

RWD

Responsive Web Design

Agenda

- *Layouts med CSS*
- *Vad är Responsive Web Design?*
- *Hur gör vi en webbplats responsiv?*
- *Utvecklingsmiljö*

Layouts med CSS

- *Semantiska element*
- *Boxmodellen*
- *Flexbox*
- *Grid*

Terminologi: Container (förälder) och Item (barn)

Semantiska element

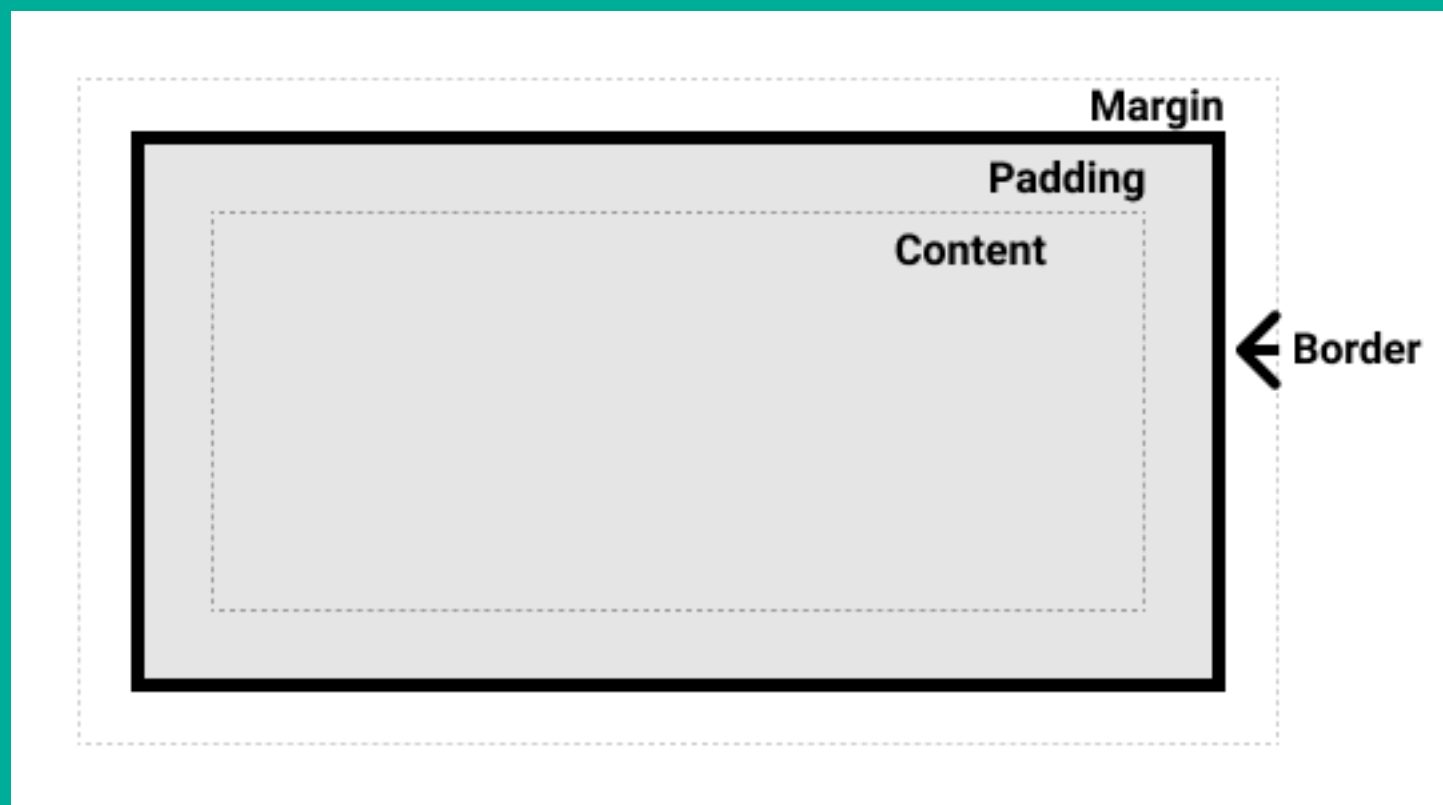
- *Ger innebörd till det markerade innehållet*
- *Tidigare användes huvudsakligen **<div>** för att gruppera*
- *Nu finns element som: **main, nav, footer, article***
- *Kombinera semantiska element med **<div>***

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

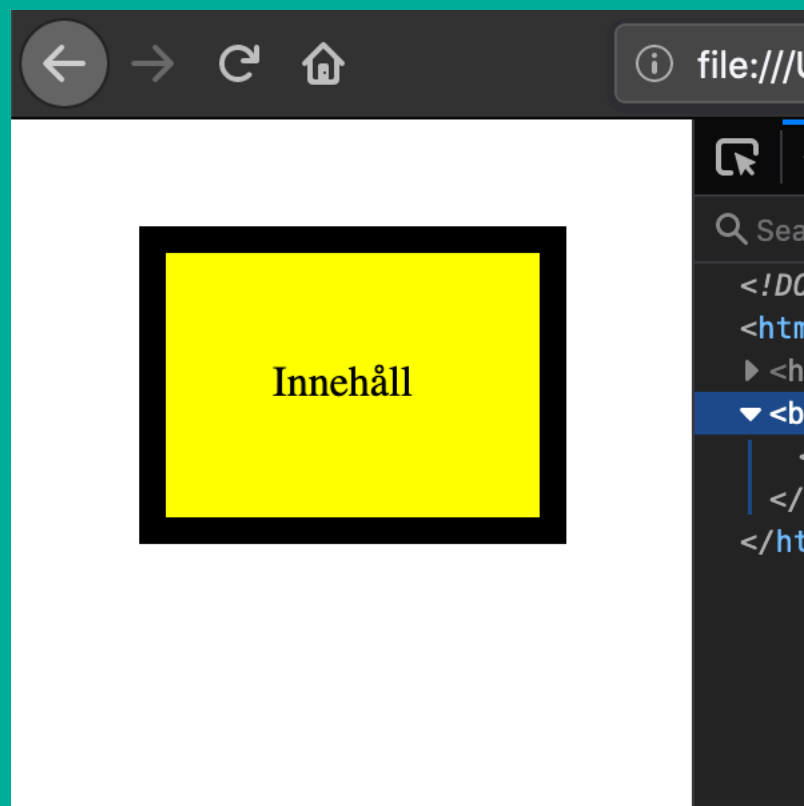
Boxmodellen

- *Består av: innehåll, margin, padding och border*
- *Blockelement styrs av boxmodellen*
- *Attributet **box-sizing** styr vilken modell som används*
 - *Värden: **content-box** eller **border-box***
- *Innehållet är vad som styr höjd och bredd*

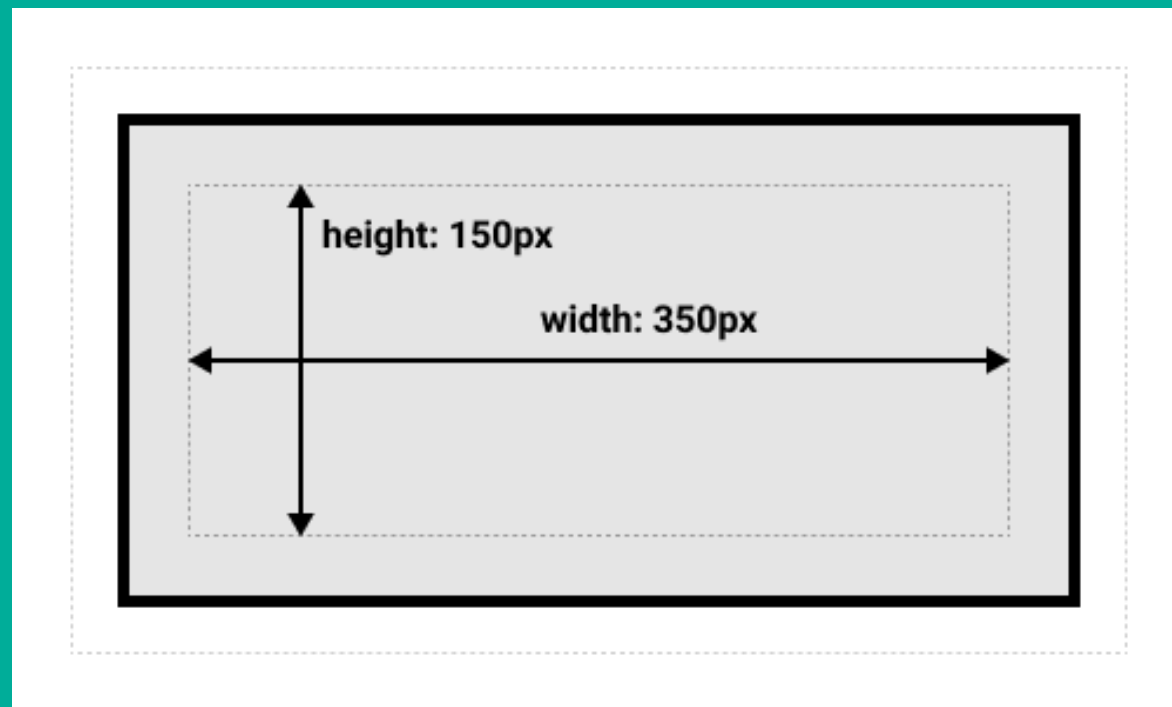
Boxmodell



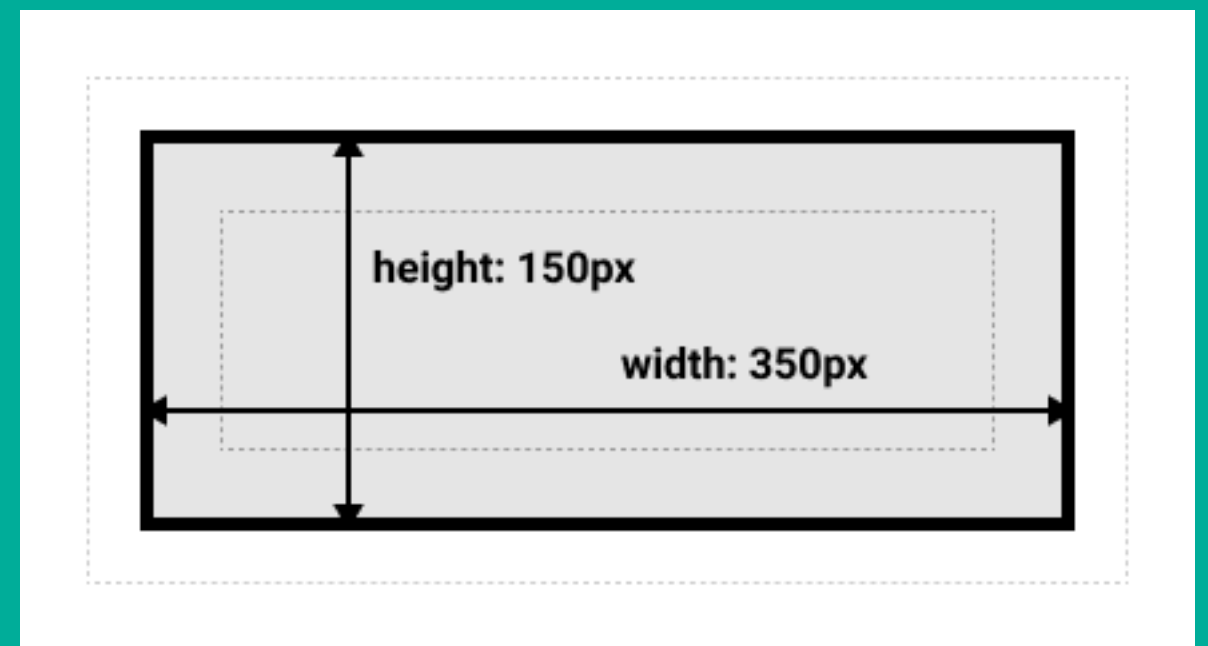
Boxmodellen



Boxmodell



*Box-sizing: content-box;
Standard*



Box-sizing: border-box;

Flexbox

- *1-dimensionell layout*
- *Bra för att strukturera innehåll horisontellt eller vertikalt*
- *Anger en **container** vars **items** ska positioneras*

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

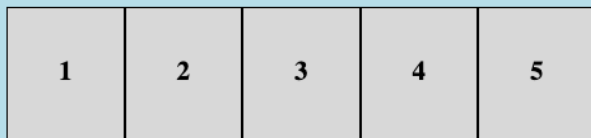
<https://flexboxfroggy.com>

Flexbox

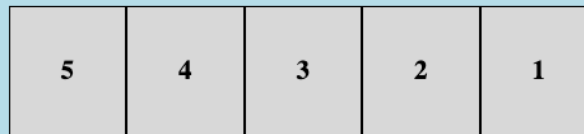
1
2
3
4
5

```
9
10 <body>
11   <main>
12     <div>1</div>
13     <div>2</div>
14     <div>3</div>
15     <div>4</div>
16     <div>5</div>
17   </main>
18 </body>
19
```

Flexbox

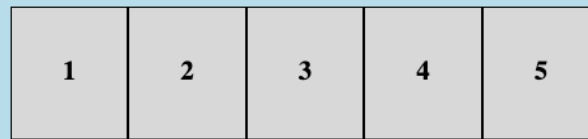


```
1  
2 main {  
3   display: flex;  
4 }  
5
```



```
1  
2 main {  
3   display: flex;  
4   flex-direction: row-reverse;  
5 }  
6
```

Flexbox



```
1  
2 main {  
3   display: flex;  
4   justify-content: center;  
5 }  
6
```



```
1  
2 main {  
3   display: flex;  
4   justify-content: space-between;  
5 }  
6
```

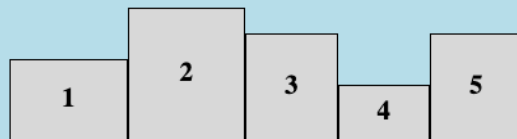


```
1  
2 main {  
3   display: flex;  
4   justify-content: space-around;  
5 }  
6
```

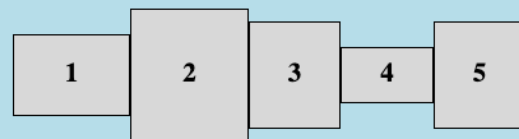
Flexbox



```
1  
2 main {  
3   display: flex;  
4   align-items: flex-start;  
5 }  
6
```

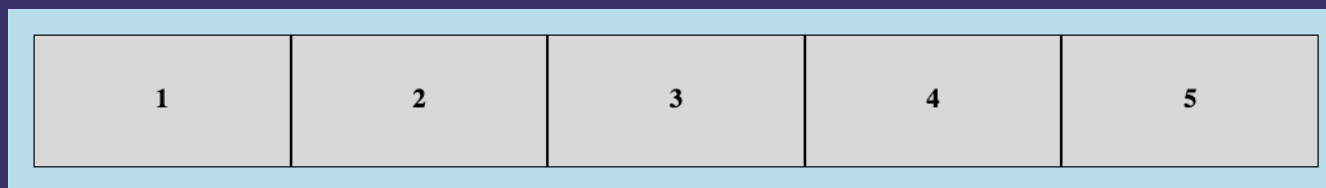


```
1  
2 main {  
3   display: flex;  
4   align-items: flex-end;  
5 }  
6
```



```
1  
2 main {  
3   display: flex;  
4   align-items: center;  
5 }  
6
```

Flexbox



```
1  
2 main {  
3   display: flex;  
4 }  
5  
6 div {  
7   flex: 1;  
8 }  
9
```

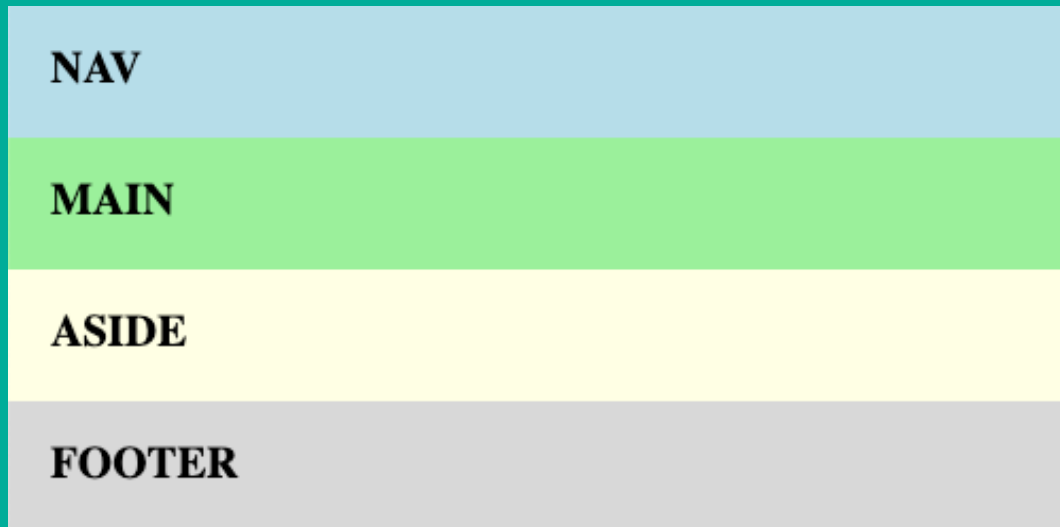
Grid

- *2-dimensionell layout*
- *Bra för att strukturera sektioner både horisontellt och vertikalt*
- *En **container** anger ytor i ett **rutnät***
- *Respektive **item** kan sedan välja att **fylla en eller flera rutor***

<https://css-tricks.com/snippets/css/complete-guide-grid/>

<http://cssgridgarden.com>

Grid



```
9
10 <body>
11     <nav>
12         NAV
13     </nav>
14
15     <main>
16         MAIN
17     </main>
18
19     <aside>
20         ASIDE
21     </aside>
22
23     <footer>
24         FOOTER
25     </footer>
26 </body>
27
```


Grid



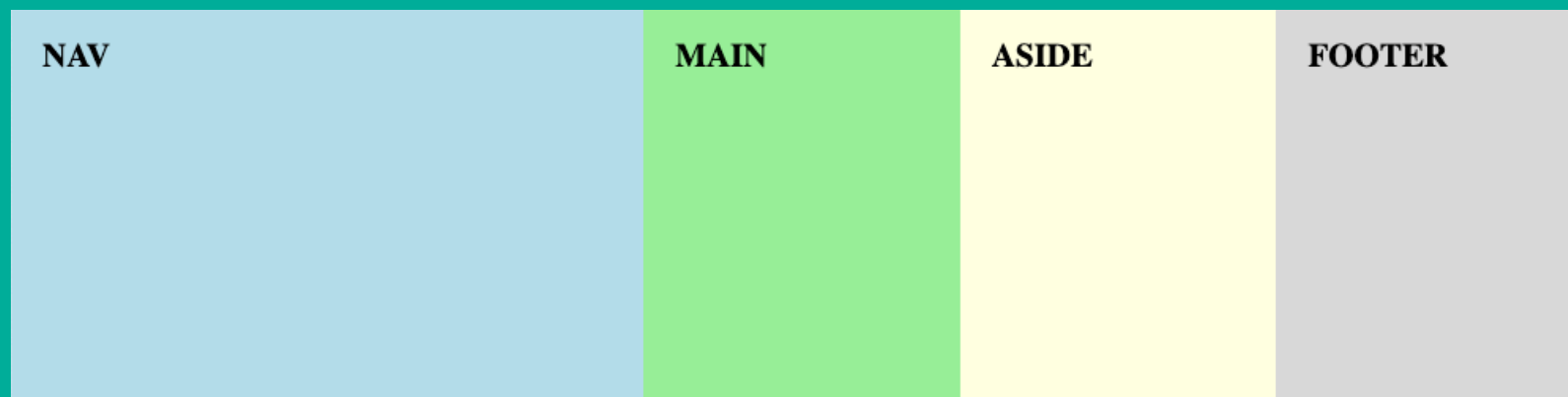
```
1
2 body {
3   display: grid;
4   grid-template-columns: 1fr 1fr 1fr 1fr 1fr;
5   grid-template-rows: 100px auto 150px;
6   grid-template-areas:
7     "nav nav nav nav nav"
8     "main main main aside aside"
9     "footer footer footer footer footer";
10 }
11
12 nav {
13   grid-area: nav;
14 }
15 main {
16   grid-area: main;
17 }
18 aside {
19   grid-area: aside;
20 }
21 footer {
22   grid-area: footer;
23 }
24
```

Grid



```
1
2 body {
3   display: grid;
4   grid-template-columns: 1fr 1fr 1fr 1fr 1fr;
5   grid-template-rows: 100px auto 150px;
6   grid-template-areas:
7     "nav nav nav nav nav"
8     "main main main aside aside"
9     "footer footer footer footer footer";
10 }
11
12 nav {
13   grid-area: nav;
14 }
15 main {
16   grid-area: main;
17 }
18 aside {
19   grid-area: aside;
20 }
21 footer {
22   grid-area: footer;
23 }
24
```

Grid



```
1  
2 body {  
3   display: grid;  
4   grid-template-columns: 1fr 1fr 1fr 1fr 1fr;  
5   grid-template-rows: 100px auto 150px;  
6   grid-template-areas:  
7     "nav nav main aside footer"  
8     "nav nav main aside footer"  
9     "nav nav main aside footer";  
10 }  
11
```

Vad är RWD?

- *Myntades 2010 av Ethan Marcotte*
 - *<https://alistapart.com/article/responsive-web-design/>*
- *Tidigare hade vi separata webbplatser för mobiler*
- *Adaptiv kontra Responsiv design*
- *Varför är det viktigt med en reponsiv design?*
- *Vad bör vi tänka på?*

Adaptiv vs Responsiv

- *Har tidigare tolkats som antingen eller*
- *En webbplats utnyttjar båda aspekter*
- *Responsiv, innehållet anpassas efter enheten*
- *Adaptiv, känner av funktioner och anpassar sidan därefter*
 - *Till exempel, pixeldensitet, landscape och portrait, print*

Varför är det viktigt?

- *Besök sker via en stor mängd olika enheter*
 - *Varpå en större andel sker via mobiler (sedan 2016)*
- *Google indexerar sk. **mobile first***
- *Upplevelsen är viktigare idag*
- *Kan nå ut till en större massa*

Vad bör vi tänka på?

- *Användbarhet och tillgänglighet*
 - *Ska inte behöva zooma*
 - *Ingen horisontell scroll*
 - *Ytor för att kunna klicka på knappar och länkar*
 - *Med mera*
- *Mediainnehåll (bild, ljud, video)*
- *Mobile, tablet, laptop, desktop*
- *Mobile first*
- *Inspiration av andra, <https://mediaqueri.es>*

Hur gör vi?

- *Media queries*
- *Breakpoints*
- *Viewport*
- *Enheter*
- *Specificity*

Media Queries

- *Mediatyper: all, print, screen*
- *Gruppera regler inom @media*
- `<link rel="stylesheet" href="print.css" media="print">`

Media Queries

```
1
2 /* Gäller under 1250px i sidbredd */
3 @media (max-width: 1250px) {}
4
5 /* Gäller över 1250px i sidbredd */
6 @media (min-width: 1250px) {}
7
8 /* Endast skärmar */
9 @media screen and (max-width: 320px) {}
10
11 /* Orientation */
12 @media (max-width: 320px) and (orientation: portrait) {}
13
14 /* Print */
15 @media print {}
16
```

Media Queries

```
1
2 .info-box {
3   width: 700px;
4 }
5
6 /* Endast skärmar */
7 @media screen and (max-width: 600px) {
8   .info-box {
9     width: 100%;
10  }
11 }
12
```

Breakpoints

- *Specifika punkter (bredd) där media queries tillämpas*
- *Varierar i och med att nya telefoner och surfplattor*
- *Till exempel:*
 - *Telefoner - 600px*
 - *Tablet (portrait) - 900px*
 - *Tablet/Laptop - 1200px*
 - *Desktop - 1800px*

Viewport

- *Används för att styra layout på mobiler*
- *Kan styra hur inzoomad användaren är med **initial-scale***
- *Kan styra om användaren får zooma med **user-scalable***
- *Med **width=device-width** så täcker innehållet mobilfönstret*

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Enheter

- *På mobiler tenderar vi att nyttja procent (%) för bredd*
 - *Tips: **max-width***
- *Viewport height (**vh**) och width (**vw**)*
 - *Till exempel: **100vh** = 100% av fönstrets bredd*
- *Viewport minimum (**vmin**) och maximum (**vmax**)*
 - *Nästan samma som **vh** och **vw** men tar istället det lägsta (**vmin**) eller högsta (**vmax**) värdet av fönstrets höjd eller bredd*

Specificity

- *Styr vilken regel som kommer appliceras till ett element*
- *Den högsta summan av antal selectors mellan två regler styr*
- *Är bådas summa densamma gäller den senaste regeln*
- *Kan skriva över med hjälp av **!important***
- *<https://specificity.keegan.st>*

Specificity

Specificity Calculator

Sort by specificity

A visual way to understand [CSS specificity](#). Change the selectors or paste in your own.

nav > a:hover::before

- 0 IDs
- 1 Classes, attributes and pseudo-classes
- 3 Elements and pseudo-elements

+ Duplicate

ul#primary-nav li.active

- 1 IDs
- 1 Classes, attributes and pseudo-classes
- 2 Elements and pseudo-elements

+ Duplicate

Specificity Calculator was built by [Keegan Street](#). The [specificity calculator JavaScript module](#) is available on GitHub or via `npm install specificity`.

<https://specificity.keegan.st>

Specificity

```
1
2 .main-area p.info-box {
3   font-size: 18px;
4 }
5
6 .info-box {
7   font-size: 12px !important;
8 }
9
10
```

Utvecklingsmiljö

- *Testa olika skärmstorlekar*
- *Testa olika förutsättningar (uppkoppling, m.m.)*

Frågor?