributter manuelt.

Statsbiblioteket har et sitemap, der er generet og vedligeholdt af Plone, og derfor altid er opdateret. Den tilhørende beskrivelse er Plone standard. Den er kort og oversættelsen er bestemt ikke brugervenlig. "En oversigt over indhold på sitet. Beskrivelsen tilknyttet det enkelte indholdsobjekt vises kun, hvis du holder musen over en titel i et par sekunder."

Statsbibliotekets overordnede navigation foregår fra forsiden, og navigationsmulighederne her følger ikke med ud på undersiderne. Undtagelsen er linket til sitemappet, som optræder på alle undersider på samme sted. Tjek log statistik for sitemappet.

## **Krav 14. Forståeligt og enkelt**

### **Klart sprog**

Krav 14 om et klart og forståeligt sprog er måske det allervigtigste krav. Det er til gavn for alle brugere. Men især hjælper den store gruppe af brugere med læsevanskeligheder eller med problemer på det kognitive område.

Nogle generelle råd omkring gode tekster til nettet er:

- Skriv kort og præcist.
- Brug korte sætninger
- Sig det vigtigste først
- Brug mange overskrifter
- Brug mange afsnitsinddelinger

### **Statsbiblioteket**

Men selvom det måske er det vigtigste råd er det måske også det sværeste at overholde i praksis. Især fordi rådene er åbne for fortolkninger. Som en sikring af den skriftlige kvalitet har statsbiblioteket indsat en række overordnede webredaktører, der har det overordnede ansvar for at de tekster lever op til retningslinjerne.

Med moderniseringen af sitet har Statsbiblioteket taget et stort skridt i retning af mere tilgængelige tekster. Tidligere var hjemmesiden meget stor, og med mange specielle artikler om mange specielle emner. Det høje uddannelsesniveau i organisationen afspejlede sig i mange af artiklerne, der nærmere var kendetegnet ved høj akademisk standard frem for letlæselige tekster. Beslutninger om at slanke hjemmesiden, byggede i høj grad på analyse af logfiler, der klart viste, at de mange hjørner af det tidligere website, stort set ingen besøgende havde. Brugere kom og kommer stadig på statsbibliotek.dk for at låne bøger og andet materiale, og ikke af ret mange andre årsager.

Jeg har i min praktik periode udfærdiget et design dokument, som skal fungere som en rettesnor for statsbibliotekets ca. 20 webredaktører. Et af hovedpunkterne i dette dokument er retningslinjer for hvordan man formulere sig brugervenligt, når man skriver til nettet. (se Bilag 2)

## ***Bilag 2. Designdokument for statsbiblioteket.dk***

### ***Om designdokumentet***

Formålet med dette dokument er at give de mange webredaktører på Statsbiblioteker et sæt retningslinjer for formen af det indhold, som de leverer til statsbiblioteket.dk.

Retningslinjerne skal hjælpe til at indholdet bliver så brugervenligt som muligt, samt at statsbibliotekets overordnede designlinje bliver fulgt, så hele hjemmesiden fremstår som et samlet hele, både visuelt og indholdsmæssigt.

### ***Det overordnede design på statsbiblioteket.dk***

Statsbiblioteket.dk skal være et enkelt og informativt site. Brugere skal nemt og hurtigt kunne finde de informationer, de leder efter. Intentionen er at minimere støj ved at undgå overflødige informationer, grafikker og effekter på hjemmesiden.

Design og layout bygger på princippet om, at bruge kendte konventioner fra internettet for at øge brugervenligheden. Ved at bruge løsninger, der er kendt fra andre hjemmesider, bliver hjemmesiden lettere for brugerne at aflæse og afkode.

### ***Indhold***

#### **Brugervenlige tekster**

De fleste brugere skimmer websider og læser kun, hvis noget fanger deres interesse. Ud fra denne viden kan man opstille en række simple regler for at skrive brugervenlige tekster til Internettet.

- Tekstafsnit bør være korte samt skimmevenlige, og alle tekstafsnit bør forsynes med informative overskrifter, der hurtigt og nemt kan aflæses.
- Indholdstunge afsnit kan med fordel skrives om eller brydes op ved hjælp af en punktopstilling, hvor hvert enkelt punkt kort beskriver en hovedpointe.
- Skriv gerne konklusionerne øverst på siden, og uddyb, argumenter og diskuter længere nede.

Hvis teksten er meget lang, kan hovedpunkterne eventuelt efterfølges af et link til en underside med den fulde tekst. Det bedste er dog, hvis alle tekster kan skrives sådan, at de kan læses umiddelbart og uden referencer til resuméer eller længere versioner.

Herudover bør man tilstræbe, at teksten fremstår som et hele og kan læses uafhængigt af andre sider på hjemmesiden. Man kan ikke forudsætte, at læseren har læst den foregående side, og man bør derfor undgå vendinger i stil med "som vi beskrev på forrige side".

Dette afsnit er et eksempel på en tekst, med en brugervenlig opbygning.

#### **Når du skriver/indsætter tekst**

- Hvis du skal bruge tekst fra et andet sted, skal du **altid** kopiere det ind i en simpel teksteditor som f.eks. Notesblok/Notepad, og derfra kopiere det igen, før du indsætter det i Plone. Dette er for at undgå at forskellige usynlige formateringer følger med over i Plone og giver uhensigtsmæssige resultater.
- Brug **fed** til evt. fremhævelser. Pas på ikke at fremhæve for meget. Som hovedregel max. en fremhævning pr. overskrift/underrubrik.
- Brug kun kursiv til citater. Ikke til fremhævelser.
- Brug altid de logiske overskriftniveauer. Altså størst overskrift øverst på siden. Og mindre overskrifter (underrubrikker) længere nede for at dele teksten op.
- Brug ikke STORE BOGSTAVER til at fremhæve.
- Undgå for mange forkortelser og brug af &.

Hvis noget går galt i din formatering, så kopier indholdet over i en simpel teksteditor, og kopier det herfra tilbage til Plone igen.

## Billeder

Billeder kan i visse tilfælde indeholde informationer, der er svære at formidle med ord. De kan bringe liv til ellers kedelige sider. Og en teksttung side kan desuden gøres mere behagelig at se på ved hjælp af et relevant billede.

- Brug altid skarpe og relevante billeder
- Brug som hovedregel max. et billede pr. webside.
- ALT-tekst er **obligatorisk**. Som default sætter Plone **ALT-teksten** til billedets titel. Tjek at **ALT-teksten** er kort og beskrivende.

Placering af billeder i Plone kan af og til drille. For det mest forudsigelig resultat brug "Image Alignment Venstre", som lader tekst bryde omkring billedet eller "Alignment Inline", som ikke lader tekst bryde om sig.

"Alignment Højre" kan give problemer og resultatet kan være svært at styre. For at opnå det bedste resultat kan det være nødvendigt at eksperimentere.

## Billedestørrelser

Max bredder på billeder. Det er **vigtigt** at disse bredder overholdes. Ellers vil billederne rage ud over de spalter, de befinder sig i.

- Max bredde i hovedområdet: 500px.
- Max bredde i portletter: 300px.

Se supplerende læsning bagest i designdokument for en beskrivelse af, hvordan du ændrer størrelsen på et billede.

## Links

- Brugernes øjne fanges af links. Hvis linkteksten er sigende finder brugerne hurtigere, det de leder efter. Brug derfor ikke indholdsløse ord til links(f.eks. ordet "her").
- Hvis du linker til et dokument, så lad dokumentets navn være linkteksten og skriv filtype og størrelsen efter linket. F.eks. Program (pdf 10.7mb) eller Skema (xls 1.2mb)
- Link må aldrig åbnes i separat vindue, da det begrænser brugerens navigeringsmuligheder.

- Vær opmærksom på at linkunderstregningen ikke må medtage efterfølgende punktum eller mellemrum.

## Opret webside

Plone laver webadresser efter titlen på websiden. Derfor skal du skrive titlen så kort og præcist som muligt. Så vil webadressen også blive kort og præcis. F.eks. :  
*www.statsbiblioteket.dk/kurser/etnografi-antropologi* frem for:  
*www.statsbiblioteket.dk/kurser/etnografi-og-antropologi-informationssogning-for-specialestuderende*

Hvis du ønsker en længere og mere sigende titel på selve websiden, skal du først gemme websiden med den korte titel (Plone sætter webadressen første gang du gemmer). Dernæst kan du redigere den ønskede titel ind f.eks. : "Etnografi og antropologi. Informationssøgning for specialestuderende".

OBS: Du må **ikke** skrive i feltet **Beskrivelse**, når du opretter en webside.

## Tabeller

Generelt skal tabeller kun bruges til tabulerbare data som f.eks. lister, dagsprogrammer til events, kurser, lister etc.

Der er fire tabeltyper til rådighed.

- Plain
  - Tabel med flydende kolonnebredde, der retter sig ind efter data i de enkelte celler.
- Plain grå
  - Samme som ovenstående men med grå baggrund
- 2-4 kolonner fast bredde (hvid)
  - Tabel hvor kolonnebredden er fast, uafhængigt af indholdet.
- 2-4 kolonner fast bredde (grå)
  - Samme som ovenstående med grå baggrund

Hvis der ønskes overskrifter til de enkelte kolonner vælges **Opret tabeloverskrifter**

## Format for kontaktoplysninger

- Jens Jensen, Universitetsparken, 8000 Århus C (uden DK foran) · Telefon (ikke forkortet) 1234 5678, [jj@email.dk](mailto:jj@email.dk)
- Skriv ikke email foran emailadressen, da det giver sig selv. Undgå at lave et mailto-link på navnet, da det ikke giver sig selv hvad der linkes til.
- Skriv ikke faxnummer, da det bruges meget sjældent. I de tilfælde der er brug for faxnummer kan brugeren spørge efter det.

## FAQ

### Hvorfor kan jeg ikke logge ind?

P.g.a. problemer med, hvordan LDAP og Plone arbejder sammen, risikerer man efter at have tastet sit brugernavn/password forkert ind, at blive låst ude fra Plone. Nogle gange optræder

problemet efter kun 1 forkert indtastning. Der gives ikke besked om dette. Kontakt helpdesk, for at få din brugerprofil låst op igen.

#### **Hvorfor åbner vi ikke udgående links i nye vinduer?**

Vi ønsker at overholde W3.org's standarder, og vil derfor ikke åbne udgående links i nye vindue. Ved at blive i samme vindue bevarer brugeren selv kontrol over sin browser. Endvidere kan browserens tilbageknop bruges til at navigere med.

#### **Skal linklister helst være med bullets eller uden?**

Linklister skal have bullets, da det er en liste. Bullets gør generelt lister mere overskuelige.

#### **Hvorfor har links de farver?**

Vi ønsker at bruge browserens standard farver på links. Når vi bruger kendte konventioner fra Internettet vil flest mulige forstå det. Dermed bliver det mere brugervenligt.

## ***Supplerende læsning***

### **Links til videre læsning**

Plone begyndertutorial, der dækker alt fra oprettelse af sider og mapper til indsættelse af billeder og oprettelse af links:

<http://learnplone.org/documentation/tutorial/plone-25-and-kupu-14/tutorial-all-pages>

En lidt mere grundig manual finder du hos plone.org.

<http://plone.org/documentation/manual/plone-2.5-user-manual>

Plone.org er også stedet du skal lede, hvis du står med et mere specielt problem.

### **Billedbehandling**

Hvis du har et billedbehandlingsprogram på din maskine så brug den. Ellers brug et online billedbehandlingsprogram som f.eks. Picnik: <http://www.picnik.com/app#/in/upload>

### **Hjælp til online billedbehandling med Picnik**

- Upload dit billede til picnik.com med **get photo** under Photos
- Beskær dit billede, med **crop** funktionen. Lav gerne en tæt beskæring, hvor du skærer overflødigt billedmateriale væk.
- Du har mulighed for at justere på lys, kontrast, farvebalance etc. Brug det kun varsomt, da det generelt forringer billedkvaliteten.
- Vælg **resize** for at sætte den nye billedstørrelse. Husk maxbredderne: 500px og 300px
- Billedet kan være blevet let sløret efter at du har ændret størrelsen på det. Det kan afhjælpes ved at give billedet en **anelse** skarphed (sharpen). Pas på der ikke kommer "auraer omkring kanter i billedet.
- Til sidst gemmer du billedet under Save and Share.
  - Fotos gemmes som jpg med en kompression på ca. 8, hvilket giver en god balance mellem størrelse og kvalitet.
  - Anden grafik med kun få farver, gemmes som gif.

### **Hjælp til basal billedbehandling med en anden editor**

Dette er en generel beskrivelse, så din billededitor vil sandsynligvis fungere en smule anderledes.

- Hent dit billede i din yndlingsbilledbehandlingsprogram

- Beskær dit billede. Lav gerne en tæt beskæring, hvor du skærer overflødig billedmateriale væk. Typisk hedder beskæringsværktøjet **beskær/crop**, men du kan også komme ud for, at du skal bruge et markerings værktøj, og derefter vælge **cut**.
- Nu kan du evt. justere en smule på lys, kontrast eller farvebalance, men brug det varsomt, da det generelt forringer billedkvaliteten.
- Vælg **billedestørrelse/imagesize/resize** for at sætte den nye billedstørrelse. Afkryds evt. **bevar proportioner/constrain proportions**, for at bevare billedets proportioner. Vælg pixels som **enhed/unit**. Husk maxbredderne: 500px og 300px
- Billedet kan være blevet let sløret efter at du har ændret størrelsen på det. Det kan afhjælpes ved at give billedet en anelse **skarphed/sharpen**. Hold øje, at der ikke kommer "aurer" omkring skarpe kanter på billedet. Så har du givet for meget skarphed.
- Til sidst gemmer du billedet.
  - Fotos gemmes som jpg med en kompression på ca. 8 ud af 10 eller 80%, hvilket giver en god balance mellem størrelse og kvalitet.
  - Anden grafik med kun få farver gemmes som gif.

Et par gode gratis billedbehandlingsprogrammer er Irfanview: <http://www.irfanview.com/> som er en helt basal billedbehandler og billedfremviser.

Eller Gimp: <http://www.gimp.org/> som er et meget avanceret billedbehandlingsprogram, der næsten har samme funktionalitet som Photoshop. Begge programmer er gratis.

### ***Bilag 3 Oversigt over filer på CD'en***

CD'en indeholder følgende filer:

- Hovedopgaven i wordformat,
- Hovedopgaven i PDF
- Samtlige projektfiler (ca. 200) i zippet.fil

Derudover ligger i løs form, de kildefiler, som jeg selv har skrevet eller tilrettet.

- basket.jsp (skrevet fra bunden)
- basket\_show.xsl (skrevet fra bunden)
- short\_record\_show.xsl (tilrettelser)
- do\_request.jsp (skrevet fra bunden)

### ***Bilag 4 Udviklingsværktøjer***

Vi har tidligere i uddannelsen skrevet alt kode i hånden, og jeg har kun benyttet en lille teksteditor, som udelukkende har haft den funktion, at den kunne lave syntaks highlightning (Koden bliver farvet afhængigt af den syntaktiske funktion, den har. Det gør koden meget lettere at overskue).

På Statsbiblioteket bruger de det professionelle udviklingsværktøj IntelliJ IDEA. IDEA er enormt stykke software, som jeg kun har fået et overfladisk kendskab til, men allerede nu kan jeg se, hvilken lettelse, det er i den dagligdags programmering, at have et så stærkt værktøj til rådighed.

Konkret kan IDEA lave samme syntaks highlightning, som de specielle programmerings teksteditorer også kan. Derudover kan IDEA også komme med forslag til kode, da det kender et hav af funktioner og udtryk. Det kan vende udtryk automatisk, så du feks med



et klik kan skifte mellem at sammenligne om udtryk er forskellige eller ens, ændre loops fra f.eks. for-løkker til for-each-løkker,

IDEA indeholder desuden flere forskellige værktøjer til versionskontrol. Versionskontrol går ud på at flere kan arbejde på den samme kode på samme tid. Versionskontrollen holder så øje med hvem der sidst har tilføjet noget kode, hvad der er blevet ændret i koden, og den kan "smelte" forskellene sammen, hvis flere personer på samme tid har rettet forskellige steder i koden. Denne funktion er uundværlig, i større projekter, hvor flere programmører arbejder sammen. Den kontrolversion vi har arbejdet med hedder CVS (Concurrent Version Control). CVS er open source software, og bruges også i udviklingen af mange andre store open source projekter.