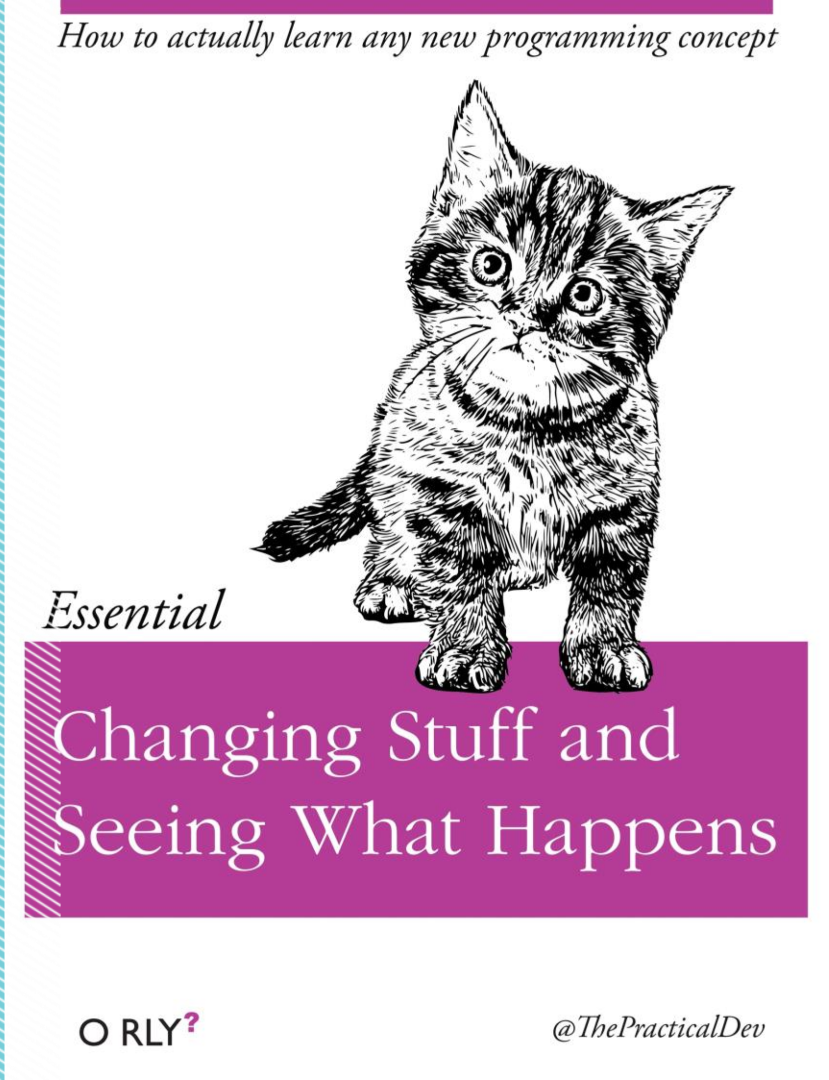
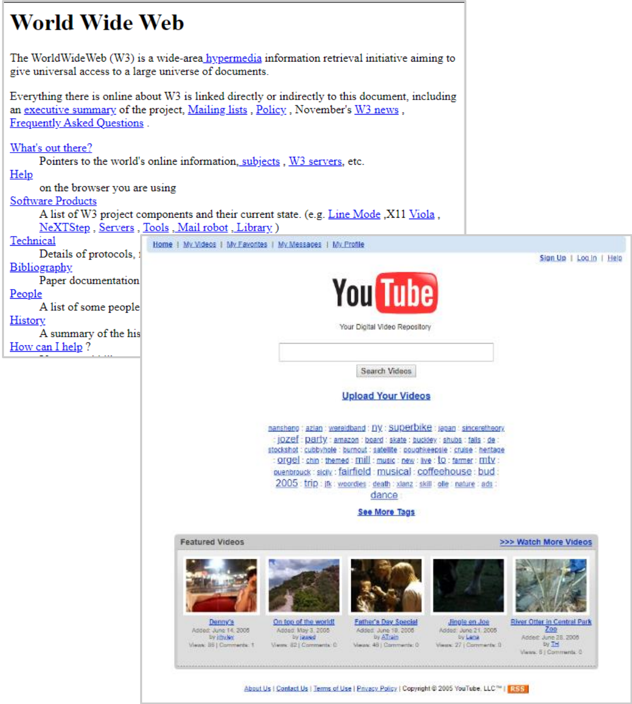
CSS Basics

(one does not simply / center a div)

Frontend Junior Program - 2022


How to actually learn any new programming concept
Essential
Changing Stuff and Seeing What Happens
O RLY?
@ThePracticalDev

CSS… Why?



World Wide Web

The WorldWideWeb (W3) is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an executive summary of the project, Mailing lists , Policy , November's W3 news , Frequently Asked Questions .

What's out there?
Pointers to the world's online information, subjects , W3 servers, etc.
Help
on the browser you are using
Software Products
A list of W3 project components and their current state. (e.g. Line Mode ,X11 Viola , NeXTStep , Servers , Tools , Mail robot , Library)
Technical
Details of protocols,
Bibliography
Paper documentation
People
A list of some people
History
A summary of the his
How can I help ?

Home | My Videos | My Favorites | My Messages | My Profile
Sign Up | Log In | Help
YouTube
Your Digital Video Repository
Search Videos
Upload Your Videos
nansheng : azlan : wereldband : ny : superbike : japan : sinceretheory : jozef : party : amazon : board : skate : buckley : shubs : falls : de : stockshot : cubbyhole : burnout : satellite : poughkeepsie : cruise : heritage : orgel : chin : themed : mill : music : new : live : to : farmer : mtv : ouenbrouck : sicily : fairfield : musical : coffeehouse : bud : 2005 : trip : jfk : woordjes : death : xlanz : skill : ollie : nature : ads : dance :
See More Tags
Featured Videos
>>> Watch More Videos

Denny's
Added: June 14, 2005
by jdrylex
Views: 86 | Comments: 1

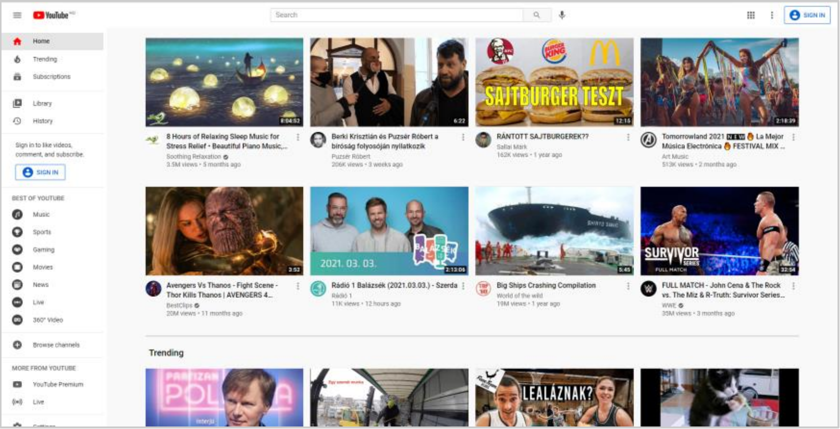
On top of the world!
Added: May 3, 2005
by jawed
Views: 82 | Comments: 0

Father's Day Special
Added: June 19, 2005
by ATrain
Views: 46 | Comments: 0

Jingle en Joe
Added: June 21, 2005
by Lena
Views: 27 | Comments: 0

River Otter in Central Park Zoo
Added: June 28, 2005
by Tri
Views: 5 | Comments: 0

About Us | Contact Us | Terms of Use | Privacy Policy | Copyright © 2005 YouTube, LLC™ | RSS



8 Hours of Relaxing Sleep Music for Stress Relief - Beautiful Piano Music,...
Soothing Relaxation
3.5M views • 5 months ago

Berki Krisztián és Puzsér Róbert a bíróság folyosóján nyilatkozik
Puzsér Róbert
206K views • 3 weeks ago

RÁNTOTT SAJTBURGEREK??
Sallai Márk
162K views • 1 year ago

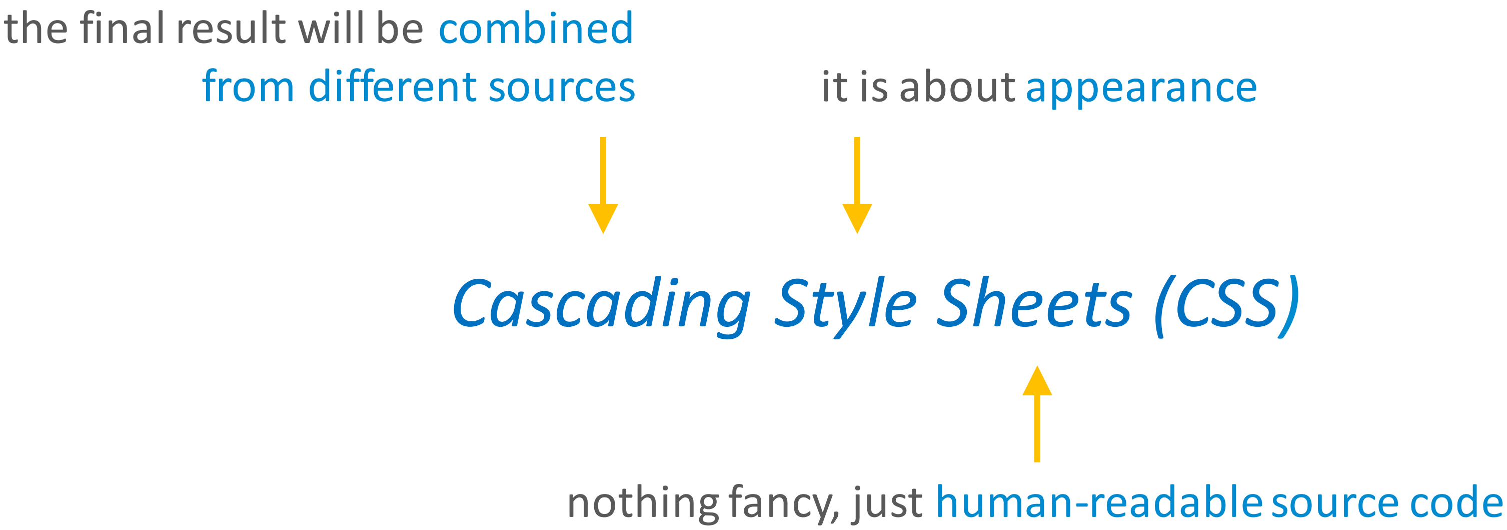
Tomorrowland 2021 NEW 🔥 La Mejor Música Electrónica 🔥 FESTIVAL MIX ...
Art Music
513K views • 2 months ago

Avengers Vs Thanos - Fight Scene - Thor Kills Thanos | AVENGERS 4...
BestClips
20M views • 11 months ago

Rádió 1 Balázsék (2021.03.03.) - Szerda
Rádió 1
11K views • 12 hours ago

Big Ships Crashing Compilation
World of the wild
19M views • 1 year ago

FULL MATCH - John Cena & The Rock vs. The Miz & R-Truth: Survivor Series...
WWE
35M views • 3 months ago



the final result will be combined from different sources

it is about appearance

Cascading Style Sheets (CSS)

nothing fancy, just human-readable source code

CSS is a standard

links to documentation,
please explore!

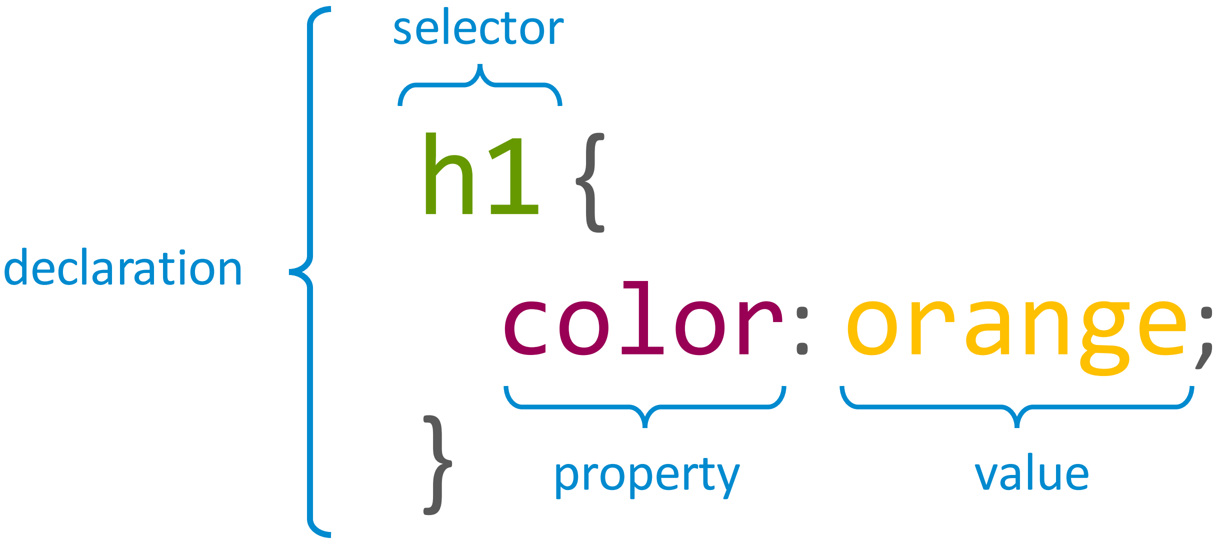
“Cascading Style Sheets (CSS) is a ~~simple~~ mechanism for adding style (e.g., fonts, colors, spacing) to Web documents.”

CSS describes how elements should be displayed on screen
(or on paper / in speech / on other media).

CSS has standardized W3C specifications
CSS1 - W3C Recommendation 1996
CSS2.1 - W3C Recommendation 2011
CSS3 - is split into smaller modules, progressing on the standardization track.

SYNTAX

Basic syntax



selector

```
h1 {
    color: orange;
}
```

declaration

property

value

Basic syntax

Order matters

```
p {
  color: green;
}

p {
  color: lime;
}
```

the color of p will be lime

Comments in CSS-file

```
/*
This is a comment
that can span one
or multiple lines
*/

div {
  width: 200px; /* Another comment here */
}

// definitely NOT a comment
```

⚠

Adding styles to the page - external styles

```html
<!DOCTYPE HTML>
<html>
<head>
    <meta charset="utf-8">
    <title>Styles</title>
    <link rel="stylesheet" href="css/style.css">
</head>
<body>
    <h1>Heading</h1>
    <p>Content</p>
</body>
</html>
```

style.css

```css
h1 {
  color: #000080;
  font-size: 200%;
  text-align: center;
}

p {
  padding: 20px;
  background: yellow;
}
```

Internal styles

```
<!DOCTYPE HTML>
<html>
<head>
    <meta charset="utf-8">
    <title>Internal styles</title>
    <style>
        h1 {
           font-size: 120%;
           font-family: Verdana, Arial, Helvetica, sans-serif;
           color: #333366;
        }
    </style>
</head>
<body>
    <h1>Hello, world!</h1>
</body>
</html>
```

Inline styles

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8" />
        <title>Inline styles</title>
    </head>
    <body>
        <h1 style="font-size: 120%; color: #cd66cc">Hello, world!</h1>
    </body>
</html>
```

Importing styles

```
@import ’imported.css'

h1 {
  color: #000080;
  font-size: 200%;
  text-align: center;
}
```

imported.css:

```
body {
  background: #fc0;
}

p {
  font-weight: bold;
}
```

How to add styles to the page

All described methods of using CSS can be used either alone or in combination with each other.

When use in combination, it is necessary to remember their hierarchy:

- Inline style - highest priority
- Internal style, external style - lower priority

Absolute units

cm	centimeters
mm	millimeters
in	inches (1in = 2.54cm)
px	pixels
pt	points (1pt = 1/72 of 1in)
pc	picas (1pc = 12 pt)

Relative units

em	Relative to the font-size of the element (2em means 2 times the size of the current font)
ex	Relative to the x-height of the current font (rarely used)
ch	Relative to width of the "0" (zero)
rem	Relative to font-size of the root element
vw	Relative to 1% of the width of the viewport
vh	Relative to 1% of the height of the viewport
vmin	Relative to 1% of viewport's smaller dimension
vmax	Relative to 1% of viewport's larger dimension

https://developer.mozilla.org/en-US/docs/Web/CSS/length#units

Property values

```css
/* Strings: */
li:before {
  content: 'Hello'
}

/* Numbers: */
p {
  font-weight: 600; line-height: 1.2;
}

/* URLs: */
a {
  background: url(warn.png) no-repeat
}

/* Keywords: */
p {
  text-align: right;
}
```

Color:

- By hexadecimal values: #6609CF, #fc0
- By name: white, silver, black, lightblue, ...
- RGB: rgb(255, 0, 0)
- RGBA: rgba(0,255,0,0.3)
- HSL: hsl(120,100%, 25%)
- HSLA: hsla(120,100%, 50%, 0.3)

MEDIA TYPES

Specify media type: @import

```
@import 'style-screen.css' screen;
@import 'style-print.css' print, speech;

h1 {
  color: #000080;
  font-size: 200%;
  text-align: center;
}
```

all	Used for all media type devices
print	Used for printers
screen	Used for computer screens, tablets, smart-phones etc.
speech	Used for screen readers that "read" the page out loud

Media type using media queries

```css
@media screen {
  body {
    font-family: Arial, Verdana, sans-serif;
    color: #333;
  }
}

@media print {
  body {
    font-family: Times, 'Times New Roman', serif;
    color: #000;
  }
}
```

Media type in HTML “media” attribute

```
<html>
  <head>
    <meta charset="utf-8" />
    <title>Devices</title>
    <link media="print, handheld" rel="stylesheet" href="print.css" />
    <link media="screen" rel="stylesheet" href="main.css" />
  </head>
  <body>
    <p>...</p>
  </body>
</html>
```

SELECTORS

Basic selectors

ID

`#firstname`

Selects the element with id="firstname"

```css
#firstname {
  width: 520px;
  padding: 100px;
  background: #fc0;
}
```

Class

`.intro`

Selects all elements with class="intro"

```css
.intro {
  font-size: 11px;
}
```

Type

`p`

Selects all <p> elements

```css
p {
  text-align: right;
  font-size: 1.5rem;
}
```

Universal

`*`

Selects all elements

```css
* {
  font-size: 11px;
}
```

Combinators

```
div, p
```

Groups of selectors

A comma-separated list of selectors represents the union of all elements selected by each of the individual selectors in the list

```
div p
```

Descendant combinator

Selects all <p> elements inside <div> elements

```
div > p
```

Child combinator

Selects all <p> elements where the parent is a <div> element

```
div + p
```

Adjacent sibling combinator

Selects all <p> elements that are placed immediately after <div> elements

```
p ~ ul
```

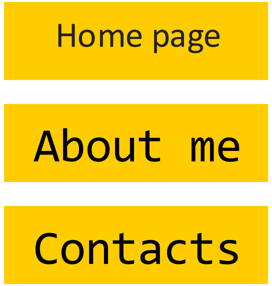
General sibling combinator

Selects every <ul> element that are preceded by a <p> element

Descendant combinator

```css
.main-nav {
  margin: 0;
  padding: 0;
  list-style: none;
}

.main-nav li {
  margin: 0 0 10px 0;
  padding: 3px;
  background: #fc0;
}

.main-nav a {
  color: #000;
}
```

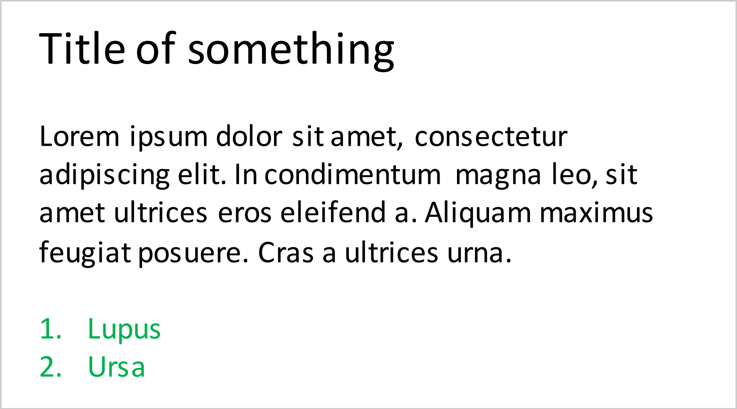
```html
<ul class="main-nav">
  <li><a href="#">Home page</a></li>
  <li><a href="#">About me</a></li>
  <li><a href="#">Contacts</a></li>
</ul>
```



Child combinator

```
ol > li {
  color: rgb(0, 255, 0);
}
```

```
<h2 class="headline">Title of something</h2>
<p>Lorem ipsum dolor sit amet...</p>
<ol>
  <li>Lupus</li>
  <li>Ursa</li>
</ol>
```



Title of something

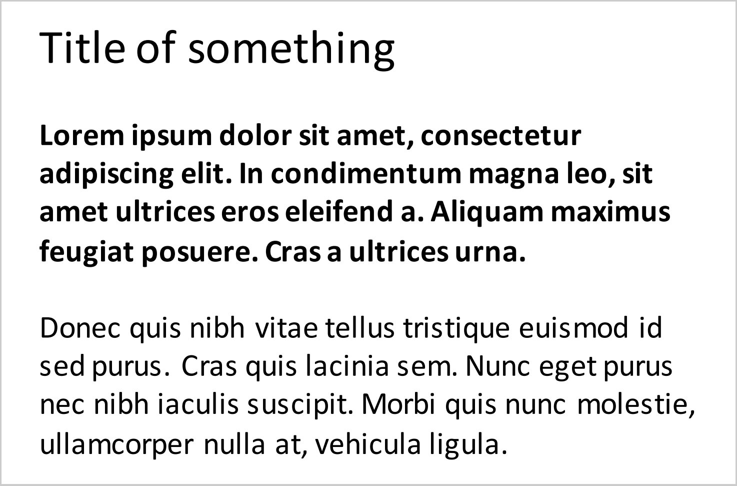
Lorem ipsum dolor sit amet, consectetur adipiscing elit. In condimentum magna leo, sit amet ultrices eros eleifend a. Aliquam maximus feugiat posuere. Cras a ultrices urna.

1. Lupus
2. Ursa

Adjacent sibling combinator

```css
.headline + p {
  font-weight: bold;
}
```

```html
<h2 class="headline">Title of something</h2>
<p>Lorem ipsum dolor sit amet...p>
<p>Donec quis nibh vitae...</p>
```



Title of something

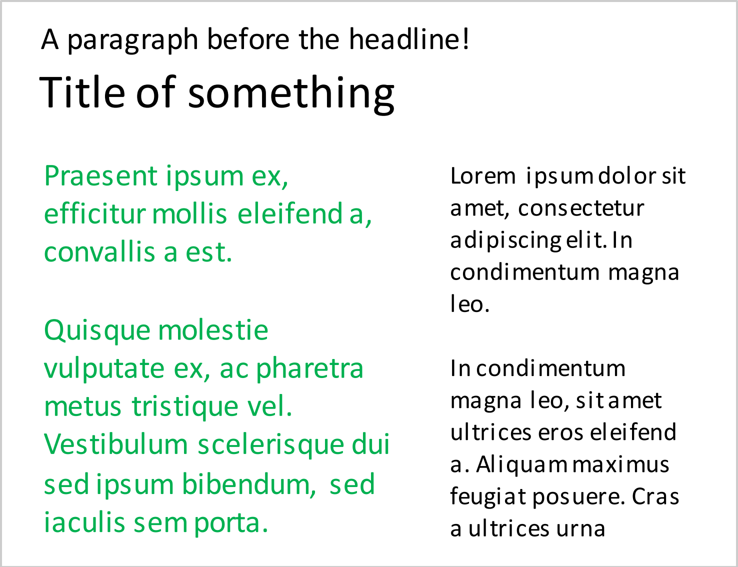
Lorem ipsum dolor sit amet, consectetur adipiscing elit. In condimentum magna leo, sit amet ultrices eros eleifend a. Aliquam maximus feugiat posuere. Cras a ultrices urna.

Donec quis nibh vitae tellus tristique euismod id sed purus. Cras quis lacinia sem. Nunc eget purus nec nibh iaculis suscipit. Morbi quis nunc molestie, ullamcorper nulla at, vehicula ligula.

General sibling combinator

```
.headline ~ p {
  color: rgb(0, 255, 0);
}
```

```
<p>A paragraph before the headline!<p>
<h2 class="headline">Title of something</h2>
<aside>
  <p>Lorem ipsum dolor sit amet...</p>
  <p>In condimentum magna leo...</p>
</aside>
<p>Praesent ipsum ex, efficitur...</p>
<p>Quisque molestie vulputate ex...</p>
```



A paragraph before the headline!

Title of something

Praesent ipsum ex, efficitur mollis eleifend a, convallis a est.

Quisque molestie vulputate ex, ac pharetra metus tristique vel. Vestibulum scelerisque dui sed ipsum bibendum, sed iaculis sem porta.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. In condimentum magna leo.

In condimentum magna leo, sit amet ultrices eros eleifend a. Aliquam maximus feugiat posuere. Cras a ultrices urna

Attribute presence and value selectors

[target]

Selects all elements with a "target” attribute (with any value)

[target="_blank"]

Selects all elements with target="_blank" (exact value)

[title~="flower"]

Selects elements with a title attribute containing the word "flower"

[lang|="en"]

Selects all elements with a lang attribute value starting with "en"

Substring matching attribute selectors

a[href*="w3c"]

Selects <a> elements with href attribute value containing the substring "w3c"

a[href^="https"]

Selects <a> elements with href attribute value starting with "https"

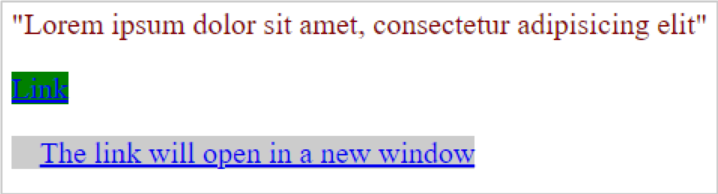
a[href$=".pdf"]

Selects <a> elements with href attribute value ending with ".pdf"

Attribute selectors

```
[title] {
  color: maroon;
}

a[href] {
  background: green;
}

a[target="_blank"] {
  background: #ccc;
  padding-left: 15px;
}
```

```
<blockquote title="Some title text">Lorem ipsum dolor
 sit amet, consectetur adipisicing elit</blockquote>
<p>
  <a href="#link">Link</a>
</p>
<p>
  <a href="#link" target="_blank">The link will open
in a new window</a>
</p>
```



"Lorem ipsum dolor sit amet, consectetur adipisicing elit"

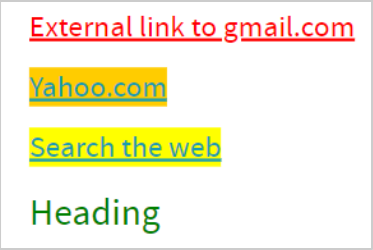
Link

The link will open in a new window

Advanced attributes selectors

```css
/* Attribute value starts with some text */
a[href^="http://"] {
  color: red;
}

/* If link ends with ".com" */
a[href$=".com"] {
  background: #fc0;
}

/* If link contains "google" */
[href*="google"] {
  background: yellow;
}

/* One of the several attribute values */
[title~="block"] {
  color: green;
}
```

```html
<p>
  <a href="http://gmail.com" target="_blank">
     External link to gmail.com</a>
</p>
<p>
  <a href="http://www.yahoo.com">Yahoo.com</a>
</p>
<p>
  <a href="http://google.com.ua">Search the web</a>
</p>
<h3 title="block tag">Heading</h3>
```



External link to gmail.com

Yahoo.com

Search the web

Heading

CASCADING

Cascading

- **Origin and Importance**
- **Scope**
- **Specificity**
- **Order of Appearance**

Cascading

Cascading refers simultaneous use of different style rules to elements
by connecting multiple style files,
inheritance of properties and other methods.

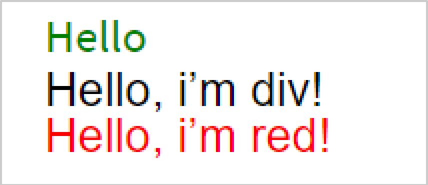
The higher style rule is placed in this list,
the lower its priority and vice versa:

1. Browser’s style
2. Author’s style
3. The author's style adding !important

Inheritance

```css
body {
  color: green;
}

div {
  color: black;
}

div.red {
  color: red;
}
```

```html
Hello
<div>Hello, i’m div!</div>
<div class="red">Hello, i’m red!</div>
```



Hello
Hello, i’m div!
Hello, i’m red!

CSS Specificity

a

1 x element selector

Sith power: 0,0,1

p a

2 x element selectors

Sith power: 0,0,2

.foo

1 x class selector *

Sith power: 0,1,0

a.foo

1 x element selector
1 x class selector

Sith power: 0,1,1

p a.foo

2 x element selectors
1 x class selector

Sith power: 0,1,2

.foo .bar

2 x class selectors

Sith power: 0,2,0

p.foo a.bar

2 x element selectors
2 x class selectors

Sith power: 0,2,2

#foo

1 x id selector

Sith power: 1,0,0

a#foo

1 x element selector
1 x id selector

Sith power: 1,0,1

.foo a#bar

1 x element selector
1 x class selector
1 x id selector

Sith power: 1,1,1

.foo .foo #foo

2 x class selectors
1 x id selector

Sith power: 1,2,0

style

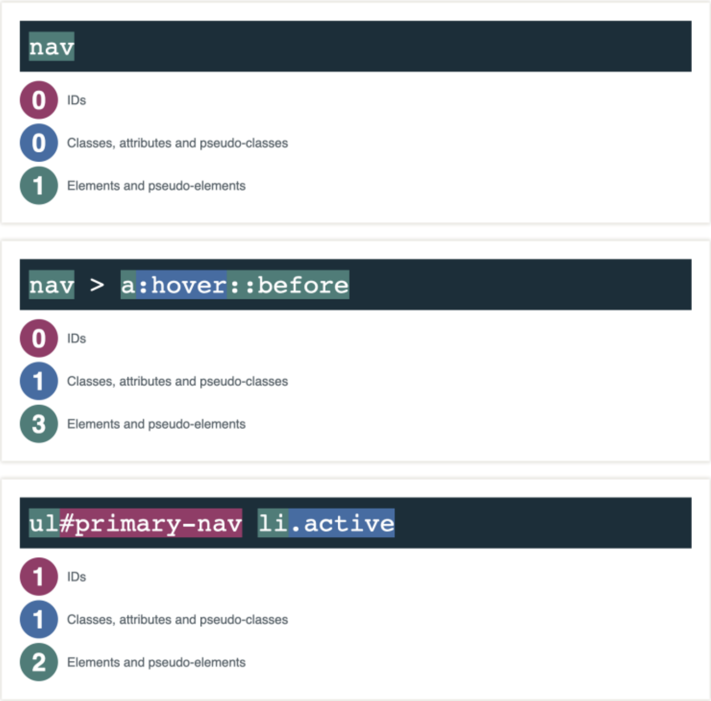
1 x style attribute

Sith power: 1,0,0,0

*** Same specificity**
class selector =
attribute attribute =
pseudo-classes

!important

CSS Specificity



```
nav
```

0 IDs

0 Classes, attributes and pseudo-classes

1 Elements and pseudo-elements

```
nav > a:hover::before
```

0 IDs

1 Classes, attributes and pseudo-classes

3 Elements and pseudo-elements

```
ul#primary-nav li.active
```

1 IDs

1 Classes, attributes and pseudo-classes

2 Elements and pseudo-elements

[Specificity Calculator (keegan.st)]

!important

!important declaration overrides any other CSS declaration.

```
selector {
  property: property value !important;
}
```

```
<!DOCTYPE html>
<html>

  <head>
    <meta charset="utf-8">
    <title>Document</title>
    <style>
      #one {
        color: red;
      }

      #two {
        color: blue !important; /* No overrides for this */
      }
    </style>
  </head>

  <body>
    <p id="one" style="color: yellow;">First paragraph</p>
    <p id="two" style="color: yellow;">Second paragraph.</p>
  </body>

</html>
```

!important

You can use *!important* (it is not an anti-pattern), when your goal is that the property should not be overridden.

Never use *!important* to override other rules!
Do use selectors with proper specificity to achieve that.

PSEUDO CLASSES AND ELEMENTS

Pseudo-classes

The pseudo-class concept is introduced to permit selection based on information that lies outside of the document tree or that cannot be expressed using the other simple selectors.

```css
a:hover {
  border-bottom: 1px solid;
  background: #CDFEAA;
}
```

Dynamic pseudo-classes

Link

:link

```
a:link {
  color: #265301;
}
```

:visited

```
a:visited {
  color: #0000ff;
}
```

User action

:hover

```
a:hover {
  border-bottom: 1px solid;
  background: #CDFEAA;
}
```

:active

```
a:active {
  background: #265301;
  color: #CDFEAA;
}
```

:focus

```
a:focus {
  border-bottom: 1px solid;
  background: #BAE498;
}
```

Other pseudo-classes

UI element

:enabled

```
input:enabled {
  background: #ccc;
}
```

:disabled

```
input:disabled {
  background: #ccc;
}
```

:checked

```
:checked {
  margin-left: 25px;
  border: 1px solid blue;
}
```

Negation (matches-none)

:not(selector)

```
a:not([target]){
  color: #000;
}
```

Matches-any

:is(selector)

```
a:is(ul, ol) > li {
  color: #000;
}
```

(Some) structural pseudo-classes

Selector	Example	Example description
:first-child	p:first-child	Selects every <p> element that is the first child of its parent
:first-of-type	p:first-of-type	Selects every <p> element that is the first <p> element of its parent
:last-child	p:last-child	Selects every <p> element that is the last child of its parent
:last-of-type	p:last-of-type	Selects every <p> element that is the last <p> element of its parent
:nth-child(n)	p:nth-child(2)	Selects every <p> element that is the second child of its parent
:nth-last-child(n)	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child
:nth-of-type(n)	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent

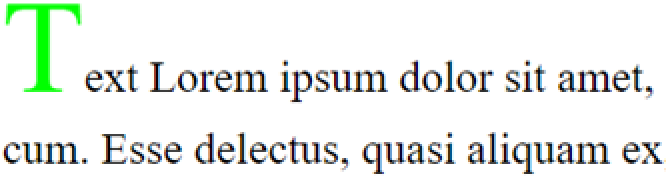
Pseudo-classes

```
a:link {
  color: #036;  /* The color of not visited links */
}
a:hover {
  color: #f00;  /* The color of links on mouse pointer hovering */
}
a:visited {
  color: #606;  /* The color of visited links */
}
a:visited:hover {
  color: #303;  /* The color of not visited links on hover */
}
a:active {
  color: #ff0;  /* The color of active links */
}
b:first-child {
  color: red;  /* The color of the first tag */
}
b:last-child {
  color: green; /* The color of the last tag */
}
```

Pseudo-elements

Selector	Example	Example description
::after	p::after	Insert something after the content of each <p> element
::before	p::before	Insert something before the content of each <p> element
::first-letter	p::first-letter	Represents the first letter of an element
::first-line	p::first-line	Describes the contents of the first formatted line of an element

```css
p::first-letter {
  color: lime;
  font-size: 300%
}
```

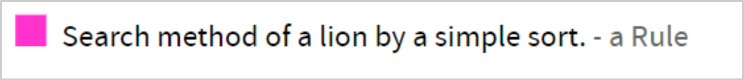


Text Lorem ipsum dolor sit amet,
cum. Esse delectus, quasi aliquam ex

Pseudo-elements

```
p:before {
  content: "";
  display: inline-block;
  width: 20px;
  height: 1em;
  margin-right: 10px;
  background: #f3c;
}
p:after {
  content: " - a Rule";
  color: #666;
}
```

```
<p>Search method of a lion by a simple sort.</p>
```



Search method of a lion by a simple sort. - a Rule

Groups of selectors

```css
h1, h2, h3 {
  font-family: Arial, Helvetica, sans-serif;
}

h1 {
  font-size: 160%;
  color: #003;
}
```

h1, h2, h3 share the same font-family rule

FONTS

Web Safe Browser Fonts examples

Sans Serif

- Verdana
- Arial
- Helvetica
- Tahoma
- Trebuchet Ms

Serif

- Times New Roman
- Georgia
- Palatino
- Cambria

Monospace

- Courier New
- Lucida Console

Cursive

- Comic Sans Ms

```
body {
  font-family: Arial, Helvetica, sans-serif;
}
```

Font Properties - example

```
p {
  font-style: italic;
  font-variant: small-caps;
  font-weight: bold;
  font-size: 1.5rem;
  line-height: 120%;
  font-family: 'Oswald', sans-serif;
}
```

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

font-size units

PX

If you need fine-grained control, renders the letters exactly that number of pixels in height

Use this to avoid dependency on parent components!

EM

1em is equal to the current font-size of the element in question.

By default 1em = 16px. If you were to go and set a font-size of 20px on your body, then 1em = 20px.

%

Just like em's the very nature of percentage sizing is that it is relative. It also cascades in the same way.

If a parent has the font-size of 20px and the child has a font-size of 50%, it will be 10px.

REM

Inherited from the root element (html) and do not cascade.

EM vs REM

```
html {
  font-size: 100%; /* =16px */
}

main {
  font-size: 125%;
}

main p {
  font-size: 1em; /* =20px */
}

div p {
  font-size: 1rem; /* =16px */
}
```

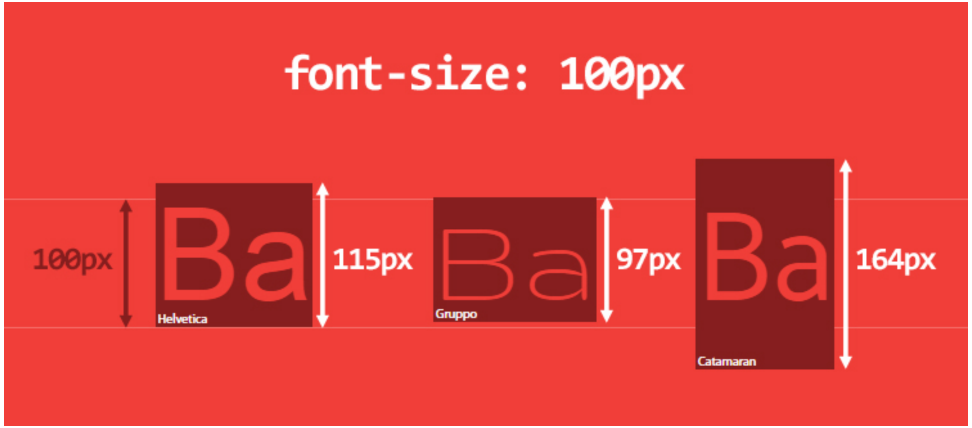
The 'em’ size is relative to its parent.

The 'rem’ size is relative to the root element.

Font-size

```
<p>
    <span class="a">Ba</span>
    <span class="b">Ba</span>
    <span class="c">Ba</span>
</p>


p {
    font-size: 100px
}
.a {
    font-family: Helvetica
}
.b {
    font-family: Gruppo
}
.c {
    font-family: Catamaran
}
```

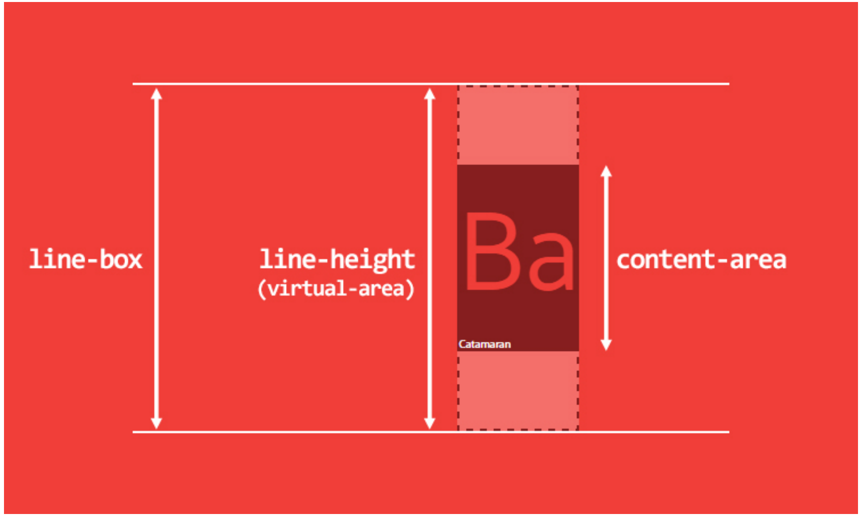
Actual size can be different for each font due to Em-square.



https://iamvdo.me/en/blog/css-font-metrics-line-height-and-vertical-align

Line-height

- The content-area height is defined by the font metrics
- The virtual-area height is the line-height, and it is the height used to compute the line-box's height

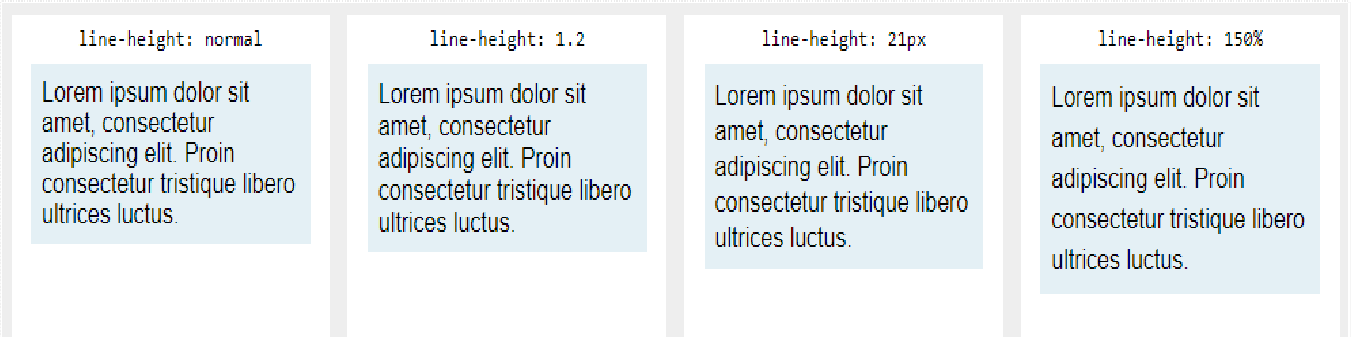


Line-height - examples

```
div {
  line-height: normal;
}
```

```
div {
  line-height: 1.2;
}
```

```
div {
  line-height: 21px;
}
```



line-height: normal

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin consectetur tristique libero ultrices luctus.

line-height: 1.2

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin consectetur tristique libero ultrices luctus.

line-height: 21px

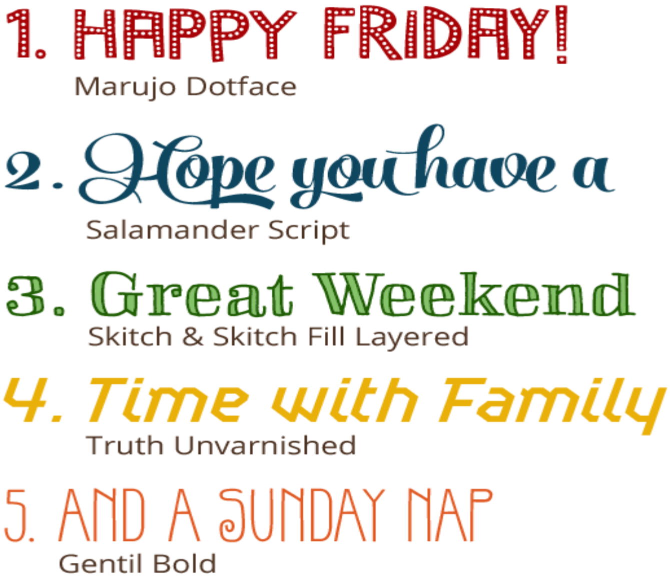
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin consectetur tristique libero ultrices luctus.

line-height: 150%

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin consectetur tristique libero ultrices luctus.

```
div {
  line-height: 150%;
}
```

Web fonts



1. HAPPY FRIDAY!
Marujo Dotface

2. Hope you have a
Salamander Script

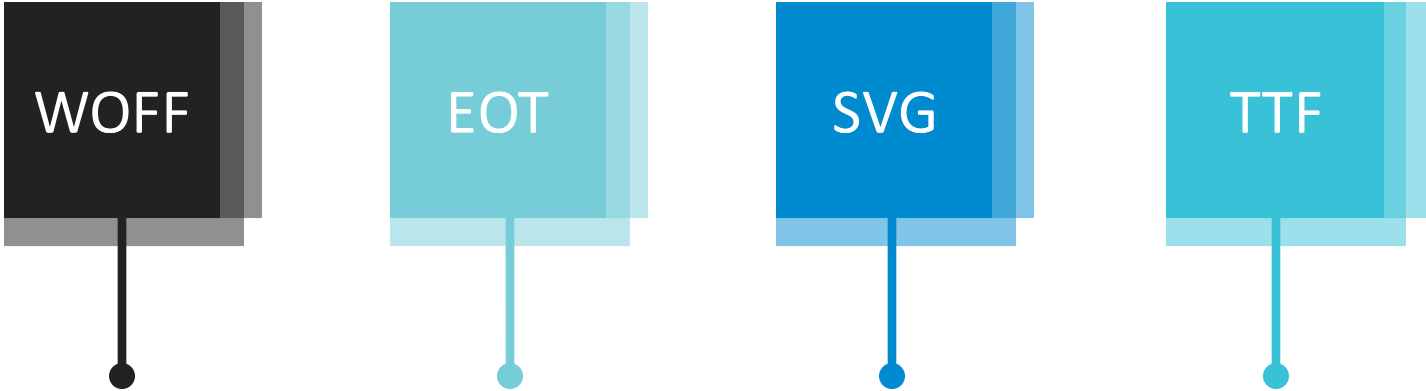
3. Great Weekend
Skitch & Skitch Fill Layered

4. Time with Family
Truth Unvarnished

5. AND A SUNDAY NAP
Gentil Bold

https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Web_fonts

Font formats



WOFF

EOT

SVG

TTF

Web Open Font Format

.woff files are
supported by all
modern browsers

Embedded OpenType

.eot files for older
Internet Explorer
versions (< 8)

Scalable Vector Graphics

.svg files are
supported by all
modern browsers

TrueType Font

.ttf .otf files partial
support in IE

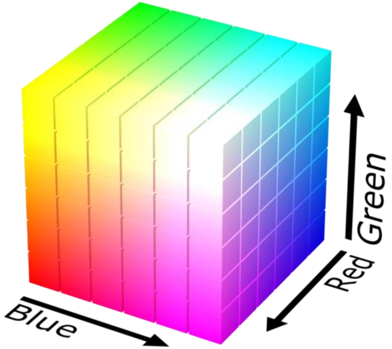
@font-face

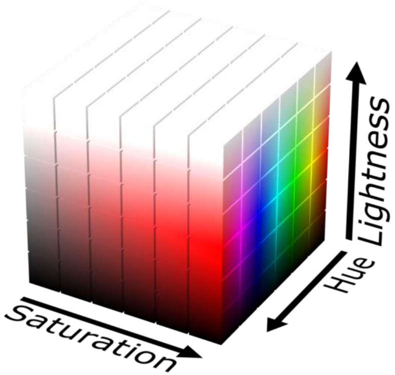
```
@font-face {
  font-family: 'MyWebFont';
  src: url('webfont.eot’);                      /* IE9 Compatibility Modes */
  src: url('webfont.eot?#iefix') format('embedded-opentype'), /*  IE6-8 */
  url('webfont.woff2') format('woff2’),         /* Modern Browsers */
  url('webfont.woff') format('woff’),           /* Modern Browsers + IE11 */
  url('webfont.ttf') format('truetype’),        /* Safari, Android, iOS */
  url('webfont.svg#svgFontName') format('svg'); /* Legacy iOS */
}


body {
  font-family: 'MyWebFont', Fallback, sans-serif;
}
```

https://css-tricks.com/snippets/css/using-font-face/

COLORS

Colors





Name:	Hex:	rgb:	hsl:
purple	#800080	rgb(128, 0, 128)	hsl(300deg 100% 25%)
lime	#00ff00	rgb(0, 255, 0)	hsl(120deg 100% 50%)
aqua	#00ffff	rgb(0 255 255)	hsl(180deg 100% 50%)

https://www.w3.org/wiki/CSS/Properties/color/keywords

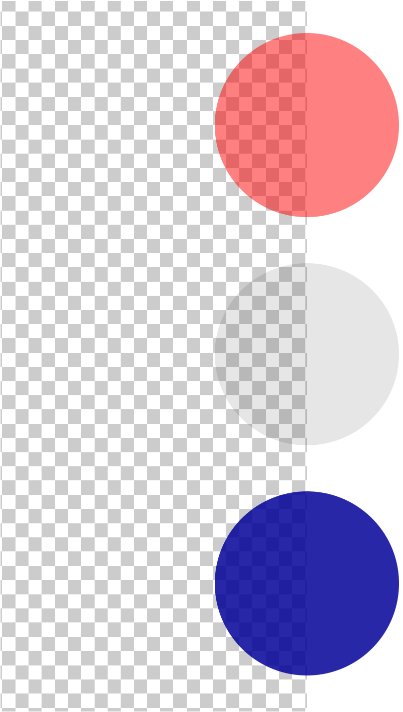
RGB color values

The rgb() function accepts the RGB value in three parameters — providing the red, green and blue hues respectively

```
em {
  color: #f00; /* #rgb */
}

em {
  color: #ff0000; /* #rrggbb */
}

em {
  color: rgb(255, 0, 0);
}

em {
  color: rgb(100%, 0%, 0%);
}
```

```
em {
  color: rgb(255, 0, 0);    /* integer range 0 - 255 */
}

em {
  color: rgb(300, 0, 0);    /* clips to rgb(255,0,0) */
}

em {
  color: rgb(255, -10, 0);  /* clips to rgb(255,0,0) */
}

em {
  color: rgb(110%, 0%, 0%); /* clips to rgb(100%,0%,0%) */
}
```

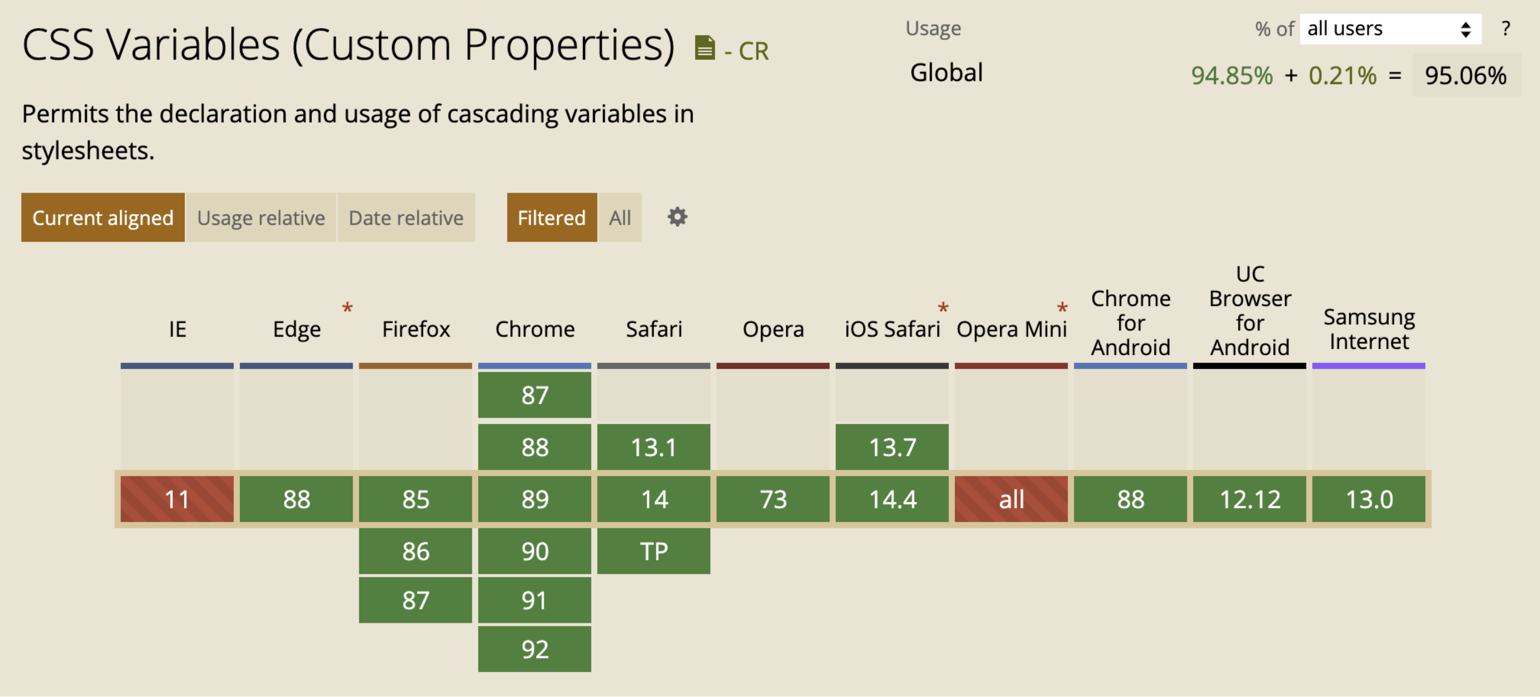
Color opacity



```
.red {
  background-color: #ff000080;
}


.smoke {
  background-color: rgba(0, 0, 0, 0.1);
}


.ink {
  background-color: hsla(240, 100%, 30%, 0.15);
}
```

CSS VARIABLES

Using CSS Custom Properties



CSS Variables (Custom Properties) - CR

Usage: Global

% of all users ?

94.85% + 0.21% = 95.06%

Permits the declaration and usage of cascading variables in stylesheets.

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			87							
			88	13.1		13.7				
11	88	85	89	14	73	14.4	all	88	12.12	13.0
		86	90	TP						
		87	91							
			92							

https://caniuse.com/?search=css%20custom%20properties

CSS Variables

CSS Variables are entities defined by CSS authors which contain specific values to be reused throughout a document

They are set using custom property notation: e.g., --main-color: black, and are accessed using the var() function, e.g., color: var(--main-color)

```css
:root {
  --spacing: 1.5em; /* declaring a variable */
  --base-colors: { /* declaring a mixin */
    color: #fff;
    background-color: color(#fff shade(+80%)); /* modifying a color */
  }
}
.some-class {
  padding: var(--spacing);
  width: calc(100% - var(--spacing)); /* dynamically calculating a value */
  @apply --base-colors;
}
```

Read more in spec   Read more in MDN

Declaring a variable

```
element {
  --main-bg-color: brown;
}


:root {
  --main-bg-color: brown;
}

.two {
  --test: 10px;
}

.three {
  --test: 2em;
}
```

Using the variable

```css
element {
  background-color: var(--main-bg-color);
}

.two {
  color: var(--my-var, red); /* Red if --my-var is not defined */
}

.three {
  background-color: var(--my-var, var(--my-background, pink));
  /* pink if my-var and --my-background are not defined */
}

.three {
  background-color: var(--my-var, --my-background, pink);
  /* Invalid: "--my-background, pink" */
}
```

MISC

Reset | Normalize

The goal of a reset stylesheet is to reduce browser inconsistencies in default line heights, margins and font sizes of headings, and so on

https://meyerweb.com/eric/tools/css/reset/

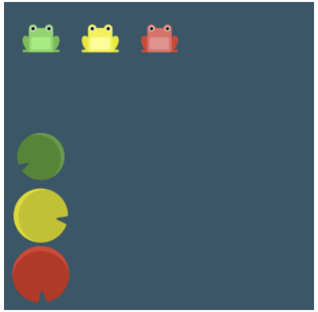
https://github.com/murtaugh/HTML5-Reset/blob/master/assets/css/reset.css

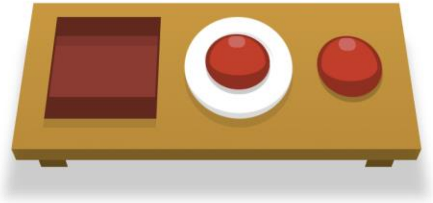
Normalize.css makes browsers render all elements more consistently and in line with modern standards. It precisely targets only the styles that need normalizing.

http://necolas.github.io/normalize.css/

```css
html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
b, u, i, center, dl, dt, dd, ol, ul, li,
fieldset, form, label, legend, table, caption,
tbody, tfoot, thead, tr, th, td, article,
aside, canvas, details, embed, figure,
figcaption, footer, header, hgroup, main, menu,
nav, output, ruby, section, summary, time,
mark, audio, video {
  margin: 0;
  padding: 0;
  border: 0;
  font-size: 100%;
  font: inherit;
  vertical-align: baseline;
}

/* ... and more */
```

Games – practicing…





https://flukeout.github.io/ - …selectors

https://flexboxfroggy.com/ - …flexbox

https://cssgridgarden.com/ - …css grid

https://cssbattle.dev/ - replicate targets with the smallest possible CSS code

Q&A

edu_hu@epam.com