



2019

# AN INTRODUCTION TO FRONTEND

Marco Montalbano

# GOAL

---

<https://git.io/fjmCg>

An introduction to Frontend for ... localhost:5001 Project name

# Hello, world!

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Slides »

### Article Title 1



**tech**

Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

like

### Article Title 2



**tech**

Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

like

### Article Title 3



**tech**

Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

like

This is a cookie policy click [here](#) to close it.

Article Title 4

# HTML



# HTML

# HTML

---

- The acronym “HTML” stands for Hypertext Markup Language
- It is not a programming language
- It is a markup language which describes the structure of a web page semantically
- It describes how to lay out the content of a web page using HTML elements called tags
- It tells the browser what to display.

# TAG

---

- Each tag has a **defined meaning**
- It can contain other tags
- They are enclosed in **angle brackets**: <p>
- Usually tags are made up of two elements: a "start tag" <p> and an "end tag" </p>
- The start tag may include **attributes** within the tag

<SPAN key="value"> testo </SPAN>

void elements (<http://bit.ly/void-elements>)

<INPUT key="value">

# TAGS

---

`<!DOCTYPE html>`

is not an HTML tag; it is an instruction to the web browser about what version of HTML the page is written in

`<!DOCTYPE html>`

`<html>`

`...`

`<title>` embedded in `<head>` tag

page title

`<title>title is visible inside the browser tab</title>`

`<html> - <head> - <body>`

base structure of an HTML document

```
<!DOCTYPE html>
<html>
  <head> ... </head>
  <body> ... </body>
</html>
```

`<meta>` embedded in `<head>` tag

metadata are not displayed on the page but are used by browsers or search engines

```
<meta charset="utf-8">
<meta name="description" content="Page description.">
<meta name="twitter:card" content="summary">
```

# TAGs

---

## <span>

is a generic inline container for phrasing content

this <span>text is on the</span> same line

this text is on the same line

## <p>

paragraph

<p>this is a paragraph</p><p>this is a paragraph</p>

this is a paragraph  
this is a paragraph

## <a>

link

link to <a href="https://www.google.com">google</a>

link to [google](https://www.google.com)

## <div> - <section>

defines a division or a section in an HTML document

### <div>

```
<div>this is a section</div>  
</div>
```

# Hello, world!

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

[Learn more »](#)

## Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

[View details »](#)

## Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

[View details »](#)

## Heading

Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

[View details »](#)

# Hello, world!

Lore ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

[Learn more »](#)

HERO

# CONTENT

# Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

[View details »](#)

# Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

[View details »](#)

## Heading

Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

[View details »](#)

# TAGS

---

<h1> - <h2> - <h...> - <h6>

heading

```
<h1>Title h1</h1>
<h2>Title h2</h2>
<h3>Title h3</h3>
<h4>Title h4</h4>
<h5>Title h5</h5>
<h6>Title h6</h6>
```

Title h1  
Title h2  
Title h3  
Title h4  
Title h5  
Title h6

<ol> - <li>

ordered list

```
<ol>
  <li>first item</li>
  <li>second item</li>
</ol>
```

1. first item
2. second item

<table> - <tr> - <td>

table – row – cell

```
<table>
  <tr>
    <td>cell A1</td>
    <td>cell A2</td>
  </tr>
</table>
```

cell A1	cell A2
---------	---------

<ul> - <li>

unordered list

```
<ul>
  <li>first item</li>
  <li>second item</li>
</ul>
```

- first item
- second item

# TAGS – VOID ELEMENTS

<br>

line break

this text contains<br>a line break

this text contains  
a line break

<input>

text input

<input type="text" name="firstname">

<hr>

defines a thematic break

this is a topic

<hr>

this is a different topic

this is a topic

-----  
this is a different topic



<img>

image



# ATTRIBUTES

---

## <TAG>

all tags can have an **id** and a **class**

```
<div id="my-id" class="my-class my-second-class">text</div>
```

## <input>

- **name** and **value** are submitted with the form data
- **type** specifies the type <input> element to display

```
<input type="text" name="firstname" value="">
```

## <a>

**href** contains a URL or a URL fragment that the hyperlink points to

link to <a href="https://www.google.it">google</a>

## <img>

the attribute **src** is mandatory and defines the image URL

```

```



## HTML 5 NEW TAG

### TAG NOT SUPPORTED IN HTML 5

<!--...-->	Define a comment
<!DOCTYPE>	Defines the document type
<a>	Defines a hyperlink href, hreflang, media, ping, rel, target, type
<abbr>	Defines an abbreviation
<acronym>	Used to define an embedded acronyms
<address>	Defines an address element
<applet>	Used to define an embedded applet
<area>	Defines an area inside an image map alt, coords, href, hreflang, media, ping, rel, shape, target, type
<article>	Defines an article cite, pubdate
<aside>	Defines content aside from the page content
<audio>	Defines sound content autobuffer, autoplay, controls, src
<b>	Defines bold text
<base>	Defines a base URL for all the links in a page href, target
<basefont>	Used to define a default font-color, font-size, or font-family for all the document
<bdo>	Defines the direction of text display dir
<big>	Used to make text bigger
<blockquote>	Defines a long quotation cite
<body>	Defines the body element
 	Inserts a single line break
	Defines a push button autofocus, disabled, form,

<datalist>	Defines a dropdown list
<dd>	Defines a definition description
<del>	Defines deleted text cite, datetime
<details>	Defines details of an element open
<dialog>	Defines a dialog (conversation)
<dfn>	Defines a definition term
<dir>	Used to define a directory list
<div>	Defines a section in a document
<dl>	Defines a definition list
<dt>	Defines a definition term
<em>	Defines emphasized text
<embed>	Defines external interactive content or plugin height, src, type, width
<fieldset>	Defines a fieldset disabled, form, name
<figure>	Defines a group of media content, and their caption
<font>	Used to define font face, font size, and font color of text
<footer>	Defines a footer for a section or page Defines a form
<form>	accept-charset, action, autocomplete, enctype, method, name, novalidate, target
<frame>	Used to define one particular window (frame) within a frameset
<frameset>	Used to define a frameset, which organized multiple windows (frames)
<h1> to <h6>	Defines header 1 to header 6
<head>	Defines information about the document
<header>	Defines a header for a section or page
<hgroup>	Defines information about a section in a document
<ins>	Defines inserted text cite, datetime
<keygen>	Defines a generated key in a form autofocus, challenge, disabled, form, keytype, name
<kbd>	Defines keyboard text
<label>	Defines an inline sub window for, form
<legend>	Defines a title in a fieldset
<li>	Defines a list item value
<link>	Defines a resource reference href, hreflang, media, rel, sizes, type
<map>	Defines an image map name
<mark>	Defines marked text
<menu>	Defines a menu list label, type
<meta>	Defines meta information charset, content, http-equiv, name
<meter>	Defines measurement within a predefined range high, low, max, min, optimum, value
<nav>	Defines navigation links
<noframes>	Used to display text for browsers that do not handle frames
<noscript>	Defines a noscript section
<object>	Defines an embedded object data, form, height, name, type, usemap, width
<ol>	Defines an ordered list reversed, start
<optgroup>	Defines an option group label, disabled
<option>	Defines an option in a drop-down list disabled, label, selected, value
<samp>	Defines sample computer code
<script>	Defines a definition list async, type charset defer, src
<section>	Defines a section cite
<select>	Defines a selectable list autofocus, disabled, form, multiple, name, size
<small>	Defines small text
<source>	Defines media resources media, src, type
<span>	Defines a section in a document
<strong>	Defines strong text
<style>	Defines a style definition type, media, scoped
<sub>, <sup>	Defines sub/super-scripted text
<table>	Defines a table summary
<tbody>	Defines a table body summary
<td>	Defines a table cell colspan, headers, rowspan
<textarea>	Defines a text area autofocus, cols, disabled, form, maxlength, name, placeholder, readonly, required, rows, wrap
<tfoot>, <thead>	Defines a table footer / head
<th>	Defines a table header colspan, headers, rowspan, scope
<time>	Defines a date/tim datetime
<title>	Defines the document title
<tr>	Defines a table row datetime



## REFERENCES

- W3C – <https://www.w3.org/html>
- Codecademy – <https://www.codecademy.com/learn/learn-html>
- MDN web docs – <https://developer.mozilla.org/en/docs/Web/HTML>
- w3schools – <http://www.w3schools.com/TAGs>

# HTML



HTML  
in action



<http://bit.ly/fe4b-html>



CSS

# CSS

---

- The acronym “CSS” stands for **Cascade Style Sheet**
- **It is not** a programming language
- Used for describing the **presentation** of an HTML document
- Defines fonts, layout, colors, etc. of the whole page and single elements
- It tells the browser **how** to display informations.

# CSS

---

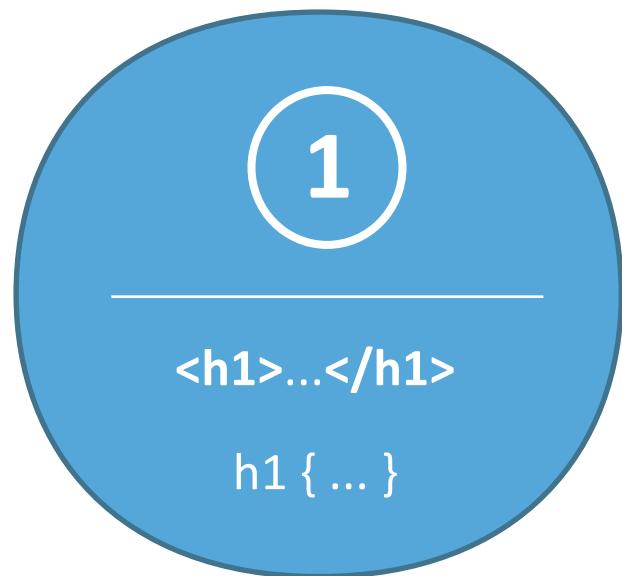
A web page is styled according to  
a set of **style rules**

```
selector → div {  
property →   color: blue;    value  
              font-size: 15px;  
            }  
          }
```

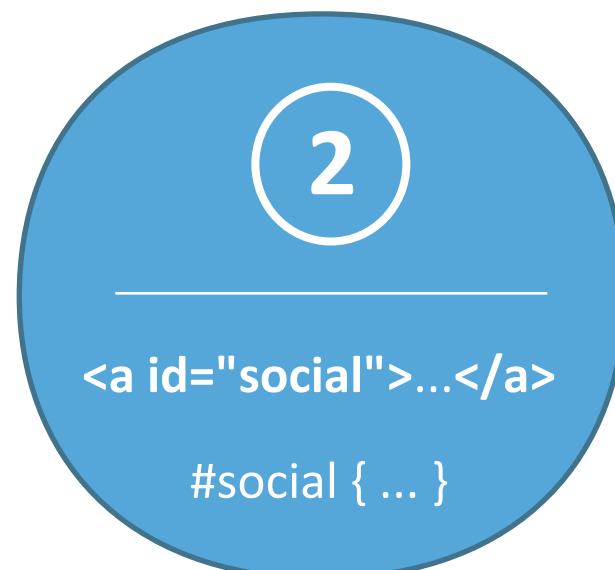
# SELECTORS

---

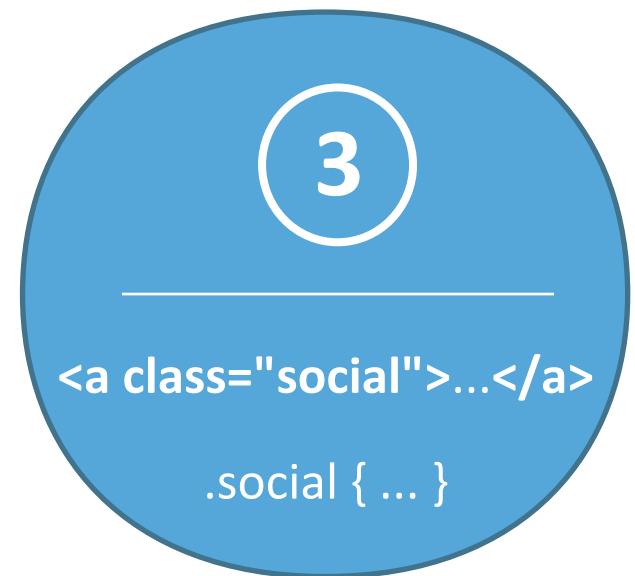
in order to identify which elements are affected by style rules we need a **selector**



tag name



element ID  
(unique)



element class

# SELECTORS PRO

---

## further examples of how to combine selectors

`div` all `<div>`

`div, span` `<div>` and `<span>`

`div span` `<span>` nested in `<div>`

`div > span` `<span>` children of `<div>`

`.elm-class` elements with class "elm-class"

`div.elm-class` all `<div>` with class "elm-class"

`#elm-id` element with id "elm-id"

`div#elm-id` `<div>` with id "elm-id"

`#elm-id.elm-class` element with id "elm-id" and class "elm-class"

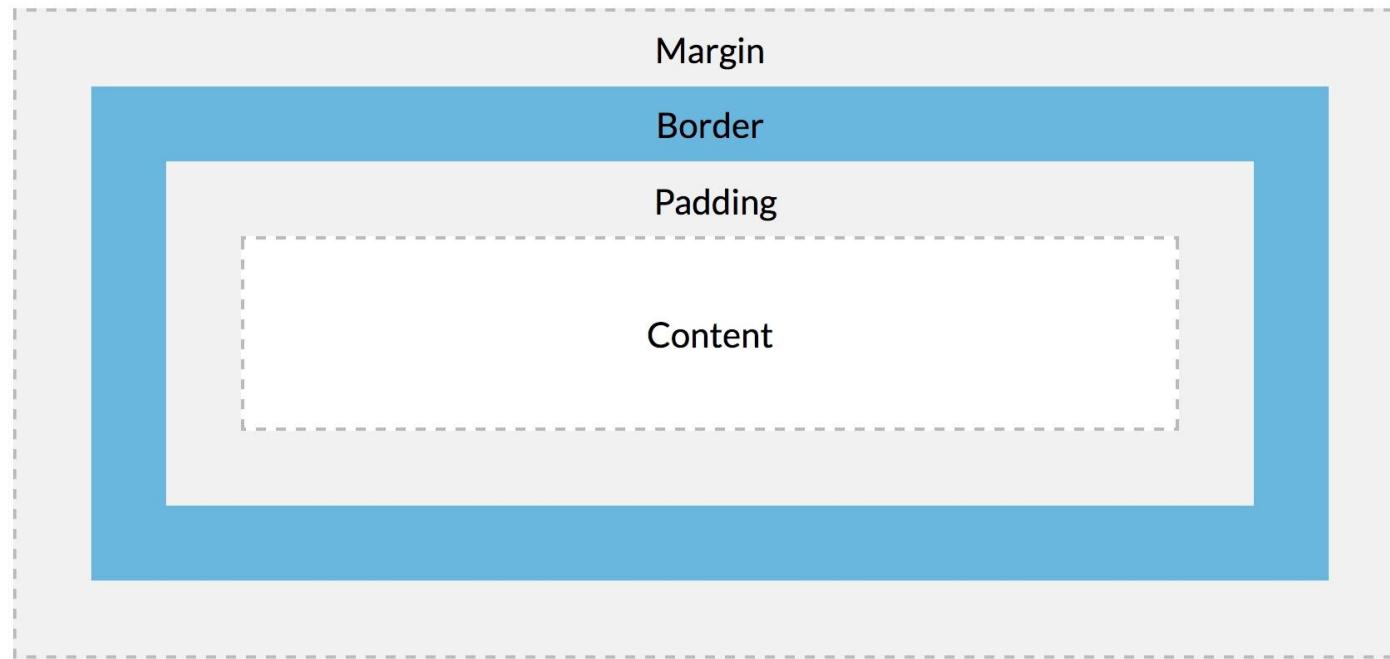
`:hover` element during mouse-over

`:focus` element during focus

# BOX MODEL

---

the CSS box model is essentially a box that wraps around every HTML element



# PROPERTIES and VALUES

---

height: 150px;  
width: 50%;

sets a width and a height to the element

border: 1px solid #F1F1F1;  
border-top: 2px solid #F1F1F1;

adds a border around the element

padding: 10px;  
padding: 10px 15px;  
padding: 10px 15px 10px 10px;  
padding-top: 10px; padding-right: 15px; ...

increases space around the border and content of the element

margin: 10px;  
margin: 10px 15px;  
margin: 10px 15px 10px 10px;

increases the element margin

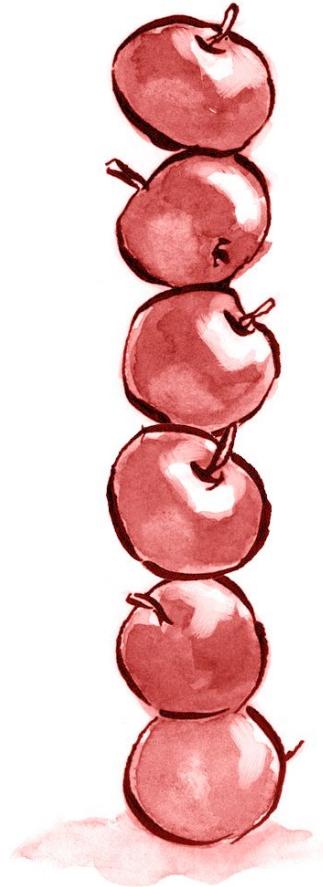
# PROPERTIES PRO

---

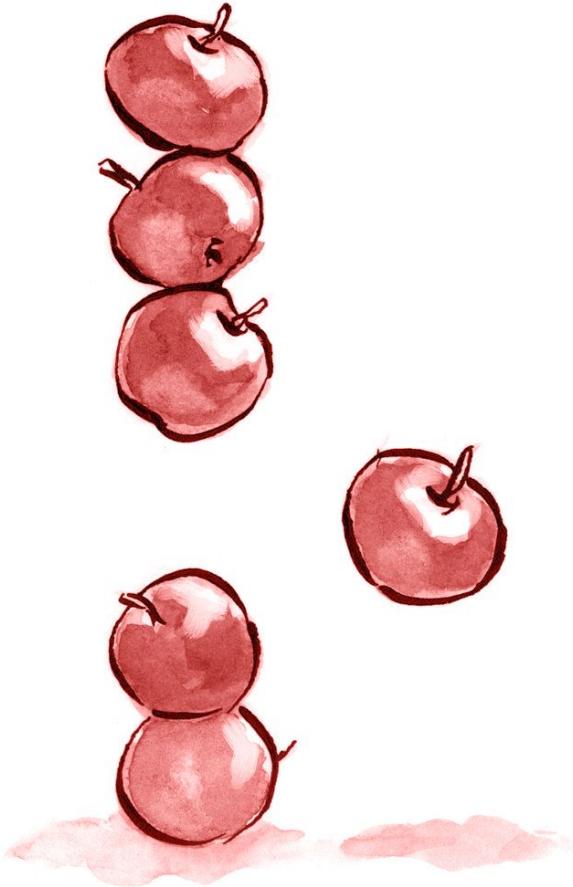
color	#FF0000	text-align	left, right, center
background-color	blue	text-shadow	1px 1px 1px blue
background-image	url("html5.png")	text-decoration	<del>line-through, underline</del>
font-family	Arial, sans-serif	position	static, relative, absolute
font-size	14px	top	
font-weight	<b>bold</b>	right	
font-style	<i>italic</i>	bottom	
		left	

# POSITIONS

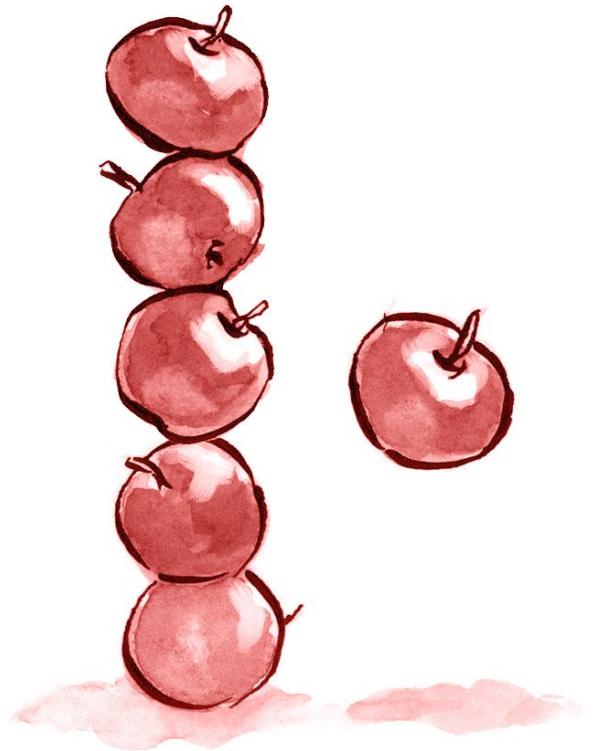
---



Static



Relative



Absolute

# DISPLAY

---

every HTML element has a default **display** value depending on what type of element it is

**display: block;**

<div> and <p> are block elements.

A block-level element always starts on a new line and takes up the full width available.



<p>This is a paragraph</p>

<p>This is another paragraph</p>

**display: inline;**

<a> does not start on a new line and only takes up as much width as necessary



This <a>link</a> does not start on a new line and only takes up as much width as necessary.

CSS Selectors	
*	All elements
div	<div>
div *	All elements within <div>
div span	<span> within <div>
div, span	<div> and <span>
div > span	<span> with parent <div>
div + span	<span> preceded by <div>
.class	Elements of class "class"
div.class	<div> of class "class"
#itemid	Element with id "itemid"
div#itemid	<div> with id "itemid"
a[attr]	<a> with attribute "attr"
a[attr='x']	<a> when "attr" is "x"
a[class~='x']	<a> when class is a list containing 'x'
a[lang='en']	<a> when lang begins "en"
CSS Colours	
#789abc	RGB Hex Notation
#acf	Equates to "#aaccff"
rgb(0,-25,50)	Value of each of red, green, and blue. 0 to 255, may be swapped for percentages.
CSS Pseudo Selectors and Pseudo Classes	
:first-child	First child element
:first-line	First line of element
:first-letter	First letter of element
:hover	Element with mouse over
:active	Active element
:focus	Element with focus
:link	Unvisited links
:visited	Visited links
:lang(var)	Element with language "var"
:before	Before element
:after	After element

CSS Sizes	
0	0 requires no unit
Relative Sizes	
em	1em equal to font size of parent (same as 100%)
ex	Height of lower case "x"
%	Percentage
Absolute Sizes	
px	Pixels
cm	Centimeters
mm	Millimeters
in	Inches
pt	1pt = 1/72in
pc	1pc = 12pt
CSS Positioning	
display	clear
position	z-index
top	direction
right	unicode-bidi
bottom	overflow
left	clip
float	visibility
CSS Dimensions	
width	min-height
min-width	max-height
max-width	vertical-align
height	
CSS Colour and Background	
color	background-repeat
background	background-image
background-color	background-position
background-attachment	

CSS Box Model	
CSS Boxes	
margin	border-color
margin-top	border-top-color
margin-right	border-right-color
margin-bottom	border-bottom-color
margin-left	border-left-color
padding	border-style
padding-top	border-top-style
padding-right	border-right-style
padding-bottom	border-bottom-style
padding-left	border-left-style
border	border-width
border-top	border-top-width
border-bottom	border-right-width
border-right	border-bottom-width
border-left	border-left-width
CSS Text	
text-indent	word-spacing
text-align	text-transform
text-decoration	white-space
text-shadow	line-height
letter-spacing	

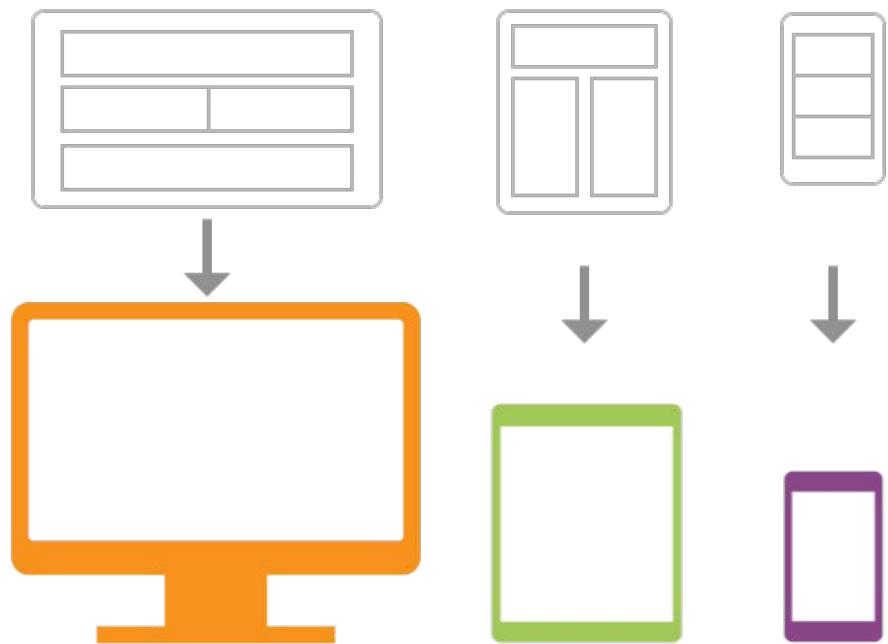
CSS Paging	
size	page-break-inside
marks	page
page-break-before	orphans
page-break-after	widows
CSS Interface	
cursor	outline-style
outline	outline-color
outline-width	
CSS Aural	
volume	elevation
speak	speech-rate
pause	voice-family
pause-before	pitch
pause-after	pitch-range
cue	stress
cue-before	richness
cue-after	speak-punctuation
play-during	speak-numeral
azimuth	
CSS Miscellaneous	
content	list-style-type
quotes	list-style-image
counter-reset	list-style-position
counter-increment	marker-offset
list-style	
CSS Fonts	
font	font-weight
font-family	font-stretch
font-style	font-size
font-variant	font-size-adjust

# AWD vs RWD

---

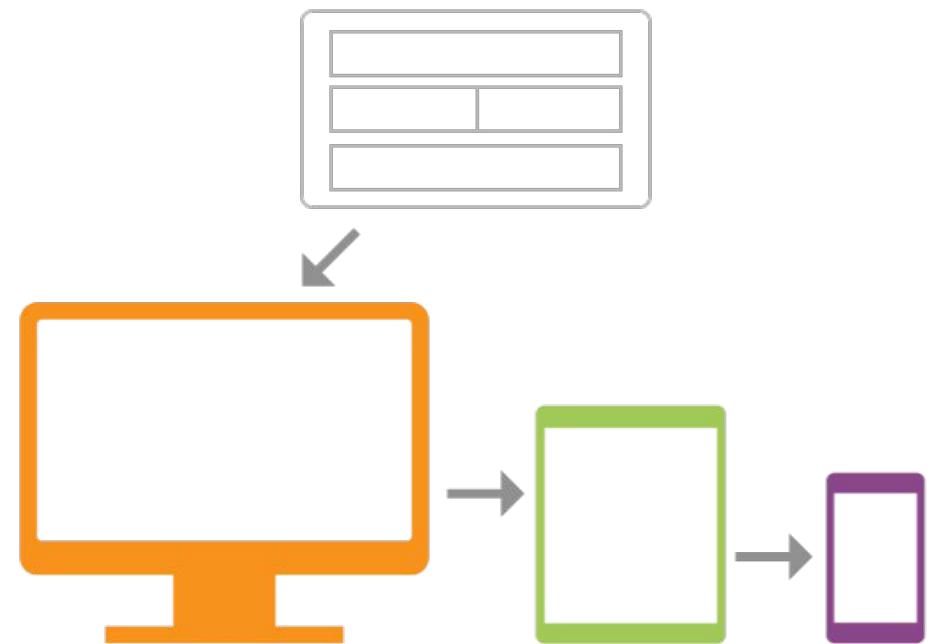
## ADAPTIVE WEB DESIGN

multiple templates  
specific for each device



## RESPONSIVE WEB DESIGN

a single design that  
adjusts to each device



# RESPONSIVE WEB DESIGN

---

- Rule 1 – <meta name="viewport" content="width=device-width, initial-scale=1.0">
- Rule 2 – use % values instead of px
- Rule 3 – define breakpoints using media query

Media Query uses `@media` rule to include a set of CSS properties  
only if the specific condition is true

```
.col {  
    width: 100%;  
}  
  
@media (min-width: 1200px) {  
    .col {  
        width: 50%;  
    }  
}
```

# LAYOUT EVOLUTION

<http://bit.ly/css-layout-evolution>

At the beginning the most sophisticated layout tool designers had was

<table> ... </table>



Then we began to hack out layouts with complicated mixtures of absolute positioning, floats ..

position: absolute;  
float: left;

.. and inline-block

display: inline-block;

zoom: 1;

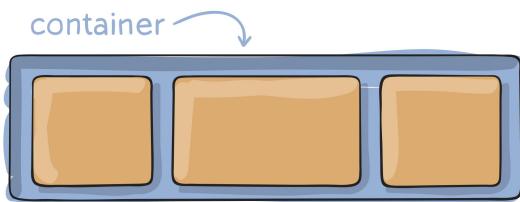
\*display: inline;

flexbox

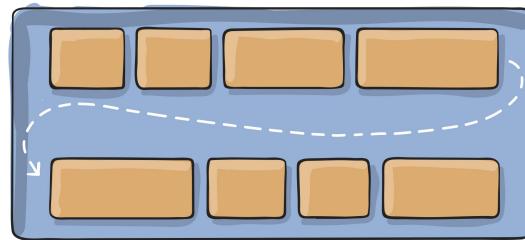
# FLEXBOX

<https://mzl.la/2Ckg35E> - <https://css-tricks.com/snippets/css/a-guide-to-flexbox>

it gives the container the ability to alter its items' width/height (and order) to best fill the available space

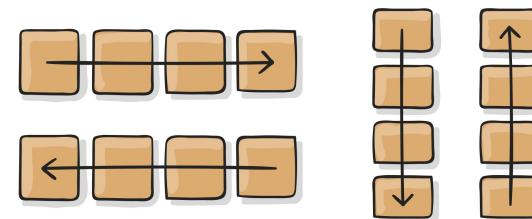


```
.container {  
    display: flex;  
}
```



```
.container {  
    flex-wrap: nowrap | wrap | wrap-reverse;  
}
```

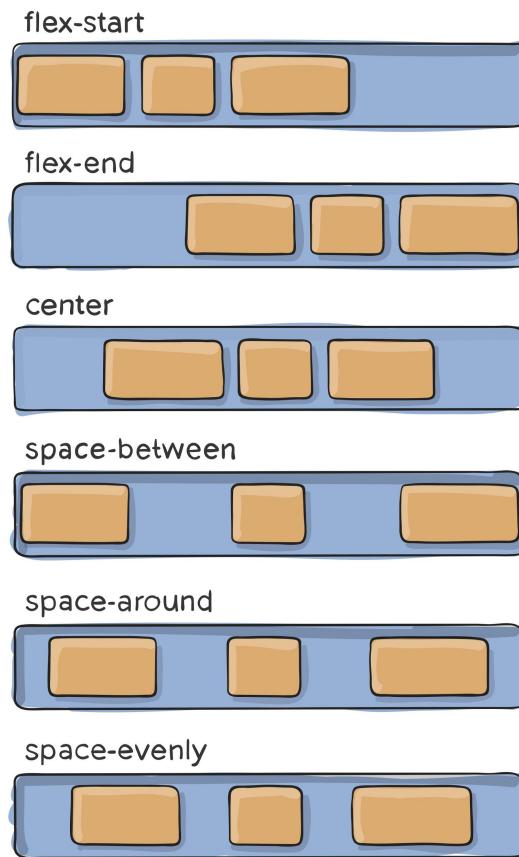
```
.container {  
    flex-direction: row | row-reverse | column | column-reverse;  
}
```



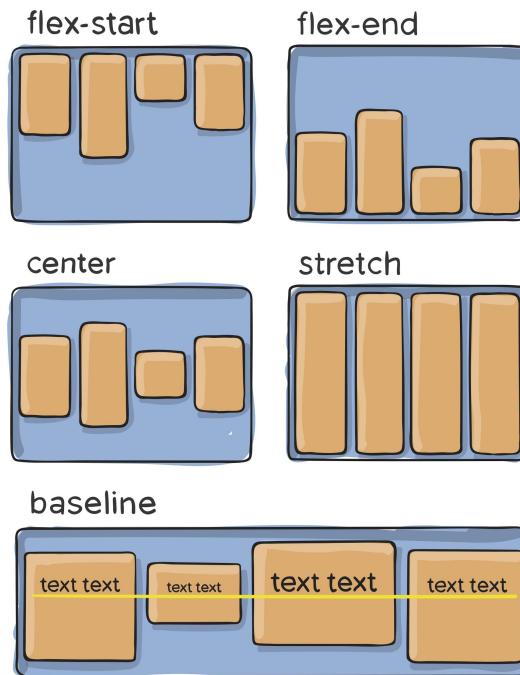
# FLEXBOX

<https://mzl.la/2Ckg35E> - <https://css-tricks.com/snippets/css/a-guide-to-flexbox>

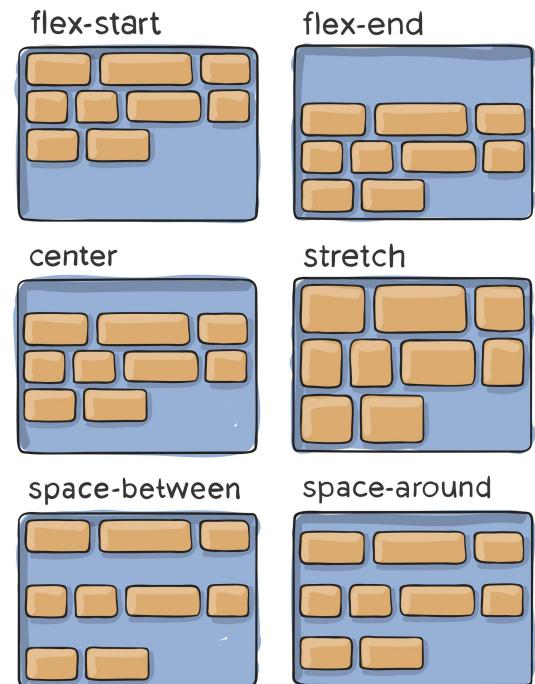
justify-content



align-items

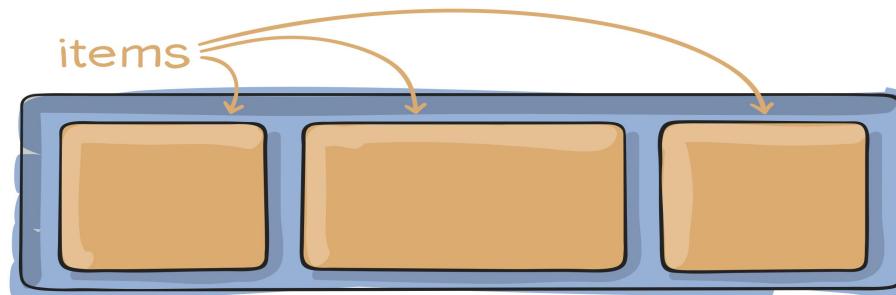


align-content

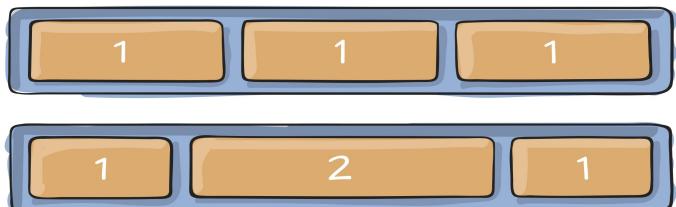


# FLEXBOX

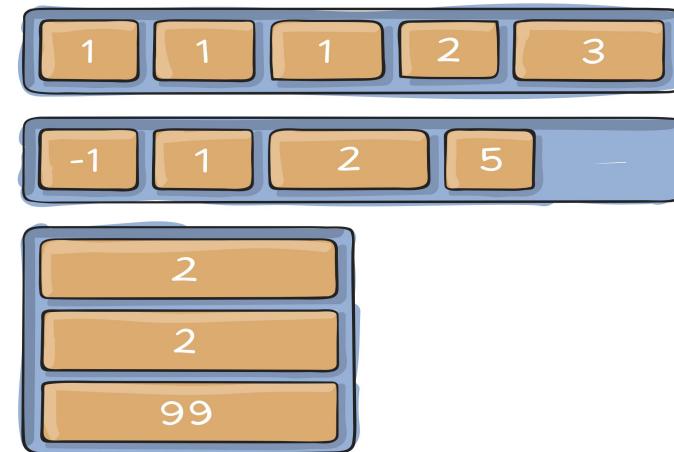
<https://mzl.la/2Ckg35E> - <https://css-tricks.com/snippets/css/a-guide-to-flexbox>



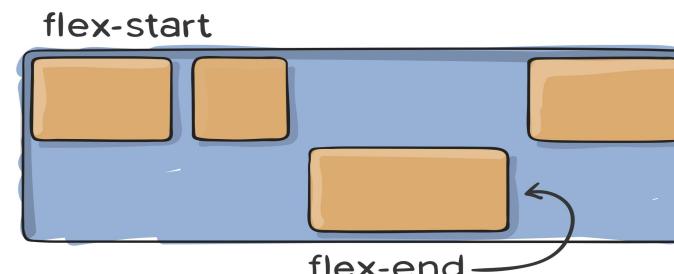
flex-grow



order



align-self



# FLEXBOX READ PLAY MORE

---

**FLEXBOX FROGGY**

Benvienuto su Flexbox Froggy, un gioco in cui devi aiutare Froggy e i suoi amici scrivendo codice CSS! Guida la rana verso la foglia di ninfea sulla destra usando la proprietà `justify-content`, la quale allinea gli elementi orizzontalmente e accetta i seguenti valori:

- `flex-start`: Allinea gli elementi alla sinistra del contenitore.
- `flex-end`: Allinea gli elementi alla destra del contenitore.
- `center`: Allinea gli elementi al centro del contenitore.
- `space-between`: Separa gli elementi con uguale spazio tra di loro.
- `space-around`: Separa gli elementi con uguale spazio attorno ad essi.

Per esempio, `justify-content: flex-end;` muoverà la rana sulla destra.

```
1 #pond {  
2   display: flex;  
3   |       |  
4 }  
5  
6  
7  
8  
9  
10
```

Prossimo

[SPONSOR] Hotjar – See how your visitors are really using your website.

Flexbox Froggy è stato creato da [Codepig](#) • [GitHub](#) • [Twitter](#) • [Impostazioni](#)

Vuoi imparare CSS grid? Gioca a [Grid Garden](#).

<http://flexboxfroggy.com/>

## WHY “CASCADING” ?

---

the rule used is chosen by **cascading down** from the more general rules to the specific rule required

selector	# of ID	# of CLASS	# of TAG	specificity
#my-id	1	0	0	100
.my-class	0	1	0	10
li	0	0	1	1
ul li	0	0	2	2
li.my-class	0	1	1	11

```
<ul>
  <li id="my-id" class="my-class"></li>
</ul>
```

the most specific rule is chosen •  
with equal specificity, the last rule is chosen •  
**style-inline** rules out every other rule •

# WHERE DO I WRITE THIS?

Style inline



don't try this at the office

this `<span style="color: blue;">text is blue</span>`

this text is blue

`<style>`



not so bad, but avoid it anyway

`<style>`

`span { color: blue; }`

`</style>`

`<link>`

load it from file

*new tag is embedded in `<head>` tag*

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

style.css

```
span {  
  color: blue;  
}
```



## REFERENCES

- W3C – <https://www.w3.org/Style/CSS/Overview.en.html>
- Codecademy – <https://www.codecademy.com/learn/learn-css>
- MDN web docs – <https://developer.mozilla.org/en/docs/Web/CSS>
- w3schools – <https://www.w3schools.com/css>

CSS



CSS  
in action



<http://bit.ly/fe4b-css>

# JS

## JAVASCRIPT

# JAVASCRIPT

---

- It's not an acronym
- It is a **scripting Object-Oriented** language
- Everything is an **object** (except **null** and **undefined**)
- It is **untyped**
- It is implemented **client-side** in **web browsers**
- It is also implemented **server-side** with **node.js**
- It makes web pages **interactive**
- It's an implementation of the **EcmaScript** standard

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/About\\_JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript)

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/A\\_re-introduction\\_to\\_JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript/A_re-introduction_to_JavaScript)

# ECMAScript

ES 1

First edition.

ES 2

Editorial changes to keep the specification fully aligned with ISO/IEC 16262 international standard.

ES 3

Added regular expressions, better string handling, new control statements, try/catch exception handling, tighter definition of errors, formatting for numeric output ...

ES 5

Adds "strict mode," a subset intended to provide more thorough error checking and avoid error-prone constructs. Adds some new features, such as getters and setters, library support for JSON, and more complete reflection on object properties.

ES 5.1

This edition 5.1 of the ECMAScript standard is fully aligned with third edition of the international standard ISO/IEC 16262:2011.

ES 6

This update adds significant new syntax for writing complex applications, including class declarations and ES6 modules. Other new features include iterators and for/of loops, Python-style generators, arrow function expression, binary data, typed arrays, new collections, promises, number and math enhancements, reflection, proxies and template literals for strings.

ES 7

The major standard language features include block-scoping of variables and functions, destructuring patterns (of variables), ...

ES 8

Includes async/await constructions, which works using generators and promises.

## timeline



# VARIABLES

```
var x = 5;
```

Variable names:

- Can contain **letters, numbers, \$ and \_**
- Can start with a **letter, \$ and \_**
- Are **case-sensitive** ( y and Y are two different variables )

```
var myUndefined;
```

```
var myNumber = 3 + 5;
```

```
var myFloat = 3.5;
```

```
var myObject = {  
    name: 'John',  
    age: 28  
};  
  
console.log(myObject.name);  
  
//Prints 'John'
```

```
var myString = 'Hello' + ' ' + 'World!!!';
```

```
var myArray = ['apple', 'lime', 3];  
  
console.log( myArray[0] );  
  
//Prints 'apple'
```

# COMPARISON OPERATORS

<https://mzl.la/2wCuFwX>

## `==` VS `===`

A **comparison operator** compares its operands and returns a logical value based on whether the comparison is true

3 == 3	true
--------	------

"3" == 3	true
----------	------

3 < 4	true
-------	------

3 === 3	true
---------	------

"3" === 3	false
-----------	-------

3 >= 3	true
--------	------

# LOOPS AND ITERATION

<https://mzl.la/2KX0uUo>

**Loops** offer a quick and easy way to do something repeatedly

```
for ( var step = 0; step < 5; step++ ) {  
    // Runs 5 times, with values of step 0 through 4.  
    console.log('Walking east one step');  
}
```

**for**

**do**

```
var i = 0;  
do {  
    i += 1;  
    console.log(i);  
} while (i < 5);
```

**while**

```
var i = 0;  
while (i < 5) {  
    i += 1;  
    console.log(i);  
};
```

# FUNCTIONS <https://mzl.la/2IEQd0C>

---

**function add() { ... };**

- Defined through keyword **function** followed by a **name** and **( )**
- The name can contain **letters**, **numbers**, **\_** and **\$** (just like variables)
- The **( )** can contain **arguments** separated by commas

```
function add(num1, num2) {  
    return num1 + num2;  
}
```

```
var add = function (num1, num2) {  
    return arguments[0] + arguments[1];  
}
```

```
var result = add(5, 5); //= 10
```

# SCOPE

---

Variables declared (using var keyword) are  
**function-level scoped**

```
// here code can't see myVariable
function myFunction() {
    var myVariable = 5;
    // here code sees myVariable
}
```

Variables declared outside a function are  
**global-level scope**

```
var myVariable = 5;
// here code sees myVariable
function myFunction() {
    // here code sees myVariable
}
```

```
// here code sees myVariable
function myFunction() {
    myVariable = 5;
}
```



# HOISTING

---

```
function getValue(condition) {  
    // myVariable exists here with a value of undefined  
    if (condition) {  
        var myVariable = 5;  
    } else {  
        // myVariable exists here with a value of undefined  
    }  
    // myVariable exists here with a value of undefined  
}
```

# HOISTING

---

JavaScript behind the scene

```
function getValue(condition) {  
  
    if (condition) {  
  
        var myVariable = 5;  
  
    } else {  
  
    }  
  
}
```

```
function getValue(condition) {  
  
    var value;  
  
    if (condition) {  
  
        myVariable = 5;  
  
    } else {  
  
    }  
  
}
```



# CODE

```
function carouselInterchange() {  
  var innerWidth = window.innerWidth,  
    imgSlide = $('.item-image-carousel');  
  
  imgSlide.each(function(index) {  
    var self = $(this);  
  
    self.attr('src', (  
      (innerWidth < 768) ? self.data('src-small') :  
      (innerWidth >= 768 && innerWidth < 992) ? self.data('src-medium') : self.data('src-big')  
    ));  
  });  
  
$(window).ready(function() {  
  carouselInterchange();  
});  
  
$(window).resize(function() {  
  carouselInterchange();  
});  
function hide(markers) {  
  setMapOnAll(null, markers);  
}  
  
function show(map, markers) {  
  setMapOnAll(map, markers);  
}  
  
function setMapOnAll(map, markers) {  
  for (var i = 0; i < markers.length; i++) {  
    markers[i].setMap(map);  
  }  
}  
  
function showLayerKML() {  
  layerKML.setMap(map);  
}  
  
$(".cbp-hrsub-inner").mouseover(function(){  
  $(".carousel-indicators").removeClass('nascondi');  
  $(".carousel-indicators").addClass('nascondi');  
});  
$(".cbp-hrsub-inner").mouseout(function(){  
  $(".carousel-indicators").removeClass('nascondi');  
});  
  
$("#navbar li").mouseover(function(){  
  $(".carousel-indicators").removeClass('nascondi');  
  $(".carousel-indicators").addClass('nascondi');  
});  
$("#navbar li").mouseout(function(){  
  $(".carousel-indicators").removeClass('nascondi');  
});  
  
$(".navbar-brand").mouseover(function(){  
  $(".carousel-indicators").removeClass('nascondi');  
});  
$(".navbar-brand").mouseout(function(){  
  $(".carousel-indicators").removeClass('nascondi');  
});  
$(".box-submenu2").mouseover(function(){  
  $(".cbp-hrmenu .cbp-hrsub-inner .col-dx-menu").removeClass('nascondi');  
  $(".cbp-hrmenu .cbp-hrsub-inner .col-dx-menu").addClass('nascondi');  
});  
$(".box-submenu2").mouseout(function(){  
  $(".cbp-hrmenu .cbp-hrsub-inner .col-dx-menu").removeClass('nascondi');  
});  
$(".icon-list-acqua, .icon-list-energia, .icon-list-pubblica").mouseover(function(){  
  $(this).find(".no").removeClass('hidden');  
  $(this).find(".yes").addClass('hidden');  
});  
$(".icon-list-acqua, .icon-list-energia, .icon-list-pubblica").mouseout(function(){  
  $(this).find(".no").addClass('hidden');  
  $(this).find(".yes").removeClass('hidden');  
});  
$(".icon-list-energia").mouseover(function(){  
  $(".no").removeClass('hidden');  
});
```

# MODULE PATTERN

```
var myModule = (function () {  
}());
```

1

```
var myModule = (function () {  
  
    var module = {};  
  
    return module;  
  
}());
```

2

```
var myModule = (function () {  
  
    var module = {},  
        privateVariable = 'Hello World'  
;  
  
    var privateMethod = function () {  
        // ...  
    };  
  
    module.publicProperty = 'Foobar';  
  
    module.publicMethod = function () {  
        alert(privateVariable);  
    };  
  
    return module;  
  
}());
```



## REFERENCES

- MDN web docs – <https://developer.mozilla.org/it/docs/Web/JavaScript>
- Codecademy – <https://www.codecademy.com/learn/introduction-to-javascript>
- JavaScript Garden – <http://bonsaiden.github.io/JavaScript-Garden>
- w3schools – <https://www.w3schools.com/js>

# JS

## JAVASCRIPT & DOM

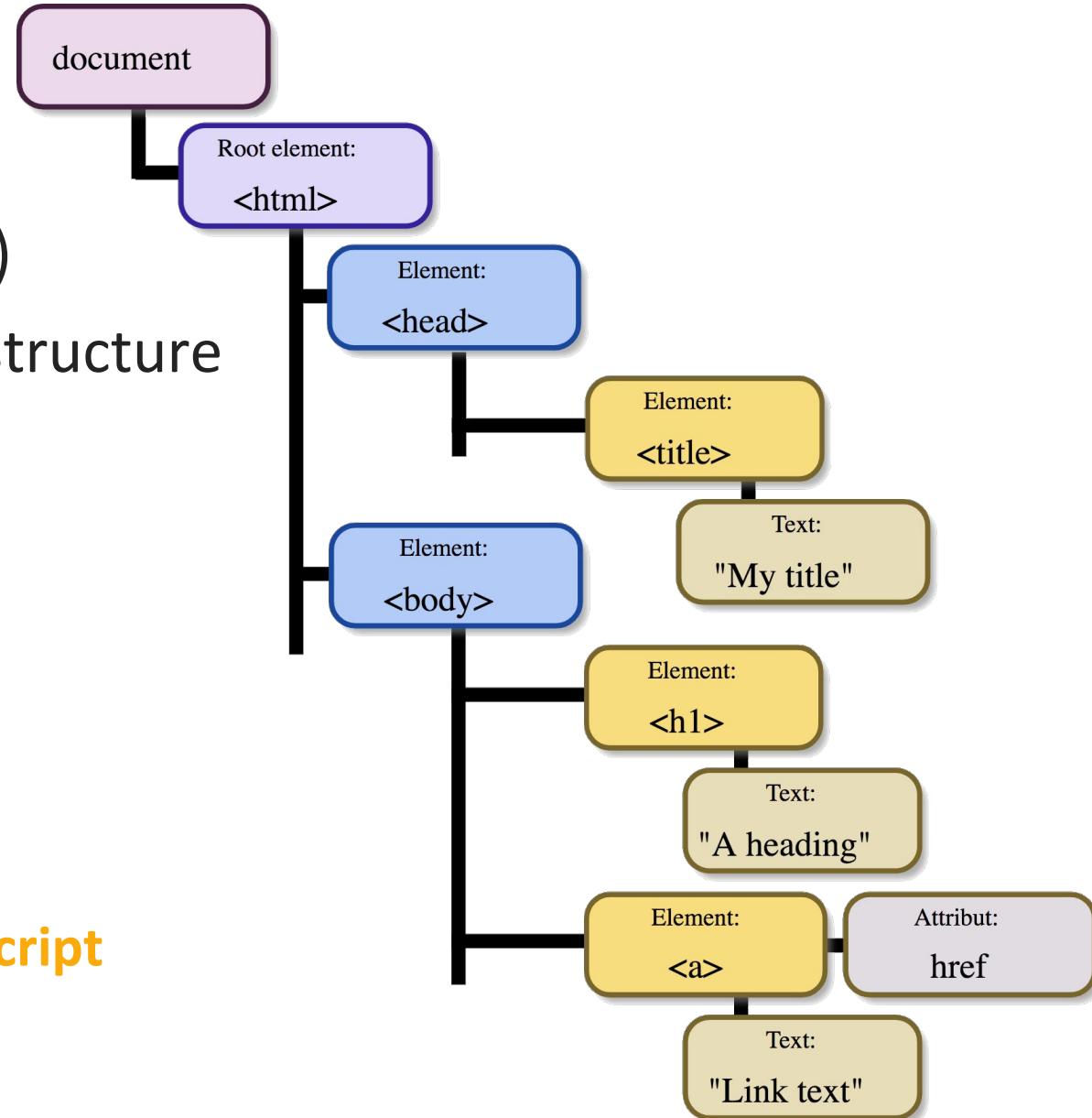
# DOM

---

## DOM ( Document Object Model )

describes the HTML document as a tree structure  
wherein each **node** is an **object**

- The **document** object represents your web page
- Document is made up of **nodes**
- **Element** nodes are html tags
- The HTML **DOM** can be accessed with **JavaScript**



# DOM • JS FUNCTIONS

## `document.getElementById( 'my-id' )`

returns the element that has the ID attribute with the specified value

```
<span id="my-id">test</span>
```

```
<script>
  var element = document.getElementById("my-id");
</script>
```

## `document.querySelectorAll( '.my-class' )`

returns a static NodeList representing a list of the document's elements that match the selector.

```
<span class="my-class my-class-1">test</span>
<span class="my-class my-class-2">test</span>
```

```
<script>
  var element = document.querySelectorAll(".my-class");
  // elements contains both spans.
</script>
```

## `document.getElementsByTagName( 'span' )`

returns a collection of all elements in the document with the specified tag name, as a NodeList object

## `document.getElementsByClassName( 'my-class' )`

returns a collection of all elements in the document with the specified class name, as a NodeList object

## `document.getElementsByName( 'email' )`

returns a collection of all elements in the document with the specified name (the value of the name attribute), as a NodeList object

## `document.querySelector( '.my-class' )`

returns the first Element within the document that matches the specified selector, or group of selectors.

```
<span class="my-class my-class-1">test</span>
<span class="my-class my-class-2">test</span>
```

```
<script>
  var element = document.querySelector(".my-class");
  // element contains the first span.
</script>
```

# DOM • NODE PROPERTIES

## childNodes → NodeList object

returns a collection of a node's child nodes

```
<div id="my-id"><span>Hello</span> <b>World</b>!</div>
```

```
<script>
  var element = document.getElementById("my-id");
  console.log( element.childNodes ); //= [ <span>, <b> ]
</script>
```

## textContent

returns the textual content of the specified node, and all its descendants

```
<div id="my-id">Hello <b>World</b>!</div>
```

```
<script>
  var element = document.getElementById("my-id");
  console.log( element.textContent ); //= Hello World!
</script>
```

## firstChild - lastChild → Node object

returns the first and last child node of the specified node

```
<div id="my-id"><span>Hello</span> <b>World</b>!</div>
```

```
<script>
  var element = document.getElementById("my-id");
  console.log( element.firstChild ); //= <span>
  console.log( element.lastChild ); //= <b>
</script>
```

## innerHTML

returns the HTML content of an element

```
<div id="my-id">Hello <b>World</b>!</div>
```

```
<script>
  var element = document.getElementById("my-id");
  console.log( element.innerHTML ); //= Hello <b>World</b>!
</script>
```

# DOM • AJAX

## COMMON EVENTS

<https://mzl.la/2G9oTm1>

click - change  
focus - blur  
submit

```
function doAjax() {  
    var xhr = new XMLHttpRequest();  
    xhr.open('GET', 'ajax-data.json');  
    xhr.send(null);  
  
    xhr.onreadystatechange = function () {  
        if (xhr.readyState === 4) {  
            if (xhr.status === 200) {  
                alert(xhr.responseText);  
            } else {  
                alert('Error: ' + xhr.status);  
            }  
        }  
    }  
};  
  
var ajaxElement = document.getElementById('ajax');  
ajaxElement.addEventListener('click', doAjax);
```

Use XMLHttpRequest objects to interact with servers

Initializes a request

An EventHandler that is called whenever the readyState attribute changes.

readyState values are:

- UNSENT
- OPENED
- HEADERS\_RECEIVED
- LOADING
- DONE

Attaches a click event to the element. When the user clicks on the element, a new ajax request will be triggered.

# THIS

<https://mzl.la/2Idy8rl>

In most cases, the value of **this** is determined by how a function is called

```
function f1() {  
    return this;  
}  
  
f1() === window;  
//= true
```

```
var o = {  
    prop: 37,  
    f2: function () {  
        return this.prop;  
    }  
};  
  
console.log( o.f2() );  
//= 37
```

```
var elm = document.getElementById('my-id');  
  
elm.addEventListener('click', f3);  
  
function f3(e) {  
    console.log( this === e.currentTarget );  
    //= true  
}
```



**this** is set to the element the event fired from

# WHERE DO I WRITE THIS?

---

Script inline 

don't try this at the office

```
<a onclick="alert('Hello World!');">greet</a>
```

not so bad, but avoid it anyway

```
<script>  
  alert('Hello World!');  
</script>
```

load it from file

*new tag is embedded at the end of <body> tag*

```
<script src="application.js"></script>
```



## REFERENCES

- MDN web docs – [https://developer.mozilla.org/en-US/docs/Web/API/Document\\_Object\\_Model](https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model)
- Codecademy – <https://www.codecademy.com/courses/javascript-beginner-en-gwcYv/0/1>
- w3schools – [https://www.w3schools.com/js/js\\_htmldom.asp](https://www.w3schools.com/js/js_htmldom.asp)
- What is the DOM? – <https://css-tricks.com/dom/>

# JS

## JAVASCRIPT & DOM in action



<http://bit.ly/fe4b-js>



JQUERY

# JQUERY

---

- It is not a programming language
- It makes things like HTML document traversal and manipulation, event handling, and Ajax much simpler
- It has an easy-to-use API that works across a multitude of browsers

# VERSIONI

---

- 1.x – support from Internet Explorer 6 (deprecated - only bug-fixing)
- 2.x – support from Internet Explorer 9 (a cleaned jQuery 1.x) (deprecated - only bug-fixing)
- 3.x – support from Internet Explorer 9

# SELECTORS

jQuery uses a CSS similar syntax for matching a set of elements in a document

`document.getElementById( 'my-id' )`



`$( '#my-id' )`

returns the element that has the ID attribute  
with the specified value

`document.getElementsByClassName( 'my-class' )`



`$( '.my-class' )`

returns a collection of all elements in the document with  
the specified class name

`document.getElementsByTagName( 'span' )`



`$( 'span' )`

returns a collection of all elements in the document  
with the specified tag name

`document.getElementsByName( 'email' )`



`$( '[name=email]' )`

returns a collection of all elements in the document  
with the specified name  
(the value of the name attribute)

# API

---

## .before( selector ) - .appendTo( selector )

DOM manipulation

```
<div id="my-id"></div>
```

```
<script>
  $('#my-id').before('this text will be added before div');
</script>
```

## .on( event, callback ) - .off( event )

event handling

```
<a href="https://www.google.it" >link to google.it</a>
```

```
<script>
  $('a').on('click', function() {
    var link = $( this ).attr( 'href' );
  });
</script>
```

## .addClass() - .removeClass() - .attr()

attributes manipulation

```
<div id="my-id" class="my-class"></div>
```

```
<script>
  $('#my-id').addClass('new-class').removeClass('my-class');
</script>
```

## .hide() - .show() - .fadeOut()

animations

```
<div id="my-id">Hello World</div>
```

```
<script>
  $('#my-id').hide();
</script>
```

# AJAX

```
function doAjax() {  
    $.ajax({  
        url: 'ajax-data.json',  
        method: 'POST',  
        data: {  
            key: 'value',  
        },  
        success: function( result ) {  
            alert( result );  
        } );  
    };  
  
    $('#ajax').on('click', doAjax);
```

`$.ajax` is used to perform an asynchronous HTTP (Ajax) request.

A set of key/value pairs that configures the Ajax request. All settings are optional.

`url`: a string containing the URL to which the request is sent.

`method`: the HTTP method to use for the request (e.g. "POST", "GET", "PUT").

`data`: data to be sent to the server.

`success`: a function to be called if the request succeeds.

Attaches a click event to the element.

When the user clicks on the element, a new ajax request will be triggered.

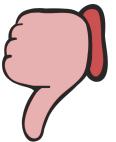
# EVENT DELEGATION

<http://bit.ly/2KisqRk>

Event delegation allows us to **attach** a single event listener, **to a parent element**, that will fire for all descendants matching a selector, whether those **descendants exist now or are added in the future**.

```
<ul id="list">  
  <li>Item #1</li>  
  <li>Item #2</li>  
  <li>Item #3</li>  
  <li>Item #4</li>  
</ul>
```

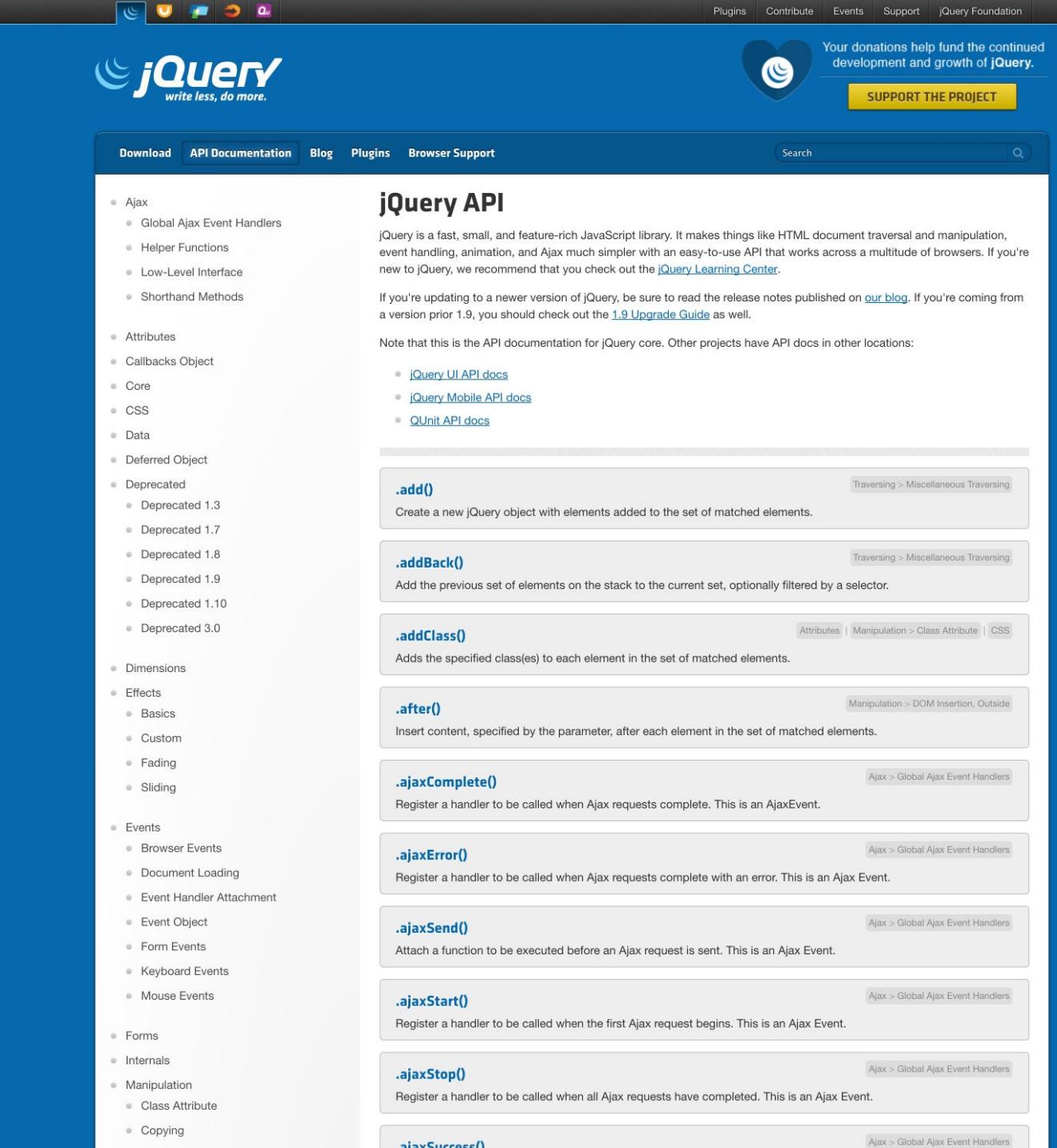
```
$( '#list li' ).on( 'click', function () {  
  console.log( $(this).text() ) ;  
}) ;
```



```
$( '#list' ).on( 'click', 'li', function () {  
  console.log( $(this).text() ) ;  
}) ;
```



```
$( '#list' ).append( '<li>Item #5</li>' ) ;
```



The screenshot shows the official jQuery API documentation website. At the top, there's a navigation bar with links for "Download", "API Documentation" (which is currently selected), "Blog", "Plugins", and "Browser Support". There's also a search bar and a "SUPPORT THE PROJECT" button. The main content area features a sidebar on the left with a hierarchical menu of jQuery API categories, each with a small circular icon next to it. To the right of the sidebar, the "jQuery API" section is displayed, with a brief introduction and links to release notes and other API documentation. Below this, a list of individual API methods is shown in boxes, each with a title, a brief description, and a category link.

**jQuery API**

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. If you're new to jQuery, we recommend that you check out the [jQuery Learning Center](#).

If you're updating to a newer version of jQuery, be sure to read the release notes published on [our blog](#). If you're coming from a version prior 1.9, you should check out the [1.9 Upgrade Guide](#) as well.

Note that this is the API documentation for jQuery core. Other projects have API docs in other locations:

- [jQuery UI API docs](#)
- [jQuery Mobile API docs](#)
- [QUnit API docs](#)

**.add()** Traversing > Miscellaneous Traversing

Create a new jQuery object with elements added to the set of matched elements.

**.addBack()** Traversing > Miscellaneous Traversing

Add the previous set of elements on the stack to the current set, optionally filtered by a selector.

**.addClass()** Attributes | Manipulation > Class Attribute | CSS

Adds the specified class(es) to each element in the set of matched elements.

**.after()** Manipulation > DOM Insertion, Outside

Insert content, specified by the parameter, after each element in the set of matched elements.

**.ajaxComplete()** Ajax > Global Ajax Event Handlers

Register a handler to be called when Ajax requests complete. This is an AjaxEvent.

**.ajaxError()** Ajax > Global Ajax Event Handlers

Register a handler to be called when Ajax requests complete with an error. This is an Ajax Event.

**.ajaxSend()** Ajax > Global Ajax Event Handlers

Attach a function to be executed before an Ajax request is sent. This is an Ajax Event.

**.ajaxStart()** Ajax > Global Ajax Event Handlers

Register a handler to be called when the first Ajax request begins. This is an Ajax Event.

**.ajaxStop()** Ajax > Global Ajax Event Handlers

Register a handler to be called when all Ajax requests have completed. This is an Ajax Event.

**.ajaxSuccess()** Ajax > Global Ajax Event Handlers

# HOW DO I IMPORT IT?

---

Download .zip



old method, please don't use it

CDN

you can use it, depends on what you  
need

npm

downloaded as project  
dependency

now you can import it in your document as a JavaScript file

```
<script src="jquery.min.js"></script>
```



## REFERENCES

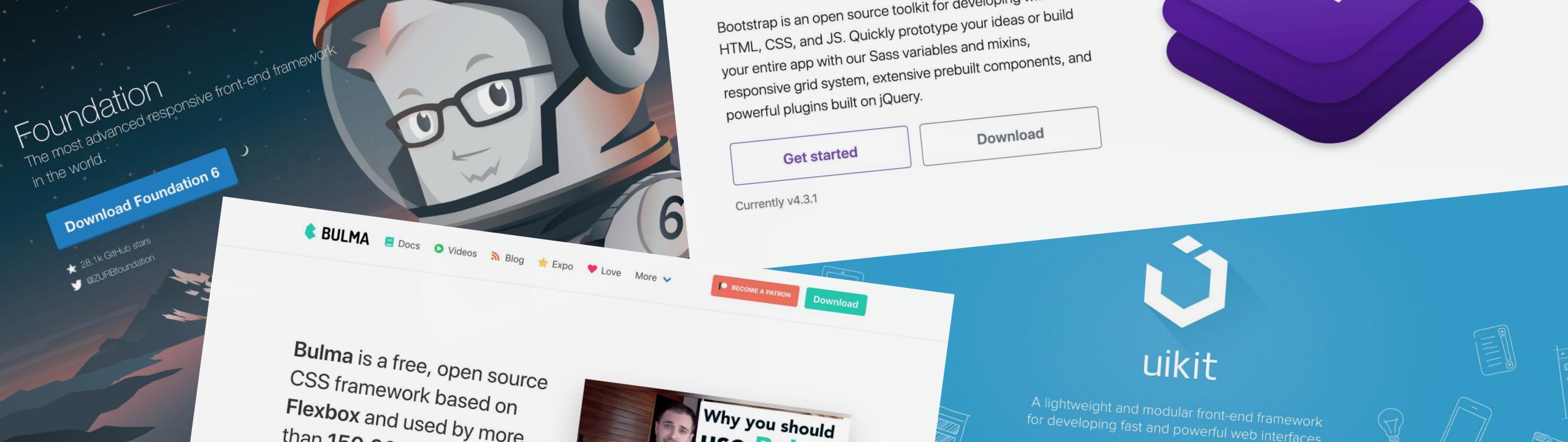
- jQuery – <https://jquery.com/>
- jQuery API – <http://api.jquery.com/>



JQUERY  
in action



<http://bit.ly/fe4b-jquery>



## FRONTEND FRAMEWORKS

# Bootstrap

Build responsive, mobile-first projects on the web with the world's most popular front-end component library.



## BOOTSTRAP

# BOOTSTRAP

---

- It is not a programming language
- Bootstrap is a front-end framework for developing responsive, mobile first projects on the web
- It is made up of HTML, CSS and JavaScript
- Can be extended via lots of plugins
- Bootstrap 4 uses FlexBox

is made up of 4 sections:

- LAYOUT
- CONTENT
- COMPONENTS
- UTILITIES

The screenshot shows the Bootstrap documentation website at <https://getbootstrap.com/docs>. The page has a dark purple header with the Bootstrap logo and navigation links for Home, Documentation, Examples, Themes, Expo, and Blog. A search bar is located in the top right. The sidebar on the left contains several sections: Getting started (Introduction, Download, Contents, Browsers & devices, JavaScript, Theming, Build tools, Webpack, Accessibility), Layout (Content, Components, Utilities, Extend, Migration, About), and a link to Slack. The main content area features an "Introduction" section with a brief overview and a "Quick start" section with instructions for adding Bootstrap via CDN or GitHub. Below that is a "CSS" section with a code snippet for linking to the stylesheet and a "JS" section explaining component dependencies and script placement.

B

Home Documentation Examples Themes Expo Blog

v4.2 [Download](#)

Search...

Getting started

Introduction

Download

Contents

Browsers & devices

JavaScript

Theming

Build tools

Webpack

Accessibility

Layout

Content

Components

Utilities

Extend

Migration

About

Introduction

Get started with Bootstrap, the world's most popular framework for building responsive, mobile-first sites, with BootstrapCDN and a template starter page.

All the tools your team needs in one place. Slack: Where work happens. ads via Carbon

slack

Quick start

Looking to quickly add Bootstrap to your project? Use BootstrapCDN, provided for free by the folks at StackPath. Using a package manager or need to download the source files? [Head to the downloads page](#).

CSS

Copy

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/boo...
```

JS

Many of our components require the use of JavaScript to function. Specifically, they require [jQuery](#), [Popper.js](#), and our own JavaScript plugins. Place the following `<script>`s near the end of your pages, right before the closing `</body>` tag, to enable them. jQuery must come first, then Popper.js, and then our JavaScript plugins.

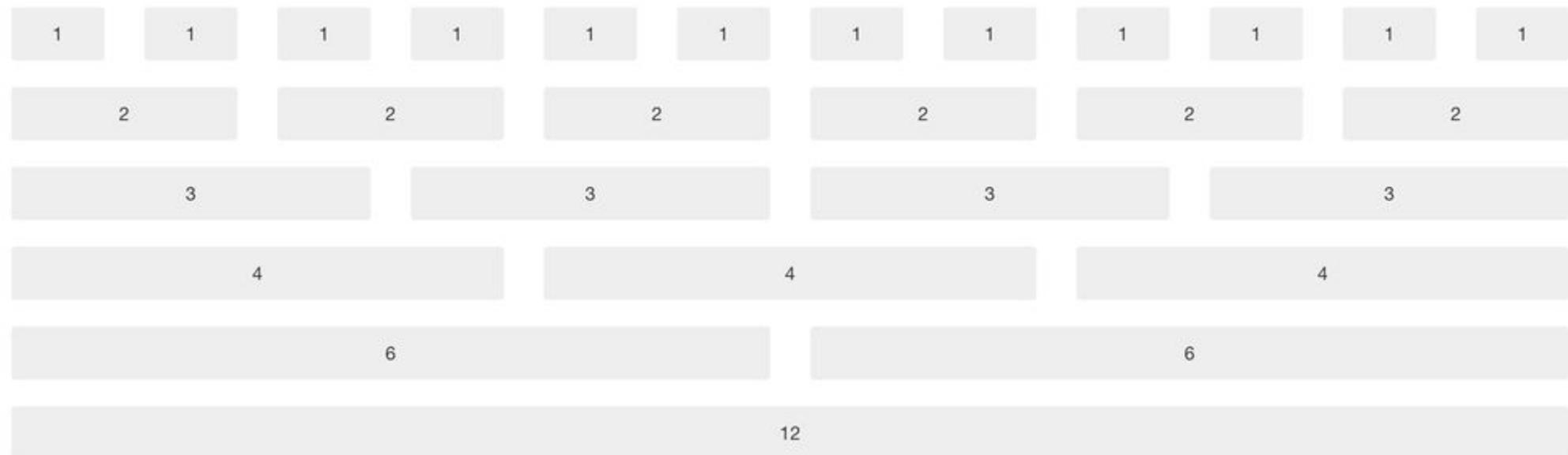
We use [jQuery's slim build](#), but the full version is also supported.

Copy

# LAYOUT - GRID SYSTEM

<https://getbootstrap.com/docs/4.3/layout/grid>

Bootstrap includes a **fluid grid system** that appropriately scales up to **12 columns** as the device or viewport size increases



# LAYOUT - GRID SYSTEM

<https://getbootstrap.com/docs/4.3/layout/grid>

- `.row` are groups of `.col`
- Content (text or other `<tag>`s) must be written inside a `.col`

```
<div class="row">
  <div class="col-xs-12 col-sm-6 col-md-4 col-lg-3"></div>
  <div class="col-xs-12 col-sm-6 col-md-4 col-lg-3"></div>
</div>
```



# LAYOUT - GRID SYSTEM

<https://getbootstrap.com/docs/4.3/layout/grid>

You can set the width of one column and have the sibling columns automatically resize around it

```
<div class="row">
  <div class="col">1 of 3</div>
  <div class="col-6">2 of 3 (wider)</div>
  <div class="col">3 of 3</div>
</div>
```

1 of 3

2 of 3 (wider)

3 of 3

# COMPONENTS

<https://getbootstrap.com/docs/4.3/components>

Bootstrap has **custom classes** to quickly style tags

<https://getbootstrap.com/docs/4.3/components/buttons>

```
<div class="btn btn-primary">Primary</div>
```

Primary

```
<div class="btn btn-primary btn-lg">Large button</div>
```

Large Primary

<https://getbootstrap.com/docs/4.3/components/forms>

```
<input type="text" class="form-control">
```

*focus*

|

# COMPONENTS

Bootstrap **offers** over a dozen **reusable components** built to provide iconography, dropdowns, input groups, navigation, alerts, and much more

Home / Library

```
<ol class="breadcrumb">
  <li class="breadcrumb-item"><a href="/home">Home</a></li>
  <li class="breadcrumb-item active">Library</li>
</ol>
```

```
<div class="input-group">
  <div class="input-group-prepend">
    <div class="input-group-text">@</div>
  </div>
  <input type="text" class="form-control">
</div>
```

@

```
<div class="dropdown">
  <div class="btn btn-secondary dropdown-toggle">
    Dropdown <span class="caret"></span>
  </div>
  <div class="dropdown-menu">
    <button class="dropdown-item">Action</li>
    <button class="dropdown-item">Another action</li>
  </ul>
</div>
```

Dropdown ▾

Action  
Another action

# COMPONENTS

---

Modal title

...

Use Bootstrap's JavaScript modal plugin to add dialogs to your site for lightboxes, user notifications, or completely custom content

```
<div class="btn btn-primary" data-toggle="modal" data-target="#myModal">  
  Launch demo modal  
</div>
```

Launch demo modal

```
<div class="modal fade" id="myModal">  
  <div class="modal-dialog">  
    <div class="modal-content">  
      <div class="modal-header">  
        <h5 class="modal-title">Modal title</h5>  
      </div>  
      <div class="modal-body">  
        ...  
      </div>  
    </div>  
  </div>  
</div>
```

# HOW DO I IMPORT IT?

---

Download .zip



old method, please don't use it

CDN

you can use it, depends on what you  
need

npm

downloaded as project  
dependency

now you can import all files in your document:

```
<link rel="stylesheet" href="bootstrap/css/bootstrap.min.css">
```

```
<script src="bootstrap/js/bootstrap.min.js"></script>
```

# Bootstrap

Build responsive, mobile-first projects on the web with the world's most popular front-end component library.

## REFERENCES

- Bootstrap – <http://getbootstrap.com>
- Github – <https://github.com/twbs/bootstrap>

# Bootstrap

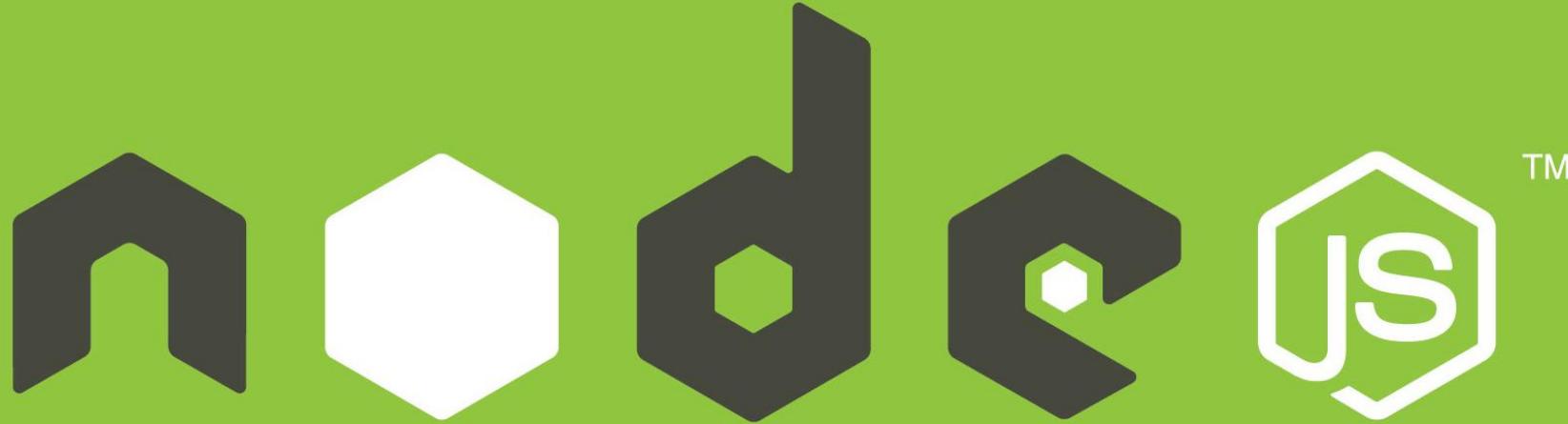
Build responsive, mobile-first projects on the web with the world's most popular front-end component library.



BOOTSTRAP  
in action



<http://bit.ly/fe4b-bootstrap-4>



NODE.JS

# NODE.JS

---

- Node.js is a JavaScript runtime built on **Chrome's V8 JavaScript engine**.
- **V8** is the JavaScript execution engine built for Google Chrome and open-sourced by Google in 2008.
- Node.js uses an **event-driven, non-blocking I/O model** that makes it lightweight and efficient
- It executes JavaScript code **server-side**
- There are a lot of applications today that use Node.js



APACHE

# CORDOVA™

Mobile apps with **HTML, CSS & JS**

Target multiple platforms with **one code base**

Free and **open source**



Reusable code  
across  
platforms



Support for  
offline scenarios



Access native  
device APIs



Build cross platform desktop apps  
with JavaScript, HTML, and CSS

NPM <https://www.npmjs.com>

---

“Node.js' package ecosystem, npm, is the largest ecosystem  
of open source libraries in the world” <https://nodejs.org>

NPM ≈ Maven ≈ Composer ≈ Bundler ≈ ...  
Java                    PHP                    Ruby

# NPM

<https://docs.npmjs.com/files/package.json>

```
{  
  "name": "project-name",  
  "description": "project-desc",  
  "version": "1.0.0",  
  "license": "MIT",  
  "dependencies": {  
    "express": "^4.14.0"  
  },  
  "peerDependencies": {  
    "jquery": "1.9.1 - 3"  
  },  
  "devDependencies": {  
    "mocha": "^3.2.0"  
  }  
}
```

package.json

**package.json** is used to give information to npm that allows it to identify the project as well as handle the project's dependencies.

**devDependencies** are dependencies not required for normal operation, but required/recommended if you want to patch or modify the project ( unit test, lint, builder, ecc. ).

**dependencies** field is used to list all the dependencies of your project that are available on npm. When someone installs your project through npm, all the dependencies listed will be installed as well.

**peerDependencies** are used if you want to express the compatibility of your package with a host tool or library, while not necessarily doing a require of this host.

# NPM <https://docs.npmjs.com>

---

```
{  
  "name": "project-name",  
  "description": "project-desc",  
  "version": "1.0.0",  
  "license": "MIT",  
  "dependencies": {  
    "express": "^4.14.0"  
  },  
  "peerDependencies": {  
    "jquery": "1.9.1 - 3"  
  },  
  "devDependencies": {  
    "mocha": "^3.2.0"  
  }  
}
```

package.json

## npm install - npm update

installs a package, and any packages that it depends on

## npm install <package-name>

installs the specified package and adds it to the package.json

## npm prune

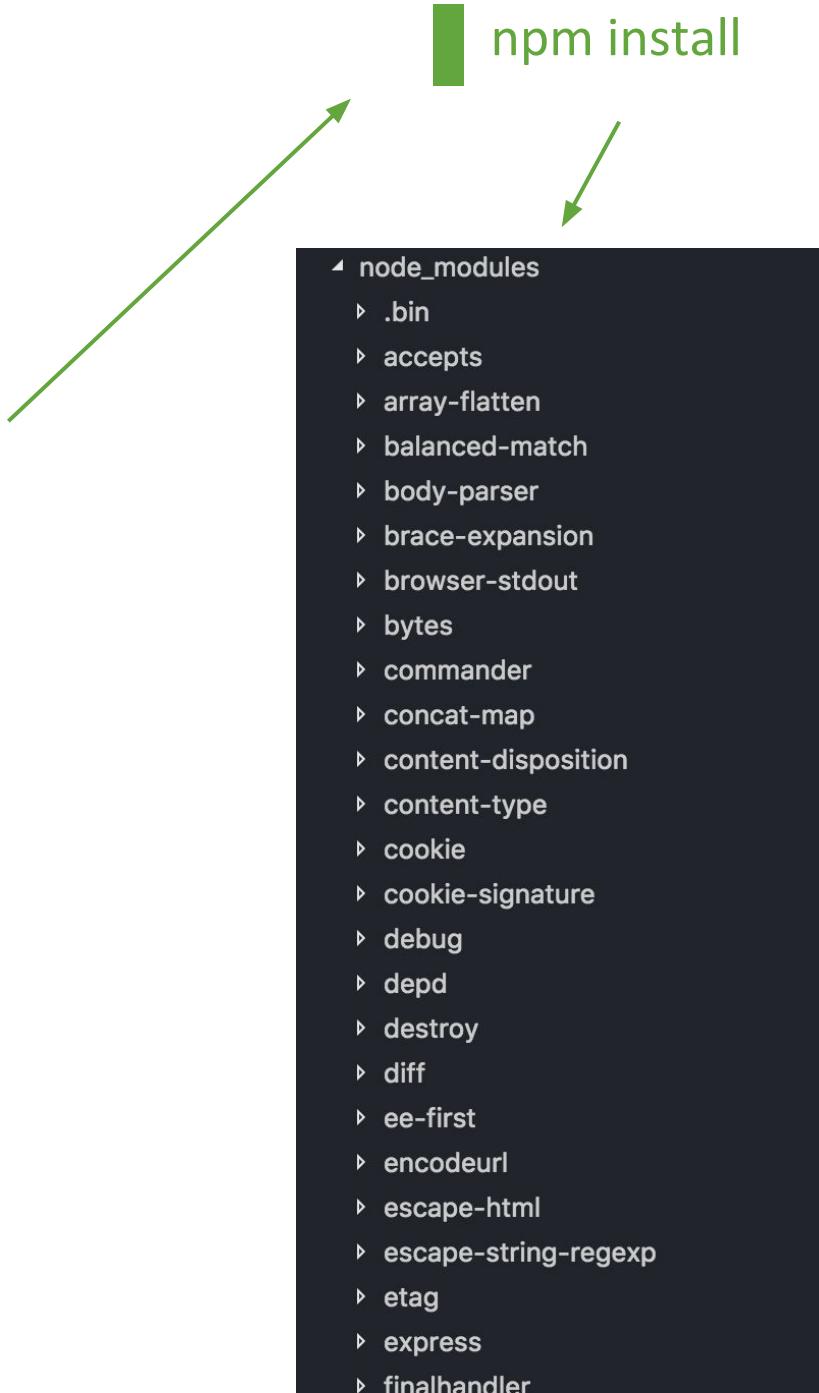
removes packages that are not listed on the parent package's dependencies list

# NPM • node\_modules

```
{  
  "name": "project-name",  
  "description": "project-desc",  
  "version": "1.0.0",  
  "license": "MIT",  
  "dependencies": {  
    "express": "^4.14.0"  
  },  
  "peerDependencies": {  
    "jquery": "1.9.1 - 3"  
  },  
  "devDependencies": {  
    "mocha": "^3.2.0"  
  }  
}
```

package.json

npm install



A folder with name `node_modules` is created. This folder contains all downloaded dependencies.

You don't need to commit and push this folder, so remember to add it to `.gitignore` file.

# HOW DO I INSTALL NODE.JS?

LTS Recommended For Most Users	Current Latest Features
 <b>Windows Installer</b> node-v10.15.3-x86.msi	 <b>macOS Installer</b> node-v10.15.3.pkg

**Windows Installer (.msi)**

**Windows Binary (.zip)**

**macOS Installer (.pkg)**

**macOS Binary (.tar.gz)**

**Linux Binaries (x64)**

**Linux Binaries (ARM)**

**Source Code**

32-bit	64-bit
32-bit	64-bit
64-bit	
64-bit	
64-bit	
ARMv6	ARMv7
ARMv8	
node-v10.15.3.tar.gz	

# WHERE ARE PACKAGES?

<https://www.npmjs.com/search?q=less>

The screenshot shows the npm search results for the query 'less'. The page title is 'less - npm search' and the URL is <https://www.npmjs.com/search?q=less>. The top navigation bar includes links for 'npm Enterprise', 'Products', 'Solutions', 'Resources', 'Docs', and 'Support'. A search bar contains the query 'less', with a 'Search' button and 'Join' / 'Log In' links. The main content area displays 3568 packages found, sorted by popularity. The first result is 'less' (exact match), published by matthew-dean, version 3.9.0, 5 months ago. It has tags like 'compile less', 'css nesting', etc. The second result is 'webpack', published by sokra, version 4.30.0, 2 days ago. It has tags like 'Packs CommonJs/AMD modules for the browser.', etc. The third result is 'stylelint', published by hudochenkov, version 10.0.0, 19 hours ago. It has tags like 'A mighty, modern CSS linter.', etc. The fourth result is 'postcss-less', published by shellscape, version 3.1.4, a month ago. It has tags like 'LESS parser for PostCSS', etc.



# GRUNT

The JavaScript Task Runner

GRUNTJS

# GRUNTJS

---

- GruntJS is a **task runner** for JavaScript
- It improves code quality adding automated tasks for **linting** and **unit testing**
- It **can do** most of that mundane **work for you**—and your team—with basically **zero effort**
- Many of the tasks you need are already available as **Grunt Plugins**

# HOW DO I INSTALL GRUNTJS?

---

```
npm install -g grunt-cli
```

 -g installs the specified package as a global package. It is useful if you want to use it as a command line tool

# GRUNTJS • PLUGINS

---

Plugins are installed adding devDependencies to the package.json

```
{  
  "name": "package-name",  
  "dependencies": {  
    ...  
  },  
  "devDependencies": {  
    "grunt": "^1.0.1",  
    "grunt-contrib-uglify": "^3.3.0",  
    "grunt-contrib-watch": "^1.0.0"  
  }  
}
```

package.json



npm install

# GRUNTJS • Gruntfile.js

---

**Gruntfile.js is used to configure or define tasks and load Grunt plugins.**

This is a base structure for a Gruntfile.js



```
module.exports = function(grunt)
{
  grunt.initConfig({});

});;

};
```

# GRUNTJS • Gruntfile.js

## uglify task configuration.

A subtask called **development** is defined, which minify application.js in application.min.js

## watch task configuration.

A subtask called **js** is defined, which listens to all modifications on specified files. Once a \*.js is modified, the task is automatically run.

Load tasks from the specified Grunt plugin. Plugins must be installed locally via npm, and must be relative to the Gruntfile.

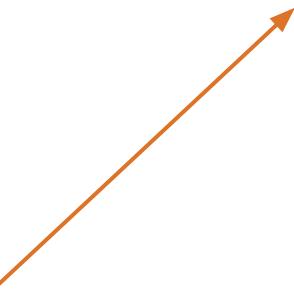
```
module.exports = function(grunt)  
{  
  grunt.initConfig({  
    uglify: {  
      development: {  
        files: {  
          'application.min.js': ['application.js']  
        }  
      }  
    },  
    watch: {  
      js: {  
        files: ['**/*.{js}',  
        tasks: ['uglify']  
      }  
    },  
  });  
  
  grunt.loadNpmTasks('grunt-contrib-uglify');  
  grunt.loadNpmTasks('grunt-contrib-watch');  
};
```

# GRUNTJS • Gruntfile.js

## Gruntfile.js

```
module.exports = function(grunt)
{
  grunt.initConfig({
    uglify: {
      development: {
        files: {
          'application.min.js': ['application.js']
        }
      },
      watch: {
        js: {
          files: ['**/*.js'],
          tasks: ['uglify']
        },
        ...
      }
    });
  });

  grunt.loadNpmTasks('grunt-contrib-uglify');
  grunt.loadNpmTasks('grunt-contrib-watch');
};
```



grunt uglify  
grunt uglify:development

grunt watch  
grunt watch:js

# WHERE ARE PLUGINS?

<http://gruntjs.com/plugins>



## Plugins

This plugin listing is automatically generated from the npm module database. Officially maintained "contrib" plugins are marked with a star ★ icon.

To install and use any plugin listed here, checkout [how to install and use the Grunt plugins section](#).

You may also be interested on how to [create your own Grunt plugin](#).

*In order for a Grunt plugin to be listed here, it must be published on [npm](#) with the `gruntplugin` keyword. Additionally, we recommend that you use the [gruntplugin grunt-init template](#) when creating a Grunt plugin.*

Showing 1 to 100 of 6,082 entries

Search:

← 1 2 3 4 5 →



**Free screencasts**  
about JavaScript,  
Flexbox, Node.js and  
more from the experts  
at [Bocoup](#).

Ads by [Bocoup](#).

Plugin	Downloads last 30 days
★ <a href="#">contrib-watch</a> by Grunt Team Run predefined tasks whenever watched file patterns are added, changed or deleted	960984
★ <a href="#">contrib-clean</a> by Grunt Team Clean files and folders	901859
★ <a href="#">contrib-copy</a> by Grunt Team Copy files and folders	852604
★ <a href="#">contrib-uglify</a> by Grunt Team Minify JavaScript files with UglifyJS	821951
★ <a href="#">contrib-jshint</a> by Grunt Team Validate files with JSHint	781714
★ <a href="#">contrib-concat</a> by Grunt Team	662822



## PREPROCESSORS

# SASS - SCSS

---

- It is a CSS pre-processor
- It extends CSS adding variables, mixins, functions and more
- It makes CSS more maintainable over time
- It can be converted to CSS using different tools

# CSS = SCSS

---

- Same syntax
- Same semantics

# SCSS = CSS + extensions

---

- Nesting
- Variables
- Partials and Import
- Mixins
- Extend/Inheritance
- Operators

## SCSS • NESTED RULES

CSS

```
#header {  
  color: black;  
}  
#header > .navigation {  
  font-size: 12px;  
}  
#header a {  
  color: blue;  
}  
#header a:hover {  
  text-decoration: none;  
}
```



SCSS

```
#header {  
  color: black;  
  
  > .navigation {  
    font-size: 12px;  
  }  
  
  a {  
    color: blue;  
  
    &:hover {  
      text-decoration: none;  
    }  
  }  
}
```

## SCSS • NESTED RULES • @media

---

CSS

```
.component {  
  width: 100%;  
}  
  
.component .blue {  
  color: blue;  
}  
  
@media (min-width: 1200px) {  
  .component {  
    width: 50%;  
  }  
}
```



SCSS

```
.component {  
  width: 100%;  
  
.blue {  
  color: blue;  
}  
  
@media (min-width: 1200px) {  
  width: 50%;  
}
```

# SCSS • VARIABLES

---

- Begin with “\$” dollar signs
- Common property values can be written only once
- You can change multiple values changing just one code line
- Variable names can use hyphens “-” and underscores “\_” interchangeably

```
$size: 100px;
```

```
$my-border: 1px solid #FFFFFF;
```

```
$border-color: #00FF00;
```

```
$my-border: 1px solid $border-color;
```

# SCSS • VARIABLES

```
.component-header {  
  color: #FFFFFF;  
  border-bottom: 1px solid #FFFFFF;  
}  
.component-footer {  
  color: #FFFFFF;  
  background-color: #FF0000;  
}  
.another-component {  
  color: #FF0000;  
}
```

CSS



```
$color-1: #FFFFFF;  
$color-2: #FF0000;  
  
.component-header {  
  color: $color-1;  
  border-bottom: 1px solid $color-1;  
}  
.component-footer {  
  color: $color-1;  
  background-color: $color-2;  
}  
.another-component {  
  color: $color-2;  
}
```

SCSS

## SCSS • INTERPOLATION SYNTAX

---

You can use variables in selectors and property names using

#{} interpolation syntax

```
$name: "foo";  
$attr: "border";  
  
p.#{ $name } {  
  #{$attr}-color: blue;  
}
```

SCSS



```
p.foo {  
  border-color: blue;  
}
```

CSS

## SCSS • @extend

---

Using `@extend` lets you share a set of CSS properties from one selector to another

```
.success {  
  color: green;  
}  
.element-1 {  
  @extend .success;  
  font-weight: bold;  
}  
.element-2 {  
  @extend .success;  
}
```

SCSS

```
.success, .element-1, .element-2 {  
  color: green;  
}  
.element-1 {  
  font-weight: bold;  
}
```

CSS



## SCSS • @extend

---

A placeholder class is a special type of class  
that only prints when it is extended

```
%message-shared {  
  color: green;  
}  
%another-message {  
  color: blue;  
}  
.element {  
  @extend %message-shared;  
}
```

SCSS

```
.element {  
  color: green;  
}
```

CSS

## SCSS • @mixin

---

- Used to define a group of properties and use them later on
- Parametric mixins are by all means functions

```
@mixin border-radius($radius: 10px) {  
  -moz-border-radius: $radius;  
  border-radius: $radius;  
}  
  
#header {  
  @include border-radius(4px);  
}
```

SCSS

```
#header {  
  -moz-border-radius: 4px;  
  border-radius: 4px;  
}
```

CSS



## SCSS • @mixin PRO

---

```
@mixin border-radius($radius: 10px) {  
  
  @if ($radius == 20px) {  
    $radius: $radius * 2px;  
  }  
  
  -moz-border-radius: $radius;  
  border-radius: $radius;  
}  
  
#header {  
  @include border-radius(20px);  
}
```

SCSS



```
#header {  
  -moz-border-radius: 40px;  
  border-radius: 40px;  
}
```

CSS

## SCSS • MATH OPERATIONS

- Suppose we have two TAGs with a total width of 1000px
- One must be  $\frac{3}{4}$  of the other

```
#content {  
    width: 750px;  
}  
  
#sidebar {  
    width: 250px;  
}
```

CSS



```
$width: 1000px;  
$contentWidth: $width * (3 / 4);  
$sidebarWidth: $width - $contentWidth;  
  
#content {  
    width: $contentWidth;  
}  
  
#sidebar {  
    width: $sidebarWidth;  
}
```

SCSS

## SCSS • MATH FUNCTIONS

---

- `ceil( 2.4 ) → 3`
- `floor( 2.6 ) → 2`
- `round( 1.67 ) → 2`
- `percentage( 0.5 ) → 50%`
- `sqrt( 25px ) → 5px`
- `abs( -15px ) → 15px`
- `min( 5, 10 ) → 5`
- `max( 5, 10 ) → 10`

## SCSS • COLOR FUNCTIONS

---

- `rgb( 90, 129, 32 ) → #5a8120`
- `red( #5a8120 ) → 90`
- `green( #5a8120 ) → 129`
- `blue( #5a8120 ) → 32`

- `lighten( #80e619, 20% ) → #b3f075`



- `darker( #80e619, 20% ) → #4d8a0f`



# SCSS • IMPORT

You can import .scss files inside other .scss files

```
@import "this-is-valid";
```

```
@import "variable";
@import "header";
@import "footer";
...
```

application.scss

```
.header {
```

```
    ...
}
```

\_header.scss

```
.footer {
```

```
    color: $color-1;
```

```
}
```

\_footer.scss

```
$color-1: #FFFFFF;
$color-2: #FF0000;
...
```

\_variables.scss

## SCSS • SCOPE

---

Variables and mixins are first looked for locally, and if they aren't found, the compiler will look in the parent scope, and so on

```
$var: red;  
  
#header {  
  $var: white;  
  color: $var; // white  
}  
  
$var: blue;
```

## HOW DO I COMPILE IT?

---

1

npm install -g node-sass



-g installs the specified package as a global package. It is useful if you want to use it as a command line tool

2

node-sass input.scss output.css



## REFERENCES

- Sass/Scss – <http://sass-lang.com>
- Less – <http://lesscss.org>



## PREPROCESSORS in action



<https://www.sassmeister.com/>

# USEFUL LINKS

---

- Can I Use? – <https://caniuse.com>
- CSS-Trick – <https://css-tricks.com>
- Codrops – <https://tympanus.net/codrops>
- Smashing Magazine – <https://www.smashingmagazine.com>
- Mozilla – <https://developer.mozilla.org>
- Google – <https://developers.google.com/web>



# Thank You!

*Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.*

# GITHUB

<https://github.com/marcomontalbano/an-introduction-to-frontend>