

Anteckningar till Videokurs

Traversy's Cram Course - A lot of goodies

Videon finns på [CSS Crash Course For Absolute Beginners - YouTube](#)

om Videon	2
Tidpunkter:	2
Ej dokumenterat här:	2
Inte i denna video	2
Mitt eget påpekande gällande Semantisk HTML & CSS	2
Intro	3
Tre olika sätt att använda style's	3
Inline - mixar presentation med styling	3
Internal - Sätter style i egen tag, i html-filen	3
External	3
CSS-selector schema:	3
En enkel uppsättning i en css-fil	4
Web safe fonts	4
Font-inställningar & genvägar	5
Radhöjd	5
Block och Inline-element	5
Block-level Elements	5
The <div> element is a block-level element	5
Inline Elements	5
The Element is an inline-level element	6
Klasser och Id	6
Skillnaden mellan klass & Id	6
Egen formatering av standardelement inom en klass	6
Ex: Denna påverkar enbart header1 inom klassen box1:	6
Bra o ha för en rubrik	6
Kanter - marginaler, border och padding	7
Box-modellen	7
Inställningar & genvägar	7
Följande tre varianter gör samma sak:	7
Nollställning av egenskaper på alla element	8
border - sätt en ram runt ett element	8
Formulär	8
Knappar och fält	8
CSS-för detta ser ut som följer	9
Slutligen:	9

om Videon

Tidpunkter:

[0:00-0:25](#) --- Intro
[0:25-2:02](#) --- Before We Begin
[2:02-2:55](#) --- What is CSS?
[2:55-3:30](#) --- What We Need To Start
[3:30-5:33](#) --- Creating a CSS file (Sublime)
[5:33-7:11](#) --- Methods For Adding CSS (mistake at [6:18](#), "Internal" should be "Inline CSS")
[7:11-11:14](#) --- Color (all 3 methods)
[11:14-13:31](#) --- CSS Selectors
[13:31-15:46](#) --- Colors in CSS
[15:46-20:00](#) --- Web Safe Fonts (sizes, weights, and height)
[20:00-24:53](#) --- Class and Id
[24:53-33:00](#) --- Box Model, Margin and Padding Shorthand, and Borders
[33:00-39:10](#) --- Text Formatting Styles (list style [38:34-39:10](#))
[39:10-44:20](#) --- Links (images [42:20](#))
[44:20-51:20](#) --- Forms (button style [48:30-51:20](#))

Ej dokumenterat här:

[51:20-57:10](#) --- Alignment and Floating
[57:10-1:03:41](#) --- Positioning in CSS
[1:03:41-1:06:20](#) --- Setting Images As Element Background
[1:06:20-1:08:55](#) --- Pseudo Classes
[1:08:55-1:25:10](#) --- Making Own Website (Outro)

Inte i denna video

- Flex-Box
- SSAS
- Practical examples
- HTML

De är dock att finna i andra video'r.

Mitt eget påpekande gällande Semantisk HTML & CSS

Många tycks använda <div> till alla behov för att skapa avdelare inom en webbsida.

Det är olyckligt, då det i o f s inte påverkar renderingen, men gör koden mycket svårare att tolka för andra. Genom att följa semantisk HTML kommer åtminstone huvuddelarna som <header> , <body> & <footer> samt delarna <main> , <section> och <article> göra koden avsevärt mycket tydligare. Därtill finns sådant som <aside> för inforutor & sådant.

Behöver man fler avdelare använder man <div> , t.ex. inom en <form> eller <article>

Heads Up: Det är stor skillnad på <head> , <header> och <heading> N.B.! :D

Intro

Tre olika sätt att använda *style's*

Inline - mixar *presentation* med *styling*

```
<h1 style="color:red; text-align:center">Hello world</h1>
```

Internal - Sätter style i egen tag, i html-filen

```
<head>
<style> h2 { color:blue; }
</head>

<body>
  <h2 >Hello world</h1>
</body>
```

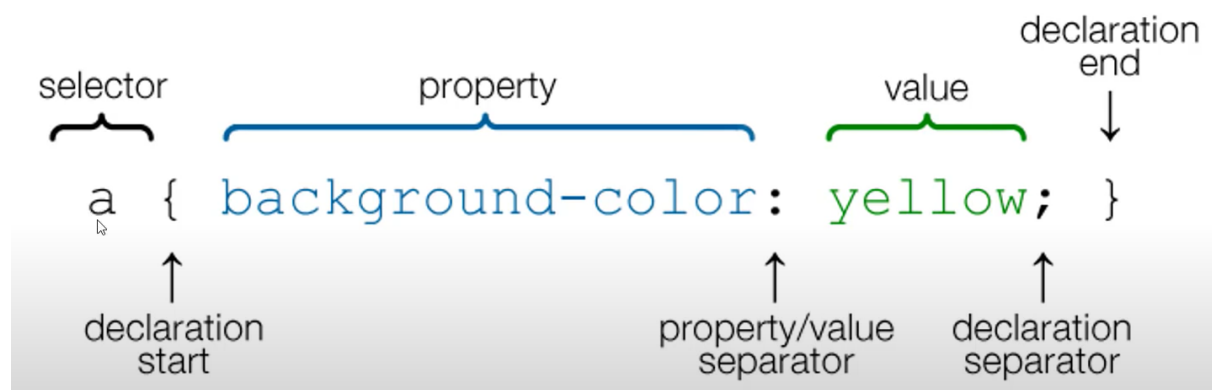
External

Samma som internal, men kan användas i hela projektet.
Anropas från HTML-sidans <head>sektion

med:

```
<link rel="stylesheet" href="style.css">
```

CSS-selector schema:



En enkel uppsättning i en css-fil

```
body {  
color: red;  
background : coral  
  
h1 {  
color: #00ff00;  
  
p {  
color: rgb(0, 0, 255);
```

Dessa är här satta i tur och ordning: Color names, HTML5 color names, Hexadecimal, RGB

Web safe fonts

De här finns "överallt" och är fria från belastningar som copyright m.m.

<i>Examples</i>	<i>Font Families</i>
font-family	Arial, Helvetica, sans-serif
font-family	"Arial Black", Gadget, sans-serif
font-family	"Bookman Old Style", serif
font-family	"Comic Sans MS", cursive, sans-serif
font-family	Courier, monospace
font-family	"Courier New", Courier, monospace
font-family	Garamond, serif
font-family	Georgia, serif
font-family	Impact, Charcoal, sans-serif
font-family	"Lucida Console", Monaco, monospace
font-family	"Lucida Sans Unicode", "Lucida Grande", sans-serif
font-family	"MS Sans Serif", Geneva, sans-serif
font-family	"MS Serif", "New York", sans-serif
font-family	"Palatino Linotype", "Book Antiqua", Palatino, serif
φοντ-φαμιλψ	Σψμβολ, σανς-σεριφ
font-family	Tahoma, Geneva, sans-serif
font-family	"Times New Roman", Times, serif
font-family	"Trebuchet MS", Helvetica, sans-serif
font-family	Verdana, Geneva, sans-serif

Font-inställningar & genvägar

```
body {  
font-family: Arial, Helvetica, sans-serif;  
font-size : 16px;  
font-weight : normal;  
}
```

Kan skrivas tillsammans så här:

```
body {  
font: normal 14px Arial, Helvetica, sans-serif;  
}
```

Radhöjd

```
Line-height:1.5em;
```

ger lite lagom space mellan raderna i ett stycke.

Block och Inline-element

Bra förklaring från W3Schools: ⇨ [HTML Block and Inline Elements](#)

Every HTML element has a default display value, depending on what type of element it is.

There are two display values: **block** and **inline**.

Block-level Elements

A block-level element always starts on a new line.

A block-level element always takes up the full width available (stretches out to the left and right as far as it can).

A block level element has a top and a bottom margin, whereas an inline element does not.

The <div> element is a block-level element

The <div> element is often used as a container for other HTML elements.

The <div> element has no required attributes, but style, class and id are common.

When used together with CSS, the <div> element can be used to style blocks of content

Inline Elements

An inline element does not start on a new line.

An inline element only takes up as much width as necessary.

The Element is an inline-level element

The element is an inline container used to mark up a part of a text, or a part of a document.

The element has no required attributes, but style, class and id are common.

When used together with CSS, the element can be used to style parts of the text

Klasser och Id

Skillnaden mellan klass & Id

En klass kan användas till flera instanser, medans ett ID används unikt för ett enda element.

html:

```
<div class="myStdFormat">
</div>
...
<div class="myStdFormat">
<a id="myIngress">
</div>
```

CSS:

```
.myStdFormat {
    color:red;
}

#myIngress {
    font-size : 18pt;
}
```

Egen formatering av standardelement inom en klass

Ex: Denna påverkar enbart header1 inom klassen box1:

```
. box-1 h1{
    font-family : Tahoma;
}
```

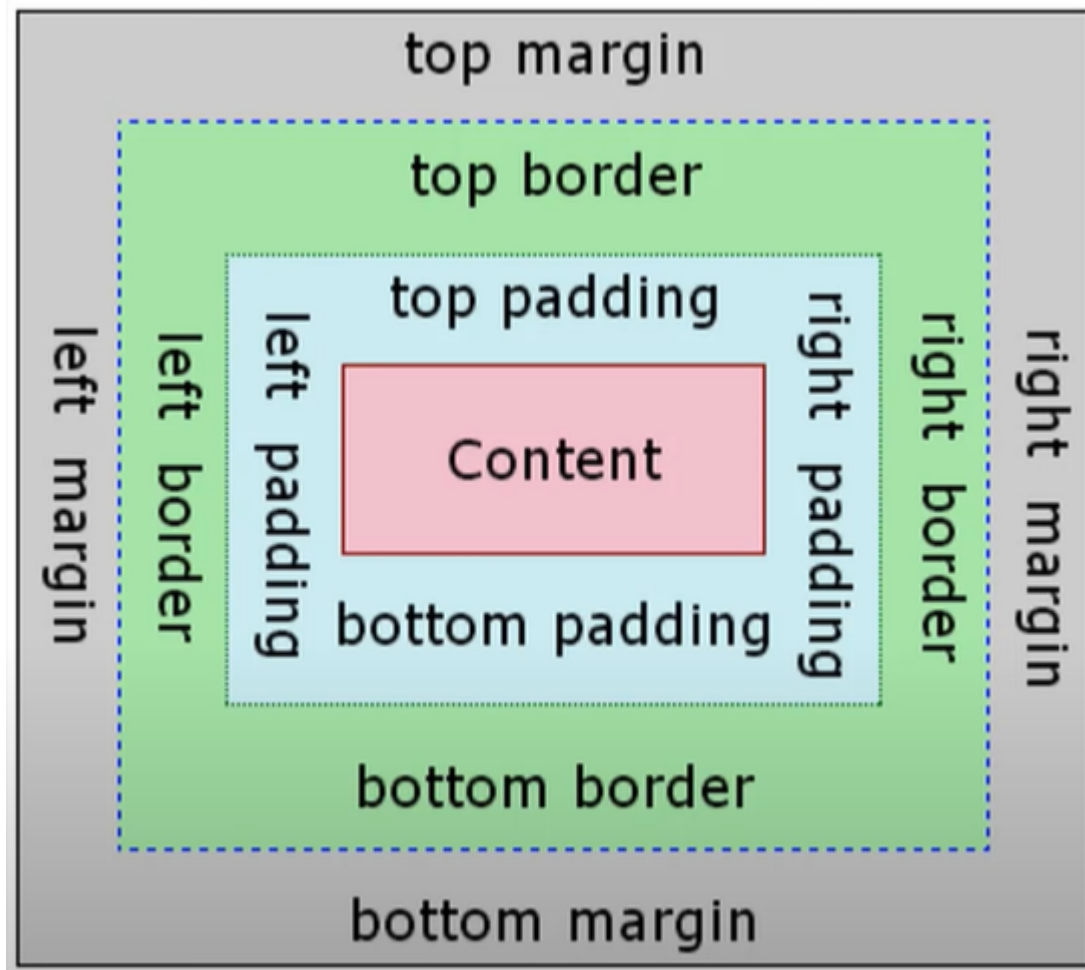
Bra o ha för en rubrik

```
.element1 {
    font-weight : 800;
    font-style: italic;
    text-decoration : underline;
    text-transform : uppercase;
    Letter-spacing: 0.2em;
    word-spacing: 1em;
}
```

Kanter - marginaler, border och padding

Box-modellen

Den ger svar på var och hur man skapar avstånd mellan elementen på sidan.



Inställningar & genvägar

Följande tre varianter gör samma sak:

```
p {  
margin-top:5px;  
margin-bottom:5px;  
margin-right: 10px;  
margin-left:10px;  
}
```

```
p {  
margin: 5px 10px 5px 10px;  
}
```

```
p {  
margin: 5px 10px;  
}
```

Nollställning av egenskaper på alla element

'*' är en identifierare som gäller alla element, så

```
* {  
  margin:0;  
}
```

kommer att ta bort alla default-marginaler och lämna till dig att sätta istället (vilket är bra vid testning och inläring, för att se vad exakt vad du styr över) samma sak gäller padding:0 .

border - sätt en ram runt ett element

```
box-1 {  
  /*alt 1 *  
  border-right : 5px red solid;  
  border- left: 5px red solid;  
  border-top : 5px red solid;  
  border-bottom: 5px red solid;  
  
  /* alt 2 - samma effekt kortare */  
  border: 5px red solid;  
  
  border-width : 3px;  
  border-bottom-width : 10px;  
  border-top-style : dotted;  
  padding-top : 20px;  
  background-color : #333;  
  color:#fff;  
}
```

Formulär

Knappar och fält

Denna kod:

```
<form>  
  <div">  
    <label>Name: </label>  
    <input type="text" name="name">  
  </div>  
  <div>  
    <label>Email: </label>  
    <input type="text" name="email">  
  </div>  
  <div>  
    <label>Message: </label>  
    <textarea name="message"></textarea>  
  </div>  
  <input type="submit" value="Submit" name="bnSubmit">  
</form>
```

genererar följande utseende:

Name:

Email:

Message:

Genom att applicera CSS kan man snygga till detta radikalt :

Name:

Email:

Message:

Etiketten på egen rad ovanför

```
.my-form label{
  display:block;
```

CSS-för detta ser ut som följer

```
.button{
  background-color:rgb(46, 26, 26);
  color:#fff;
  padding:10px 15px;
  border:none;
}

.button:hover{
  background:rgba(255, 0, 0, 0.5);
  color:#fff;
}

.my-form{
  padding:50px;
}
```

```
.my-form .form-group{
  padding-bottom:25px;
}

.my-form label{
  display:block;
}

.my-form input[type="text"], .my-form
textarea{
  padding:8px;
  width:100%;
}
```

Slutligen

Som synes har jag inte dokumenterat sista halvtimmen (ännu), för att den ändå måste ses som mer avancerad än vad denna skrift avhandlar.

Inte desto mindre är det "bra skit" som Brad lär ut och tiderna finns ju i förteckningen, så ... "go ahead! & make the day ;)"