
CSS

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

https://www.w3.org/Style/CSS/Overview.en.html

CSS rule

```css
p {
  color: lightsalmon;
}
```

```css
selector {
  property: value /* css declaration */
}
```

A property paired with a value is called a **CSS declaration**. *CSS declarations are put within CSS Declaration Blocks. And finally, CSS declaration blocks are paired with selectors to produce* **CSS Rulesets (or CSS Rules)**.

https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/Syntax

```html
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>CSS linking!</title>
    <!-- relation: interpret as style sheet -->
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Goodby world!</h1>
  </body>
</html>
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>CSSing</title>
    <style>
      h1 {
        color: lightsalmon;
      }
    </style>
  </head>
  <body>
    <h1>Goodbye World!</h1>
  </body>
</html>
```

Internal stylesheet

Får användas när:

- Kritiskt CSS som måste laddas snabbt
- Du kan inte redigera CSS-filen direkt

Inline style

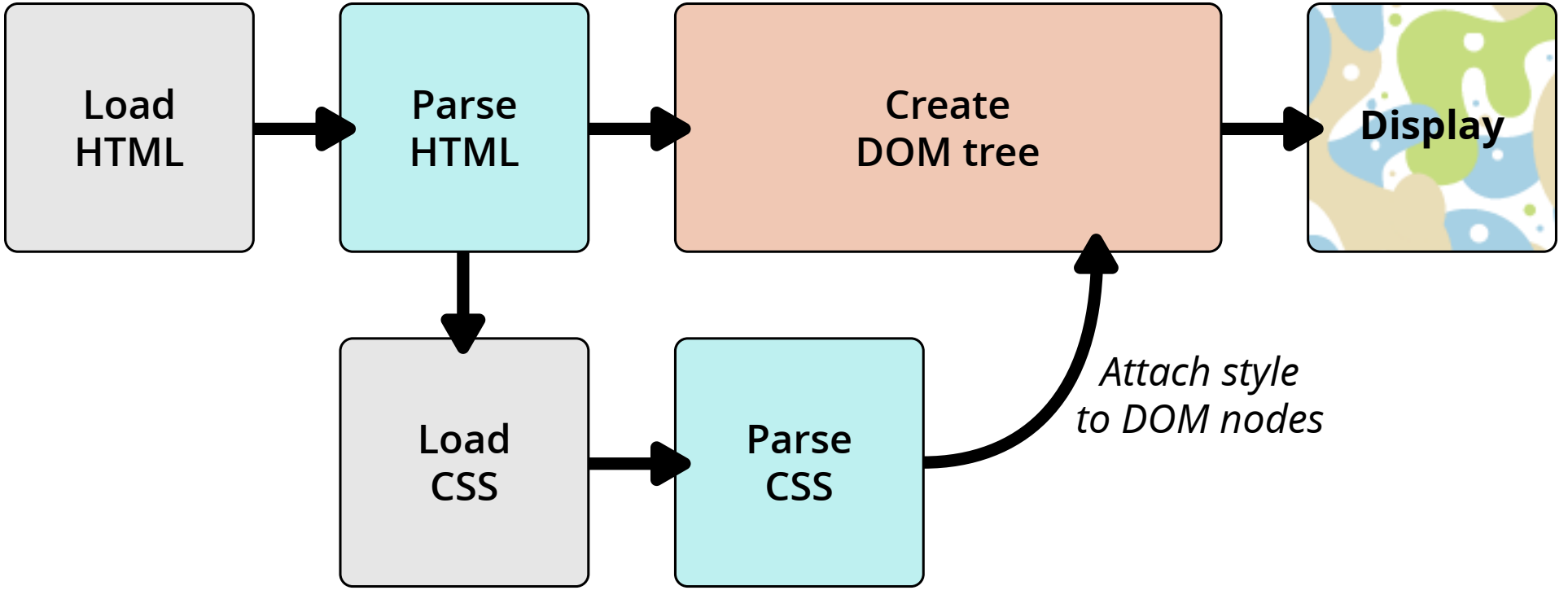
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>😤😠😑</title>
  </head>
  <body>
    <h1 style="color: lightsalmon;">Goodbye World!</h1>
  </body>
</html>
```

Inline style

Good practice att separera struktur från styling i olika filer

Får användas när:

- Sista utväg när inget annat alternativ finns
- Ibland när css ska ändras med JavaScript


Load HTML
Parse HTML
Create DOM tree
Display
Load CSS
Parse CSS
Attach style to DOM nodes

Dave Rupert
@davatron5000

Följ

Svar till @Real_CSS_Tricks

CSS is easy. It's just a few thousand key value pairs that have quirks in each browser that you have to memorize.

Översätt från engelska

18:32 - 10 juni 2017

997 Retweetar **2 079** gilla-markeringar

https://twitter.com/davatron5000/status/873578585394696192


SELECTOR

Tre primära sätt att koppla CSS till HTML

- `<tag>`
- `class`
- `id`

```
<p> My P </p>
<p class="paragraph"> My P </p>
<p id="paragraph"> My P </p>
```

```
tag {
  /* only name means tag */
 }
.class {
  /* dot means class */
}
#id {
  /* hashtag means id */
}
```

Styla främst med `class` och `tag`

`id` är främst för JavaScript

Inline style är ?

Inherit

a process whereby something, typically information or knowledge, is successively passed on.

```css
p {
  color: teal;
  font-size: 24px;
}

p {
  color: lightsalmon;
}
```

`font-size` står kvar, men den "senaste" ändringen används, color skrivs över

```html
<div>
  <p> Muh P </p>
</div>
```

```css
div {
  color: lightsalmon;
}
```

färgen ärvs till barnet

```html
<div>
  <p> Muh P </p>
</div>
```

```css
div p {
  color: lightsalmon;
}
```

Alla p som ligger i div

Vilket värde som används baseras på 3 saker

- Importance
- Specificity
- Source order

`!importance`

Sista utvägen

Använd helst inte alls

```
.full-width {
  margin: 0px !important;
}
```

I princip omöjig att skriva över, vi måste ha möjligheten att skriva över värden

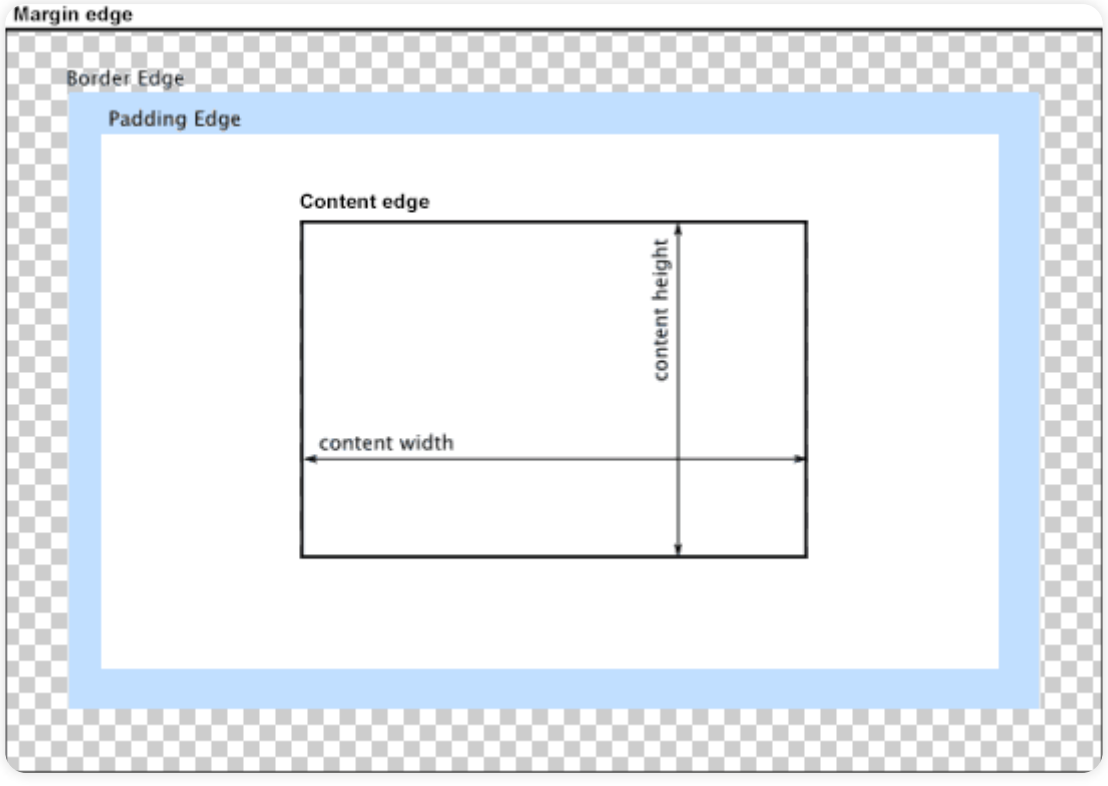
Specificity

https://specificity.keegan.st/

Styla med `class` helst

Använd `ID` främst för t.ex. JavaScript

BOXMODELLEN


Margin edge
Border Edge
Padding Edge
Content edge
content height
content width

Shorthands

```
margin-top: 1rem;
margin-right: 1rem;
margin-bottom: 1rem;
margin-left: 1rem;
```

```
margin: top right bottom left;
```

```
margin: 1rem 1rem 1rem 1rem;
```

```
margin: top/bottom left/right;
```

```
margin: 1rem 2rem;
```

```
margin: all;
```

```
margin: 1rem;
```

Collapsing margins

När två `margins` går emot varandra försvinner den ena

Som när vi t.ex. har `margin-bottom` på det övre elementet och `margin-top` på det undre

https://css-tricks.com/what-you-should-know-about-collapsing-margins/

HTML CSS Result

EDIT ON CODEPEN

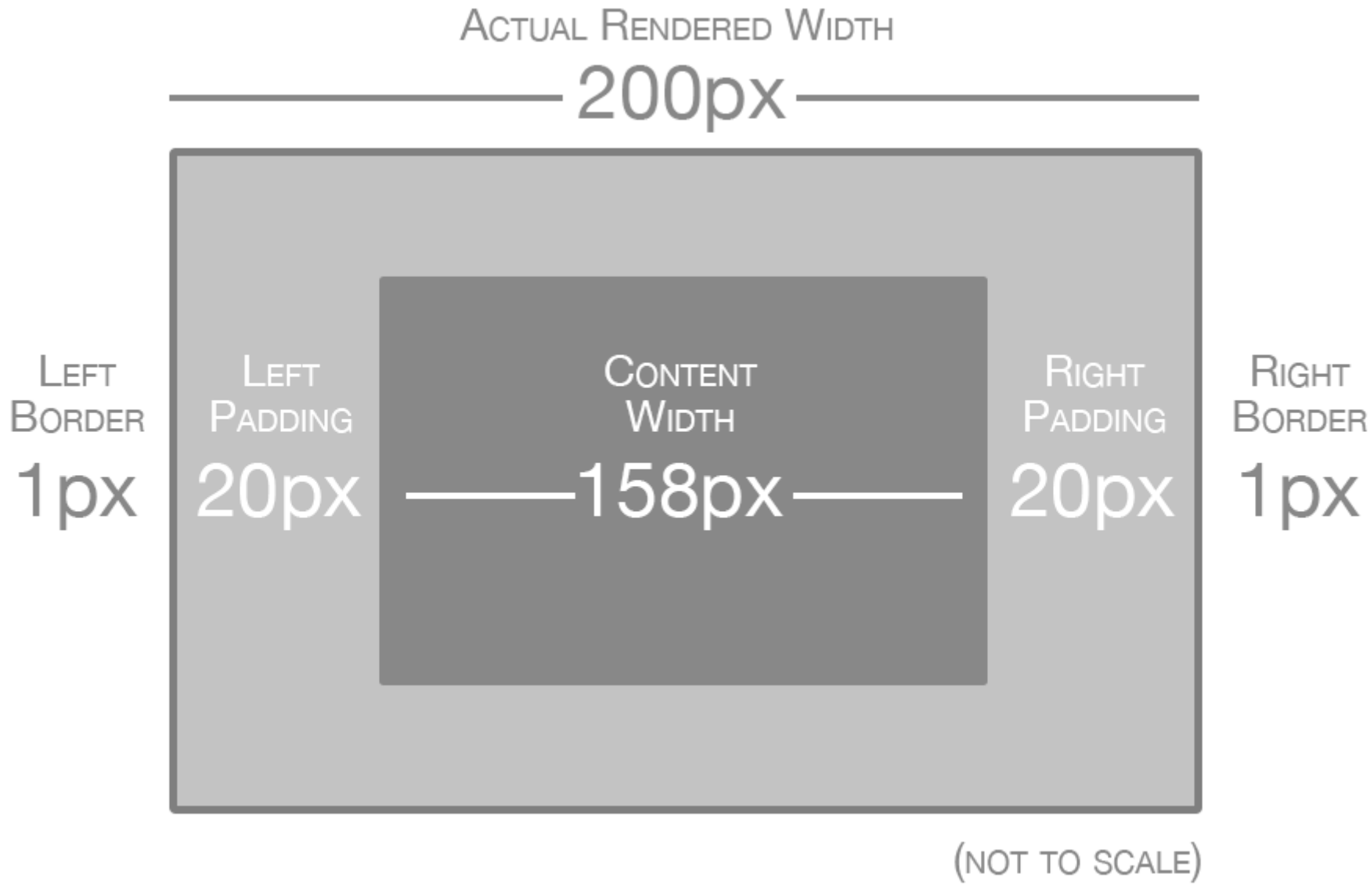
```
  height:150px;
}

.module__top {
  margin-bottom: 25px;
  background-color: #ff6b6b;
}

.module__bottom {
  margin-top: 50px;
  background-color: #4e97cd;
}
```

Boven

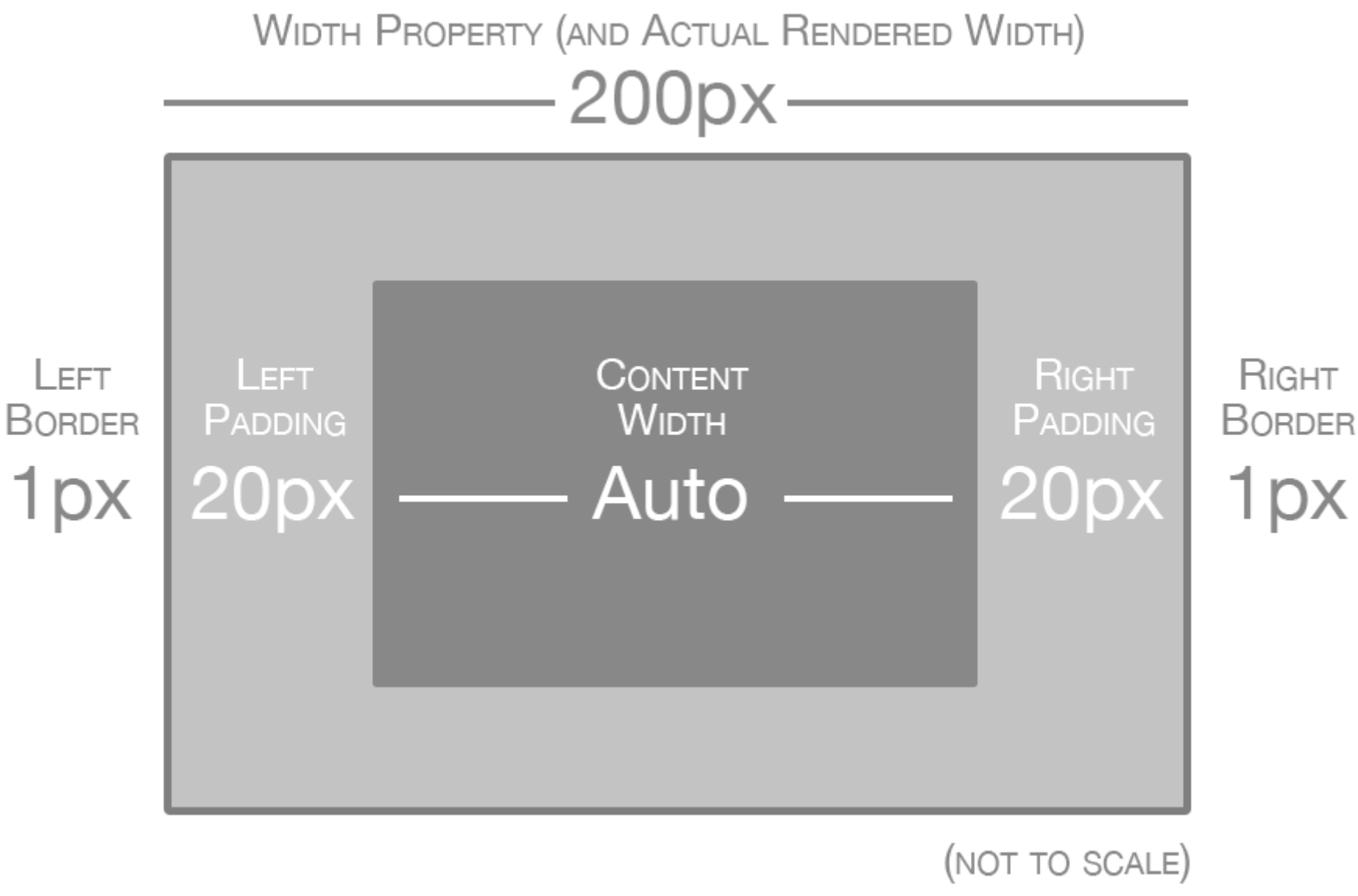
```
* {
  box-sizing: border-box;
}
```

http://blog.teamtreehouse.com/box-sizing-secret-simple-css-layouts

Boxmodellen enlight CSS Specifikationen



Boxmodellen enlight Internet Explorer 🤡



Samt om vi använder:

```
* {
  box-sizing: border-box;
}
```

POSITION

```
 /* relative to itself */
position: relative
```

```
 /* absolute pixel value */
position: absolute
```

```
 /* absolute viewport pixel value */
position: fixed
```

CSS UNITS

Allting översätts i slutändan till pixlar

Computed values

Men vi ska försöka använda relativa enheter

```css
article {
  height: 100px;  /* actual pixels */
  height: 100%;   /* height of parent */
  height: 100rem; /* 1 * default browser font-size */
  height: 100em;  /* 1 * current font-size */
  height: 100vh;  /* 100% of viewport */
  height: 100cm;  /* centimeter, not recommended */
  height: 100in;  /* inches, not recommended */
}
```

Använd främst `%`, rem samt `em`


FLOAT

The float CSS property specifies that an element should be placed along the left or right side of its container, allowing text and inline elements to wrap around it. The element is removed from the normal flow of the web page, though still remaining a part of the flow.

https://developer.mozilla.org/en-US/docs/Web/CSS/float

CSS Level 2 (Revision 1) The definition of 'float' in that specification.	REC Recommendation	No change
CSS Level 1 The definition of 'float' in that specification.	REC Recommendation	Initial definition

Cascading Style Sheets, level 1

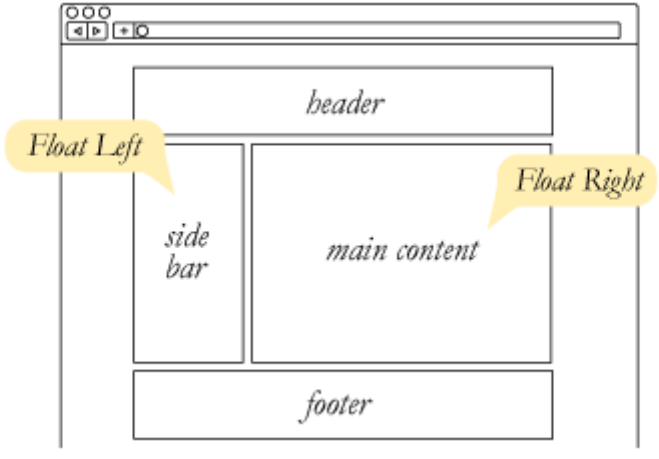
W3C Recommendation 17 Dec 1996, revised 11 Apr 2008

```css
.box {
  float: left;
}
```

```css
.box {
  float: right;
}
```

```css
.box {
  float: none;
}
```

Låt elementet ligga bredvid även fast det är `display: block`


header
Float Left
Float Right
side
bar
main content
footer

Element kan fortsätta flyta

Vi måste ibland stoppa flödet

```
.element-after-float {
  clear: both; /* both sides */
}
```

```
.element-after-float {
  clear: left; /* only left side clear */
}
```

```
.element-after-float {
  clear: left; /* only right side clear */
}
```

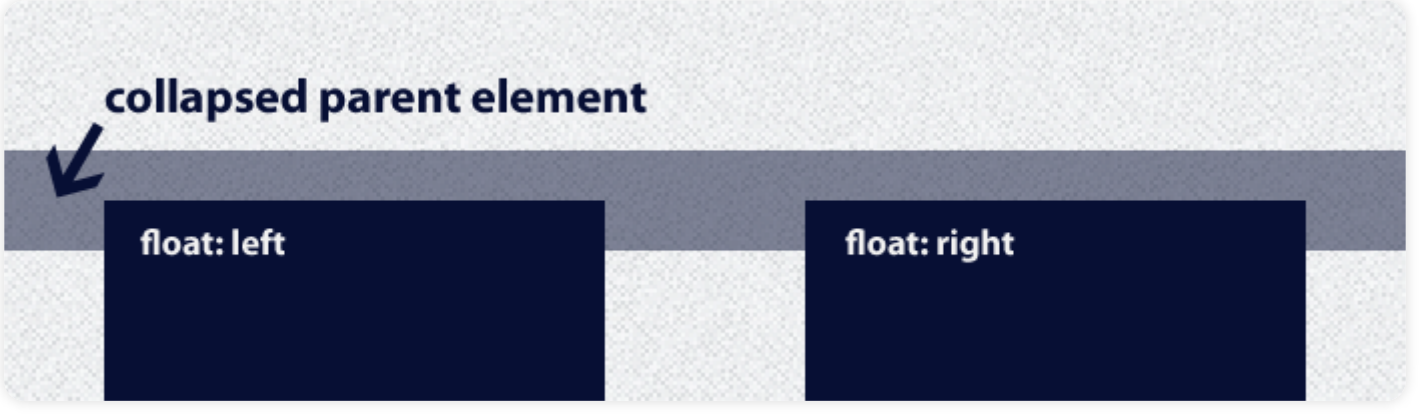
"Hacket"

```
.clearfix {
  clear: both;
}
```

Klassen används på ett element som kommer efter ett element som flyter. Stoppar kommande element från att flyta på.

Det blir systemkollaps

Om ett förälderelement enbart innehåller element som flyter kommer föräldern tappa sin höjd



Använd i sådana fall clearfix

```
<div class="parent">
  <div class="child"></div>
  <div class="child"></div>
  <div class="clearfix"></div>
</div>
```