

HTML & CSS

Välkomna tillbaka

Vad pratade vi om igår?

- 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
- Listor
- Tabeller
- Formulär

HTML vanliga element

- p
- h1, h2, h3, h4, ...
- div
- span
- a
- img
- form, input, select, button, label
- table, thead, tbody, tr, td, th
- ul, ol, li

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

Vad ska vi prata om idag?

- Git & Bash
- HTTP
- Semantiska HTML-taggar
- Introduktion till CSS

git

git är ett versionshanteringssystem ("Version Control System" VCS).

Vi kommer använda det under kursen för inlämningsuppgifter och tentamen.

Fördelar med ett VCS:

- Hela projektets historik finns sparad så det är enkelt att gå tillbaka eller se koden för en tidigare version
- Det underlättar när många personer ändrar i samma filer
- Man kan enkelt skicka koden (pusha) till en server så att man inte riskerar att förlora den

Övning - Installera git

1. Installera git om du inte har det installerat redan
Du kan ladda ner git på <https://git-scm.com/>
2. Öppna terminalen som kommer med git
3. Skriv kommandot **git** i terminalen
4. Ni ska ha fått massa text utskrivet i terminalen

Terminalen

Läs “Lathund - terminalen” för mer info

git-repon

git sparar ett projekts kod i ett **repository** (eller repo kort)

För att skapa ett nytt repo kör man följande kommando i en mapp:

```
$ git init
```

Det går även att klona ett repo som redan existerar med

```
$ git clone ADDRESS_TILL_REPO
```


Lägga till filer i git

För att lägga till filer i ett repo använder man

```
$ git add FILNAMN
```

För att se en lista på filer som lagts till används

```
$ git status
```

Göra en commit

När man är nöjd med sina ändringar kan man göra en “commit” för att spara dem till repot.

```
$ git commit -m "Min första commit"
```

En commit får ett unikt namn som är en SHA-1-hash

För att se historiken används

```
$ git log
```

För att visa en commit används

```
$ git show HASH
```

Övning - Gör en commit

1. Starta ett git-repo i valfri katalog på din dator
2. Lägg till **index.html** i repot
3. Gör en commit

GitHub

<https://github.com/>

Webbtjänst för att hosta git-repon

Ni kommer behöva ett konto på GitHub för att kunna lämna in inlämningsuppgifter

Använda git med en server

För att ladda upp sina lokala ändringar till en server används

```
$ git push
```

För att hämta ändringar från servern används

```
$ git pull
```

För att lägga till en server används

```
$ git remote add origin ADDRESS
```

Övning - GitHub

1. Skapa ett konto på GitHub om ni inte redan har det
2. Skapa en SSH-nyckel och ladda upp den till GitHub

Guide: <https://help.github.com/articles/connecting-to-github-with-ssh/>

3. Skapa ett tomt repo på GitHub
4. Ladda upp repot

Tips: Använd "git remote add" för att lägga till servern i ditt lokala repo

Andra bra saker

- `git branch`
Skapa ett separat träd
- `git checkout`
Byt nuvarande branch du jobbar i
- `git merge`
Sammanslå två “branches”
- `git revert`
Används för att ångra en commit
- `git rebase`
Används för att skriva om historiken

Övning - Git branch

- Skapa en ny branch i ditt repo som heter **development**
- Lägg till eller ändra en existerande fil i ditt repo.
- Commita ändringen
- Pusha din nya branch

Git - mer info

Jättebra guide som täcker allt vi har gått igenom hittills och mycket mer

<https://www.atlassian.com/git/tutorials/setting-up-a-repository>

HTTP-verb

- GET - För att hämta
- POST - För att lägga till en ny resurs
- PUT - Ersätt en resurs
- PATCH - Ändra en resurs
- DELETE - Ta bort en resurs

För att bara hämta ett dokument från en webserver används GET. Tex när man skriver in en adress direkt i webbläsaren.

Statuskoder

Tre siffror som berättar för klient hur anropen gick.

- 1xx - Informational
- 2xx - Success
- 3xx - Redirection
- 4xx - Client error (t ex 404)
- 5xx - Server error

Exempel - Ett HTTP-anrop

GET / HTTP/1.1

Host: www.example.com

HTTP/1.1 200 OK

HEADERS

BODY

<div> och

<div> är ett generiskt **block-level**-element som kan användas för att omsluta andra element.

Den har inget speciellt utseende och betyder inte heller något. Används ofta för att skapa layouter.

På samma sätt är ett generiskt **inline**-element. Används ofta för att ge text ett speciellt utseende.

Både <div> och skall endast användas om det inte redan finns ett element som passar. Både för SEO och för att koden blir lättare att läsa om man använder rätt element.

Exempelvis skulle man kunna använda <div>-element för överskrifter och ge dem ett speciellt utseende med CSS men om man gör det kommer inte sökmotorer förstå vad som är sidans överskrift.

“div soup”

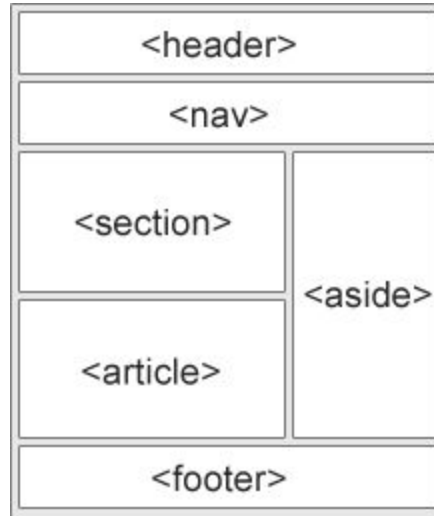
```
▼<div style="position: relative; min-height: 100%;">
  ▶<div class="vIBoZc c5">...</div>
  <div tabindex="0"></div>
  ▼<div class="nH" style="width: 1280px;">
    ▼<div class="nH" style="position: relative;">
      ▶<div class="nH w-asV aiw">...</div>
      ▼<div class="nH">
        ▼<div class="no">
          ▶<div class="nH oy8Mbf nn aeN" style="width: 202px; height: 352px;">...</div>
          ▼<div class="nH nn" style="width: 1078px;">
            ▼<div class="nH">
              ▼<div class="nH">
                ▼<div class="ar4 z">
                  ▶<div id="5" class="aeH">...</div>
                  ▼<div class="A0">
                    ▼<div id="4" class="Tm aeJ" style="height: 342px;">
                      ▼<div id="2" class="aeF" style="min-height: 392px;">
                        ▼<div class="nH">
                          ▼<div class="Blthke nH oy8Mbf aE3" role="main">
                            ▶<div class="mq" style="display: none;">...</div>
                            <div style=</div>
                            <div class="nH a0V">...</div>
                            <div class="afn"></div>
                            ▶<div class="aKh aPb">...</div>
                            <div></div>
                            <div class="aKB afn"></div>
                          ▼<div class="UI" gh="tl">
                            ▼<div class="aDP">
                              ▼<div class="ae4 aDM" id="21" role="tabpanel" aria-labelledby="35">
                                ▼<div class=">
                                  ▼<div class="nH Wg aAD aAz">
                                    ▶<div class="Cq">...</div>
                                    ▼<div class="Cr">
                                      ▼<div class="aAE J-J5-Ji J-JN-M-I" aria-label="Inbox section options" aria-expanded="false"
                                        role="button" tabindex="0" aria-haspopup="false" style="webkit-user-select: none;">
                                          <div class="J-J5-Ji J-JN-M-I-Jm"></div>
```

Semantiska HTML-taggar

I HTML5 definieras ett antal taggar som precis som `<div>` och `` inte har något speciellt utseende definierat men ändå har ett specifikt syfte.

- `<article>`
- `<aside>`
- `<header>`
- `<footer>`
- `<nav>`
- `<section>`
- `<time>`
- <https://developer.mozilla.org/en-US/docs/Glossary/Semantics>

Semantiska HTML-taggar



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.

Introduktion till CSS (Cascading StyleSheets)

- **CSS** är ett deklarativt språk som används för att dekorera HTML-element
- Tillsammans med HTML och JavaScript utgör det en av grundpelarna i webben
- Precis som HTML har CSS en väldigt enkel struktur

CSS-regler

En CSS-regel (CSS rule) består av två delar:

- En lista **egenskaper** (properties) som beskriver hur ett element ska ritas ut på skärmen
- En **selector** som beskriver vilka element som regeln ska gälla för.
- Egenskaperna omges av krullparanteser och avslutas med semikolon:

```
selector {  
    property1;  
    property2;  
    property3;  
}
```

CSS properties

Några vanliga CSS properties är:

background-color	Bakgrundsfärg
font-size	Textstorlek
border	Ram runt element
color	Färg på text
font	Typsnitt
text-align	Justera text

<https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>

CSS selectors

Selector	Exempel	Förklaring
element	<code>p { ... }</code>	Väljer alla <code><p></code> -element
id	<code>#main { }</code>	Ett element med <i>id="main"</i>
class	<code>.highlight</code>	Alla element med <i>class="highlight"</i>
attribute	<code>input[type="email"]</code>	Alla input element med attributet <i>type="email"</i>
pseudo-class	<code>p:hover</code>	Alla p taggar som man har muspekaren över

CSS selectors kombinationer

Kombination	Syntax	Exempel	Förklaring
Gruppering	A, B	p, h1 {}	Applicerar gemensam styling på alla p och h1 element
Descendant	A B	header p	Applicerar styling på alla p element inne i en header
Child	A > B	ul > li	Applicerar styling på alla li element som finns direkt under ett ul element

CSS Exempel

```
p {  
  color: green;  
}
```

```
table.mytable {  
  border-color: rgb(0, 0, 0);  
  border-width: 1px;  
  border-style: solid;  
}
```

```
#mylist > li {  
  width: 100px;  
}
```

Använda CSS i HTML

Det finns tre sätt att applicera CSS på element i ett HTML-dokument

- Använda en **<style>**-tagg i **<head>**-elementet
- Importera en fil med en **<link>**-tagg i **<head>**-elementet
- Direkt på ett element med **“style”**-attributet

CSS i HTML exempel

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="style.css" />
  <style>
    p {
      color: red;
    }
  </style>
</head>
<body>
  <h1 style="color: green;">Hello</h1>
  <p>World</p>
</body>
</html>
```

Övning - CSS

1. Skapa ett HTML-dokument med några paragrafer text.
2. Använd CSS för att ändra bakgrunden för sidan till någon snygg färg
3. Använd CSS för att ändra text-storleken för paragraferna
4. Omge vissa ord eller stycken med ``
5. Skriv CSS-kod som ger en distinkt bakgrund till klassen "highlight"
6. Omge flera .highlight element med `<div id="highlight-borders">`
7. Skriv CSS kod som ger en border till alla .highlight element inuti `#highlight-borders` elementet

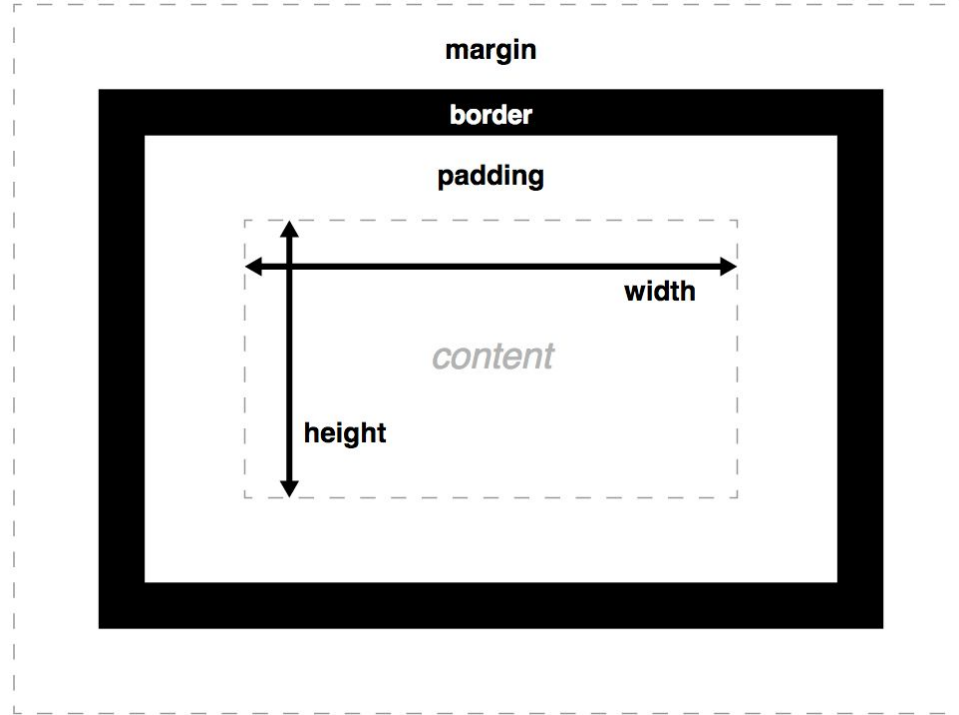
Tips: Kolla upp hur **färger** och **storlekar** definieras i CSS

CSS properties för layout

Vanliga CSS properties för layout

display	Text "inline" eller "block"
width, height	Blockstorlek
margin	Marginal
padding	Padding

CSS box model



CSS vanlig syntax

Man kan ange border, margin och padding för en enskild riktning med hjälp av: *-top*, *-right*, *-bottom*, *-left*

Exempel: **border-top** ändrar bordern enbart för toppen av ett element.

För margin & padding kan man slå ihop alla riktningar på en rad där syntaxen är:

margin: top right bottom left

(Medurs, börjar på toppen)

Exempel - **margin: 10px 20px 30px 40px**

Övningar/hemläxa

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