

# Välkomna till HTML & CSS

# Om mig

Viktor Silfverström

- Har jobbat med Webb & IT de senaste 8 åren
- Började min karriär som junior på bolaget Bilprospekt
- Driver idag bolaget North Link AB i Helsingborg

# Vad ni kommer lära er i den här kursen

- **HTML** (HyperText Markup Language), ett markup-språk som används för att beskriva strukturen i hemsidor.
- Grundläggande **CSS** (Cascading Style Sheets), ett deklarativt språk för att beskriva hur en hemsida ser ut. Färger, textstorlek, mm.
- Hur webben fungerar. **Webbläsare** och **webbserver** och hur de kommunicerar.
- Grundläggande **SEO** (Search Engine Optimisation). Hur sökmotorer indexerar hemsidor och hur man driver trafik till en hemsida eller tjänst.

# Kursens struktur

- Åtta veckor med föreläsningar och övningar
- Två obligatoriska laborationer som behöver bli godkända
- Praktisk tentamen i slutet av kursen

# Lektionernas upplägg

- Mikroföreläsningar där vi introducerar nya koncept
- Repetition av tidigare material (säg gärna till om det är något ni vill lära er djupare)
- Korta övningar som görs enskilt eller i grupp och vi går genom tillsammans
- Tid för frågor

# PingPong

<https://yh.pingpong.se/>

Utbildningsverktyg. Jag kommer att lägga upp:

- Föreläsningssides (efter lektion)
- Länkar till dokumentation, övningar och artiklar
- Övningsuppgifter
- Laborationer
- Tentamen

# Veckoplanering

- Vecka 1

- Introduktion till struktur och användningsområden för HTML
- Introduktion till **atom**-editorn
- Göra en enkel hemsida
- Vanliga HTML-taggar och hur de används
- **Hyperlänkar**
- HTML-attribut

- Vecka 2

- Listor
- Formulär och validering
- Tabeller
- Introduktion till **git** och **GitHub**
- Introduktion till **HTTP**

# Veckoplanering

- Vecka 3
  - Introduktion till **styles** och style-attributet
  - Färger och storlekar
  - Introduktion till **CSS**
  - **Klasser** och **ID:n**
  - Grundläggande **CSS-selectors**
  - **Första labben!**
- Vecka 4
  - **DOM** (Document Object Model)
  - Felsökning och avlusning med Chrome DevTools
  - Layout och positionering med CSS



# Veckoplanering

- Vecka 5

- Introduktion till **responsiv design**
- Wireframes
- Använda **media queries** för att anpassa för olika enheter
- Viewport-taggar
- Flexbox
- **Andra labben!**

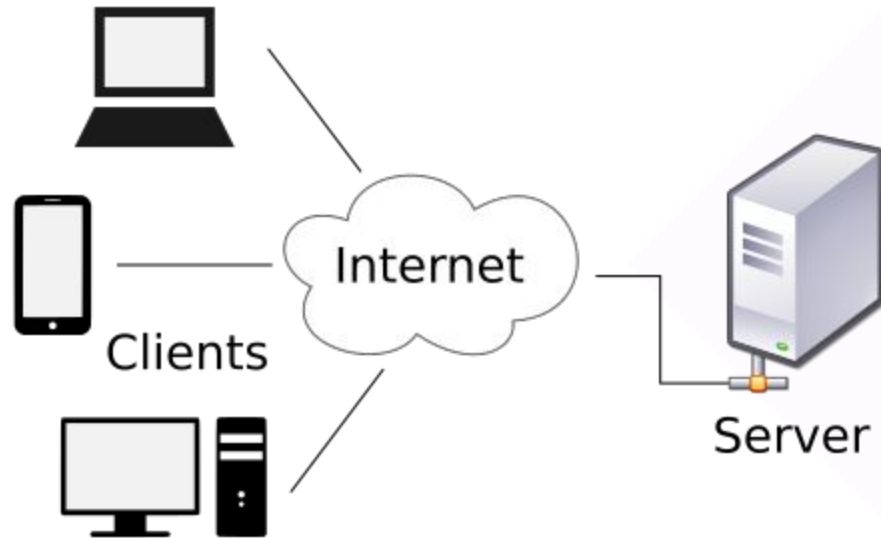
- Vecka 6

- Grid
- Introduktion till **Bootstrap**

# Veckoplanering

- Vecka 7
  - SEO
  - Meta-taggar
  - Media (spela upp ljud och bild på webben)
  - Använda **transitions** och **transforms**
  - Animationer
- Vecka 8
  - Nyheter i CSS
  - CSS-variabler
  - Olika webbläsare
  - **Preprocessorer till CSS**
  - **Tentamen!**

# Webb



# Hur fungerar webb?

- Webbläsare (klient) - är det program som en användare gör en request med
- DNS (Domain Name Server) - översätter ett domännamn till en IP
- Webbserver - tar emot en request från en webbläsare och returnerar HTML
- TCP/IP & HTTP - är de protokoll som används för kommunikationen

Läs mer på

[https://developer.mozilla.org/en-US/docs/Learn/Common\\_questions/How\\_does\\_the\\_Internet\\_work](https://developer.mozilla.org/en-US/docs/Learn/Common_questions/How_does_the_Internet_work)

# Webbutveckling



# Webbutveckling



# Webbutveckling



# Webbutveckling

**HTML**



**CSS**



**JS**





# Bra resurser

- **MDN (Mozilla Developer Network)**  
<https://developer.mozilla.org/en-US/>  
Bra som referens
- **Stack overflow**  
<https://stackoverflow.com/>
- **Khan Academy**  
<https://www.khanacademy.org/computing/computer-programming/html-css>  
Introduktion till HTML och CSS med många övningar
- **Code Academy**  
<https://www.codecademy.com/learn/learn-html>  
Kort introduktion till HTML och CSS

# Verktyg som behövs

- En text-editor
- git
- GitHub-konto
- En webbläsare (jag rekommenderar Chrome)

# Övning - Hello World!

1. Installera **atom** om ni inte redan har en text-editor som ni redan använder  
*Atom finns att ladda ner på <https://atom.io/>*
2. Skapa en ny mapp någonstans på er dator och öppna den med atom
3. Skapa en fil som heter **index.html**
4. Skriv följande innehåll i filen och spara den

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Hello</title>
  </head>
  <body>
    <p>Hello, world!</p>
  </body>
</html>
```

5. Öppna filen i din webbläsare

# Vad hände precis?

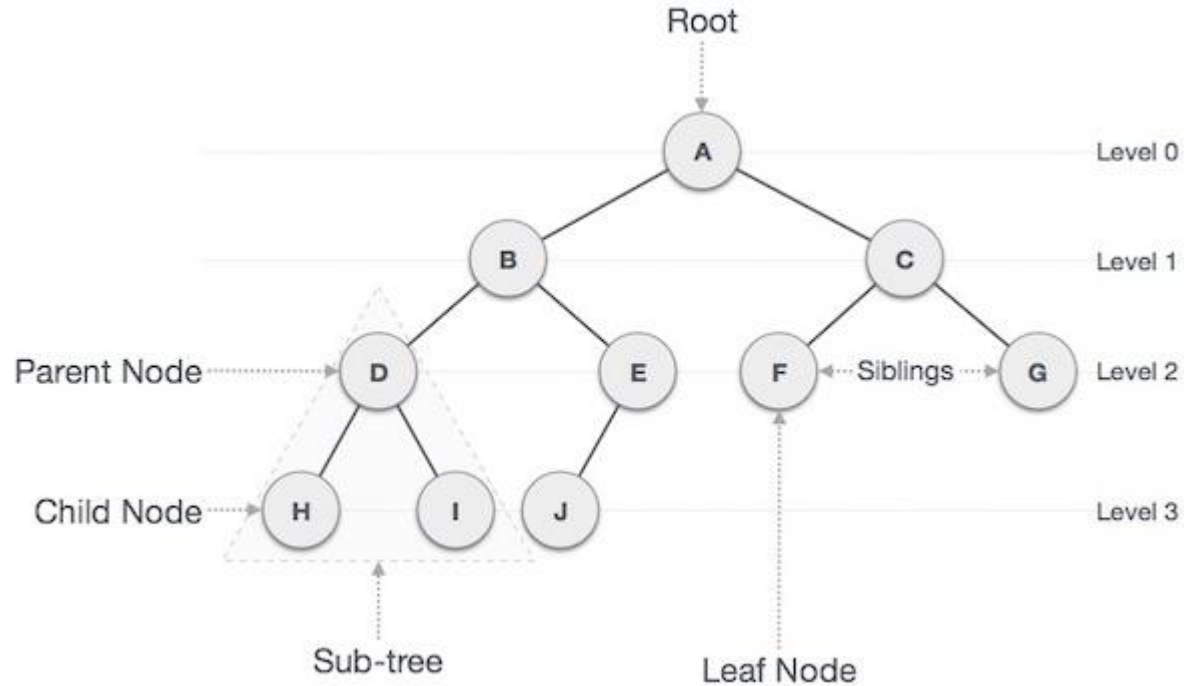
- Du skapade en .html fil som innehåller en HTML struktur
- Du bad din webbläsare att öppna filen
- Webbläsaren (klienten) gjorde en request till ditt filsystem (servern)
- Filsystemet returnerade HTML filen
- Webbläsaren läste och tolkade filen
- Webbläsaren renderar ut pixlar på din skärm enligt sin inbyggda standard
- Du ser en hemsida!

# Vad är HTML?

Det du precis såg i index.html var en giltig HTML struktur. All HTML kod ser ut på detta vis. Koden är uppbyggd på ett gäng olika element som man kan kombinera på ett oändligt olika sätt.

- HTML är ett markup-language
- HTML strukturen ser ut som ett träd
- Ett träd består av flera “noder”
- Ett träd har en stam (root) nod
- Varje nod har noll, en eller flera grenar som leder till nya noder
- Om en nod inte har en gren så kallas den för ett löv

# Träd struktur



# HTML element

- Ett element är en speciell typ av nod
- Varje element skrivs ut med hjälp av **tags**
- En tag består av ett namn som är omringad av vinkelparenteser (< >)
- En start-tag visar början av ett element och skrivs ut som <namn>
- En slut-tag visar slutet av ett element och skrivs ut som </namn>
- Löv element kan avslutas direkt med <namn />
- I HTML är namnen på alla element bestämda och innebär olika saker
- Text är inget element utan en är en “text-nod” och behöver inga taggar

Exempel

<p>Hejsan!</p>

# HTML element - attribut

- Ett element kan ha en eller flera attribut noder
- En attribut-nod hjälper oss att ge extra information om ett element utan att ange en ny gren
- En attribut består av ett namn och ett värde
- Syntaxen är “namn=värde”, varje attribut separeras med ett mellanrum
- Om värdet är text ska det omringas av citationstecken (“ eller ‘, men inte båda samtidigt)

Exempel

```
<p class="minklass" id="mittid">Hejsan!</p>
```



# ID och klasser

Det finns två attribut som går att använda på så gott som alla element i HTML

- **“id”** används för att ge ett element en unik identifierare. Det får endast finnas ett element med ett visst Id på en sida.
- **“class”** används för att ge ett element en viss klass. En klass kan användas på mer än ett element och används om många element har samma klass.

ID:n och klasser används för att komma åt element när man vill ge dem ett utseende i CSS och även för att hantera element med JavaScript.

# HTML element - typer

- Block-level element är element som innehåller strukturella element för en sida.
- Inline element är element inne i block-level elements och innehåller mindre delar av data som text, eller länkar.
- Ett block-level element kan ha flera block-level eller inline element som grenar
- Ett inline element kan ha en eller flera inline element som grenar
- Ett block element skapar ett nytt “block” (en ny rad) på en sida
- Ett inline skapar inte ett nytt “block” på en sida
- Block-level vs inline element är inte definierat i HTML5 (men fungerar likadant)

# HTML block-level element

- div - generisk block-level element som kan innehålla vad som helst
- footer - sektion för botten av sidan
- form - ett formulär
- h1, h2, h3, h4, h5, h6 - headers
- header - sektion för toppen av sidan
- hr - ett horisontell linje
- p - en paragraf
- ul, li - element för att skapa listor
- table - en tabell

[https://developer.mozilla.org/en-US/docs/Web/HTML/Block-level elements](https://developer.mozilla.org/en-US/docs/Web/HTML/Block-level_elements)

# HTML inline element

- a - anchor element, skapar länkar till andra sidor eller delar av egna sidan
- b - fetttext
- br - ny linje
- button - skapar en knapp
- i - italics
- img - ladda in en bild
- input - skapa ett fält för att användaren kan skriva text (används i form)
- select - skapar en dropdown lista som användaren kan integrera med
- span - generisk inline container
- strong - används för att betona ett budskap i text

[https://developer.mozilla.org/en-US/docs/Web/HTML/Inline elements](https://developer.mozilla.org/en-US/docs/Web/HTML/Inline_elements)

# HTML struktur

- html elementet är stommen (root elementet)
- Varje html element har två grenar, en head och en body
- head innehåller meta data om sidan och inget i head visas direkt för besökaren.
- body innehåller allt det innehåll du vill att besökaren ska ta del av.

# Grundläggande taggar

- `<h1>`, `<h2>`, ...  
Överskrifter
- `<p>`  
Påbörjar en ny paragraf
- `<br>`  
Radbrytning
- `<hr>`  
Används för att separera innehåll

# Text-formatering

Det finns många taggar för att formatera text.

- `<em>`  
Ger *emfas* till ett stycke eller ord. `<i>` går också att använda för **kursiv** text.
- `<strong>`  
Används för att markera något **viktigt**. `<b>` kan också användas för **fetstil**.
- `<pre>`  
Visar text utan att ändra indentering. T ex kod
- `<small>`  
Används för att visa liten text

# HTML exempel

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Hello</title>
  </head>
  <body>
    <p>Hello, world!</p>
  </body>
</html>
```



# HTML övning

<https://bit.ly/2PQWp6v>

övning 1, 2 & 3

# URI & URL

- URI (Uniform Resource Identifier) används för att hitta och identifiera resurser
- URI är bara en kombination av olika bokstäver, men formatet är något som är förstått av alla datorer
- URI har formatet: **scheme://path** där *scheme* är sättet hur vi hämtar filen, och *path* är var resursen existerar
- Exempel: **file:///Users/viktor/index.html** pekar till en fil på min dator
- För windows hade det sett ut såhär: **file:///c:/index.html**
- URL (Uniform Resource Locator) är en speciell typ av URI som identifierar en webb-resurs
- URL har ofta formatet: **scheme://subdomain.domain.topdomain/path** där *scheme* vanligast är http eller https

# URL Paths

- En absolut sökväg (engelska path) är en “absolut” identifierare för en resurs. Dvs oavsett vart du försöker komma åt resursen så kommer en absolut sökväg att leda dig rätt
- En relativ sökväg kommer leda dig till olika resurser (eller ingen) beroende på varifrån du använder sökvägen.
- Exempel: “index.html” sökvägen kommer att leda dig till filen “index.html” i samma mapp som du är i just nu.
- För relativa sökvägar kan du använda dig av namnet “..” för att “gå upp” en nivå i sökträdet.
- “.” syftar på var du befinner dig just nu

# Sökväg exempel

- Jag befinner mig i en mapp som heter `/a/b/c`
- Jag vill komma åt filen `/a/b/index.html`
- Antingen kan jag använda den absoluta sökvägen ovan, eller så använder jag en relativ sökväg `../index.html`
- Båda ger samma resultat. Relativa sökvägar är att föredra när vi jobbar med webb pga av sättet som en webbläsare söker upp filer. Den kommer jobba långsammare i de flesta fallen med absoluta sökvägar.
- Det betyder såklart inte att man aldrig kan använda absoluta sökvägar

# HTML hyperlänkar

- En hyperlänk (eller länk) är den funktion i en hemsida som gör webben till vad det är
- Med en hyperlänk kan du skapa en “länk” från en sida till en annan. Oavsett om sidan är din egen, eller någon annans.
- Länkar finns överallt på webben, och du använder dem hela tiden när du surfar
- En hyperlänk har tag-namnet **a** och själva länken anges i attributet **href**. Länken ska vara i textformat och är en **URL**.

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/a>

Exempel

```
<a href="http://www.google.com">Google</a>
```

# Övning - Hyperlänk

Skapa två sidor i samma mapp

Ena sidan ska heta *index.html* och ha en hyperlänk till sidan “hello.html”. Sidan ska också en länk till <http://google.com>, denna länken ska öppnas i en ny tab. (Läs om target attributen på MDN dokumentationen)

Andra sidan ska heta *hello.html* och ha en hyperlänk till sidan “index.html”

Varje sida ska också ha en heading (h1) tag som indikerar vilken sida vi är på.

# HTML bilder

- <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/img>

<img>-taggen tar ett antal attribut:

- src - Sökväg till bilden
- alt - Alternativ text. Text om bilden inte kan visas eller användaren ej kan se
- width - Bredden i antal pixlar
- height - Höjden i antal pixlar
- srcset - Används för att använda olika bildkällor beroende på skärmens pixeldensitet.

Exempel

```

```

# Övningar - bilder

- Skapa en sida där du visar en bild på dig själv
- Skapa en sida där du länkar till en bild som finns på webben
- Skapa en sida där du använder en bild som en hyperlänk



# Övningar

Gör resten av övningarna

<https://bit.ly/2PQWp6v>