



# On Page SEO

---

WU HF

Niels Harbo 2019

# Hvad er On Page SEO?



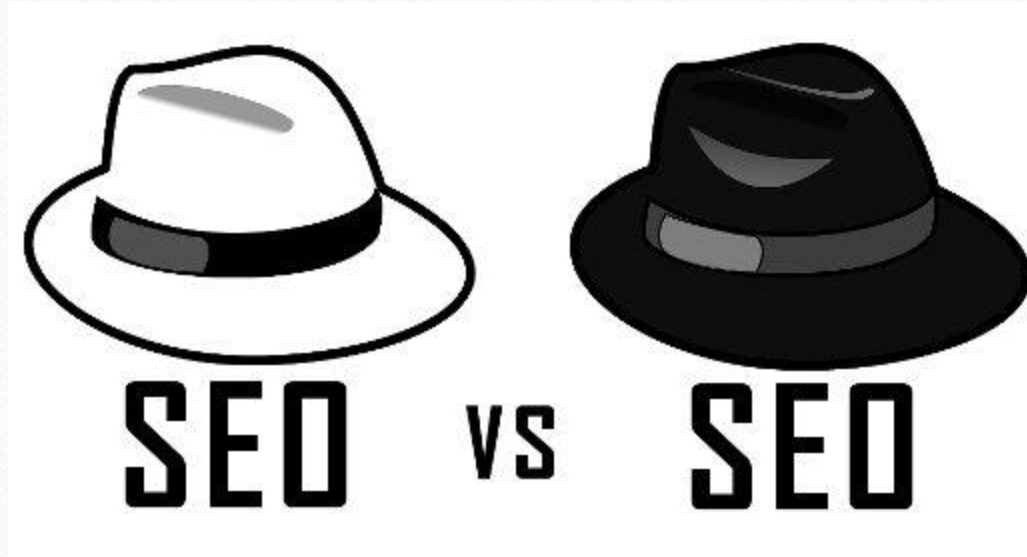
# Hvad er On Page SEO?

On Page	Off Page
Kvalitetsindhold	Link building
Optimal formatering	Guest blogging
Keyword analyser og regelmæssige opdateringer af indhold	Social media strategier
Kvalites URLs (Protokol/ pretty urls/ canonical links)	Eksterne tjenester (Trustpilot, Youtube, Soundcloud, etc.)
UI/ UX/ Accessibility	Kampagner/ Affiliate programmer/ Annoncering/ Remarketing
Hastighed (server svartid/ http requests/ ressourcer)	
Dedikeret IP adresse	
Send XML sitemap til søgemaskiner	
Rich data	
Open Graph meta data	



# White Hat & Black Hat

---



# Black hat

---

Black Hat SEO er når man forsøger at manipulere sig til høje rankings. Med teknikker som keyword stuffing, kopieret indhold, skjult indhold, links og redirects til irrelevant indhold.

Black Hat Off Page SEO benytter sig af linkarkiver, link farms og andre "kunstige" former for link building, samt link-spam på fora og div tjenester.

I reglen kan Black Hat strategier kun give kortsigtede resultater, og resulterer ofte i såkaldte *penalties*, så snart søgemaskinerne opdager det.

# White Hat

---

White Hat er Den etisk korrekte og langsigtede strategi.

White Hat kræver at man følger og respekterer søgemaskinernes anbefalinger, samt leverer relevant og originalt kvalitetsindhold, via tilgængelige og brugervenlige løsninger.



# Kvalitetsindhold/ Content is king

---

- Sæt dig i brugerens sted.  
Leverer du originalt, brugbart indhold eller bare varm luft?
- Undgå *duplicate content*. Citerer du en kilde så brug blockquote tags.
- Er mængden af indhold pr. side passende?
- Medieindhold bør indlejres i henhold til standarder og kan med fordel forsynes med transskriptioner og undertekster.

# Regelmæssige opdateringer og gode keywords

---

- SERPs er dynamiske så en god ranking kan hurtigt ændre sig hvis indholdet aldrig opdateres
- Analyse af keywords. Populære søgetermer er måske ikke altid bedst, hvis der er meget konkurrence kan man med fordel bruge mindre populære keywords.
- Ønsker man fyldestgørende, langsigtede analyser kan man med fordel embedde et google analytics script på sit site.



# Tilbyd alternativt indhold

---

Accessability (alt attributter/ title attributter/ korrekte links til alternative sprog)

Korrekt information i head elementet om alternativt indhold:

```
<link rel="alternate" hreflang="en" href="https://mit-svedige-site.dk/en">
```

På den "alternative" side skal der være et tilsvarende link til det "originale" indhold:

```
<link rel="alternate" hreflang="da" href="https://mit-svedige-site.dk/">
```

# Optimal formatering

---

- Korrekt semantisk markup. F.eks. bør en side/ artikel/ sektion kun have én h1
- Tydeligt Hierarki.
- Korrekt brug af uppercase (maskiner læser uppercase som forkortelser. Hvis uppercase bruges stilistisk, bør det specificeres m. CSS)
- Adskil grafisk lir og dekoration fra indholdet (CSS/ i stedet for indlejring)

# Basal Formatering – Udelad ALDRIG:

---

- Title tag i head elementet
- H1 element
- Doctype deklaration
- Tegnsæt deklaration
- Korrekt lang attribut på html elementet
- Viewport meta tag



# Kvalites URLs

---

- SSL-certificering burde egentlig ikke påvirke ranking, men det gør det i praksis da, andre sites typisk er mere tilbøjelige til at linke til HTTPS sider.
- Brug pretty URLs (hvis muligt) i stedet for parametre, af samme grund som overfor, brugere er mere tilbøjelige til at trafikere brugervenlige urls.

***mysite.com/artikler/learn-on-page-seo***

I stedet for:

***mysite.com?postype=article&post-id=1153***

# Kvalites URLs (Canonical links)

Angiv et canonical link hvis en url har mange kombinationsmuligheder

---

<http://mysite.com/?page=home&la=en>  
<https://mysite.com/?page=home&la=en>  
<http://mysite.com/?la=en&page=home>  
<https://mysite.com/?la=en&page=home>  
<http://mysite.com/?la=en&page=home#main>  
<https://mysite.com/?la=en&page=home#main>  
<http://www.mysite.com/?page=home&la=en>  
<https://www.mysite.com/?page=home&la=en>  
<http://www.mysite.com/?la=en&page=home>  
<https://www.mysite.com/?la=en&page=home>  
<http://www.mysite.com/?la=en&page=home#main>  
<https://www.mysite.com/?la=en&page=home#main>

***Alle disse URLs vil i reglen generere det samme indhold, Dvs. vi i teorien har mindst 12 sider med duplicate content***

# Kvalites URLs (Canonical links)

---

```
<link rel="canonical" href="https://mysite.com">
```

Canonical links forklarer søgemaskiner hvilket af de mange URLs der er den foretrukne.



# Accessibility

---

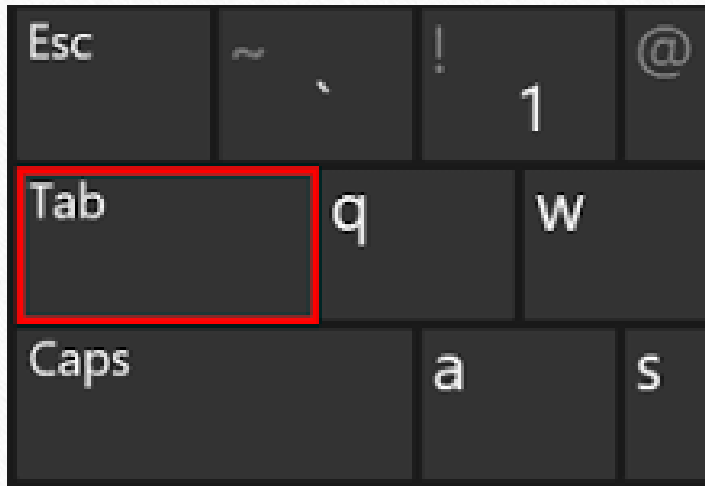
Søgemaskiner ranker sider ned, med dårlig accessibility.

# Accessibility - Eksempler

---

- Klikbare elementer der er meget tæt på hinanden rankes ned på søgninger foretaget på mobile enheder.
- Skip to content links for skærmlæsere er et andet eksempel på god praksis for accessibility.

```
<a class="skip" href="#content" tabindex="0">  
  Spring til indholdet  
</a>
```



```
.skip {  
  position: absolute;  
  top: -1000px;  
  left: -1000px;  
  height: 1px;  
  width: 1px;  
  text-align: left;  
  overflow: hidden  
}  
a.skip:active, a.skip:focus, a.skip:hover {  
  left: 0;  
  top: 0;  
  width: auto;  
  height: auto;  
  overflow: visible;  
  color: #fff;  
  background: #000;  
  padding: 1rem  
}
```



# Hastighed

---

Et website der svarer hurtigt og har lave bouncerate, ranker helt naturligt op pga. brugergenereret trafik, søgninger delinger osv. Men søgemaskiner kan også selv beslutte at et site er af lav kvalitet hvis det er langsomt.

Hastighed og svartid kan optimeres både på backend og frontend

# Hastighed - Backend

---

- Optimering/normalisering af databaser
- Server caching og cronjobs der genererer statisk HTML i stedet for databasekald ved hvert besøg.
- Kompilering af backend kodefiler

# Hastighed - Frontend

---

- Minify og bundle af kodefiler (reducerer både datamængden og antal af HTTP requests)
- Brug af CDN (Content Delivery Networks) til biblioteker, fonte mm. Data fra populære CDNs er ofte allerede lagret i brugerens browser cache.
- Komprimering af billeder/ mediefiler
- Lazy load/ Asynkron load af ressourcer



**Få anbefalinger til hastighedsoptimeringer her:**



**PageSpeed Insights**

<https://developers.google.com/speed/pagespeed/insights/>

# Google Search Console

---

- På Google Search Console kan man monitorere om der er problemer med indekseringen af nogen sider. Man kan også angive om den primære url er med eller uden www prefix.

# XML sitemap

---

- Ved at submitte et sitemap til Google Search Console (Bing har en lign. funktion) kan man fortælle hvilke sider der skal indekseres, hvor ofte de forventes opdateret og hvor høj prioritet hver side skal have.
- Man kan finde en del XML sitemap generatorer online



# Indhold der ikke ønskes indekseret

---

Forestil dig at du har en side der viser søgeresultater vha url parametre.

Mulighederne for urls er uendelige men kun relevante for den enkelte bruger. Derfor kan det være en god ide at fortælle søge maskinerne at de ikke skal indeksere søgeresultat-siden.

Dette kan vi gøre med en **robots.txt** fil på serveren eller et noindex metatag direkte i head elementet på siden:

```
<meta name="robots" content="noindex">
```

# Indhold der ikke ønskes indekseret

---

Vi kan også fortælle google at en bestemt sektion på siden ikke skal indekseres.

Det kan være nyttigt hvis vi har en sektion med f.eks. reklamer eller andet irrelevant indhold.

`<p>This is normal (X)HTML content that will be indexed by Google.</p>`

`<!--googleoff: index-->`

`<p>This (X)HTML content will NOT be indexed by Google.</p>`

`<!--googleon: index>`

# Strukturerede data

---

Hvad er strukturerede data?

Strukturerede data bruges til at forklare indholdet til søgemaskiner. De mest almindelige kaldes *microdata* og beskriver indholdet ud fra et katalog som findes på:  
**[schema.org](https://schema.org)**



# Strukturerede data (Eksempel)

---

**Website:** *Jeg elsker Virginia*

**Søgemaskine:** *Fint, det må jeg hellere indeksere, men hvem er det relevant for?*

*Brugere der elsker:*

- *Amerikanske stater?*
- *Engelske pigenavne?*
- *Tobak-sorter?*

# Strukturerede data (Eksempel)

---

Vi kan fortælle søge maskinen at Virginia er en pige (-ikke en stat eller en tobakssort)

```
<p itemscope="" itemprop="person" itemtype="http://schema.org/Person">  
  Jeg elsker <span itemprop="name">Virginia</span>  
</p>
```

# Hvad skete der lige?

---

Vi tilføjede et *itemscope* til et HTML-element for at fortælle at dette scope indeholder strukturerede data. Alle child elementer i dette scope kan nu få en property der beskriver egenskaber

```
<p itemscope="">  
  <!-- egenskab -->  
  <!-- egenskab -->  
</p>
```



# Hvad skete der lige?

---

Vi tilføjede desuden en itemtype attribut for at forælle hvilken "ting" der er tale om. Søgmaskinerne slår nu termen "Person" op på schema.org

```
<p itemscope="" itemtype="http://schema.org/Person">
```

Lad os prøve at gøre det samme og se hvilke properties vi kan give vores elskede Virginia

<https://schema.org/Person>

# Et person-scope har mange mulige properties

---

```
<p itemscope="" itemprop="person" itemtype="http://schema.org/Person">  
  Jeg elsker <span itemprop="name">Virginia</span> fordi hun er  
  <span itemprop="jobTitle">buschaufør</span>  
</p>
```

# Her validere vores strukturerede data

---

<https://search.google.com/structured-data/testing-tool>

Person		0 FEJL 0 ADVARSLER ^
@type	Person	
name	Virginia	
jobTitle	busschaufør	



# Her kan en nem editor

---

- <https://www.google.com/webmasters/markup-helper/>

# Open Graph Metadata

---

Open Graph data bruges af sociale medier bl.a. Facebook og Twitter.

Når side deles kan man bruge et tilpasset billede og tekst vha. OG data.

OG data indsættes i head elementet

# Open Graph Metadata

---

<meta property="og:title" content="Titel her">

<meta property="og:image" content="billede.jpg">

<meta property="og:image:width" content="1200">

<meta property="og:image:height" content="650">

<meta property="og:url" content="https://mitwebsite.dk">

<meta property="og:type" content="website">



# Gruppeopgave - Analyse og optimering af eksisterende site

---

- <http://frknielsen.com/>
- <https://qars.dk/>
- <http://tyra.dk/>

# Gruppeopgave - Analyse og optimering af eksisterende site

---

- Vælg et af disse 3 websites. Skriv en rapport (ca. 2 sider) der forklarer hvordan hvert site kan forbedres med On Page SEO-teknikker. F.eks:
- Optimering af indhold, keywords og URLs
- Optimering af markup, hastighed/ performance og UI
- Optimering af metadata herunder Open Graph og Micro Data.
- I skal også generere et XML site map for sitet, evt. med prioriter til hver side