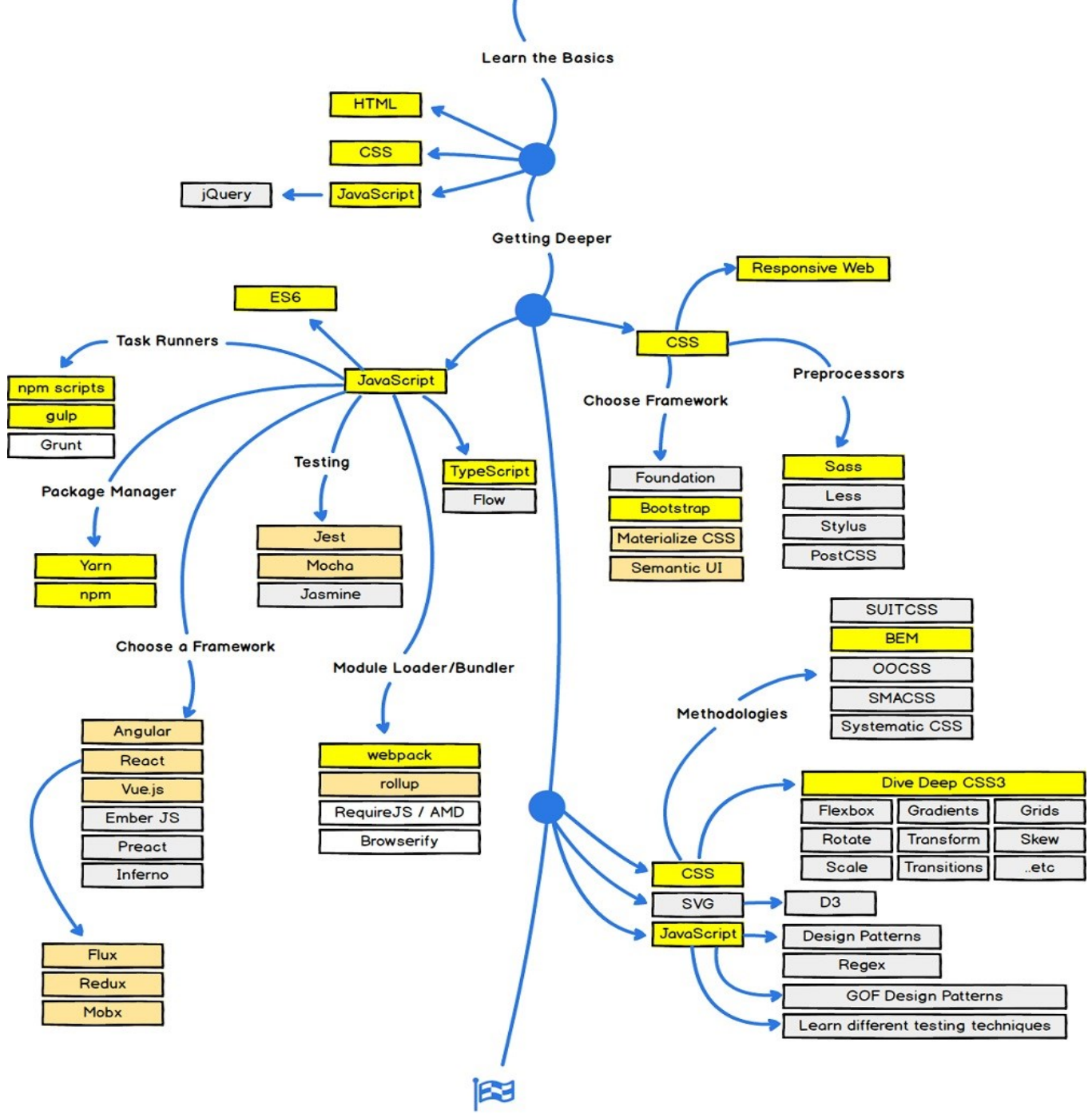
Web Development

Full stack web development

Web basic


Learn the Basics
HTML
CSS
jQuery
JavaScript
Getting Deeper
ES6
Task Runners
npm scripts
gulp
Grunt
JavaScript
Package Manager
Yarn
npm
Testing
Jest
Mocha
Jasmine
TypeScript
Flow
Choose a Framework
Angular
React
Vue.js
Ember JS
Preact
Inferno
Flux
Redux
Mobx
Module Loader/Bundler
webpack
rollup
RequireJS / AMD
Browserify
Responsive Web
CSS
Choose Framework
Foundation
Bootstrap
Materialize CSS
Semantic UI
Preprocessors
Sass
Less
Stylus
PostCSS
Methodologies
SUITCSS
BEM
OOCSS
SMACSS
Systematic CSS
CSS
Dive Deep CSS3
Flexbox
Gradients
Grids
Rotate
Transform
Skew
Scale
Transitions
..etc
SVG
D3
JavaScript
Design Patterns
Regex
GOF Design Patterns
Learn different testing techniques

Web basic

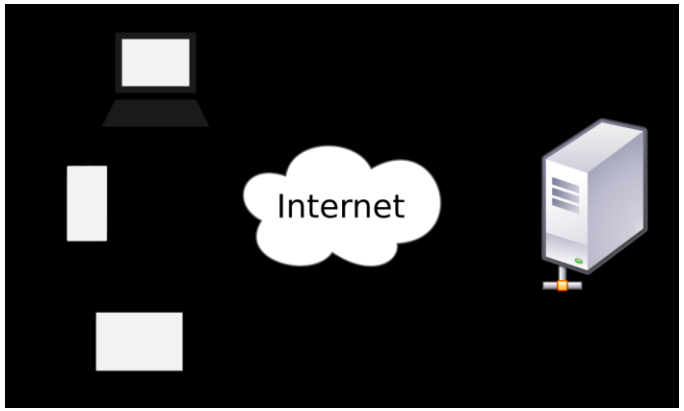
- Every **Web Developer** must have a basic understanding of **HTM**, **CSS**, and **JavaScript**.

- **Responsive Web Design** is used in all types of modern web development.

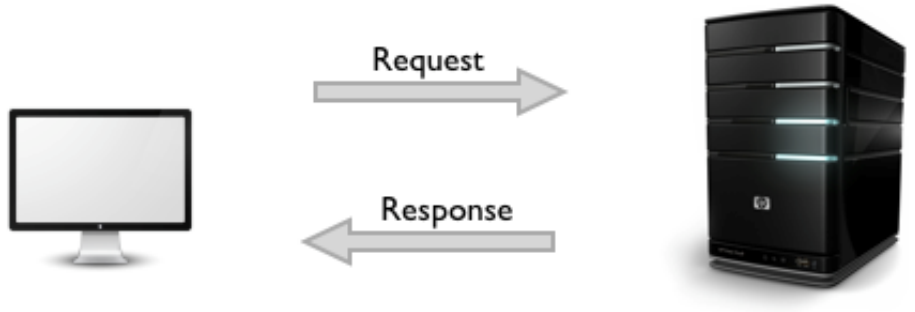
- **ECMAScript 5** (JavaScript 5) is supported in all modern browsers

Web basic

- The World Wide Web is about communication between web clients and web servers.
- Clients are often browsers (Chrome, Edge, Safari), but they can be any type of program or device.
- Servers are most often computers in the cloud.



HTTP req+resp



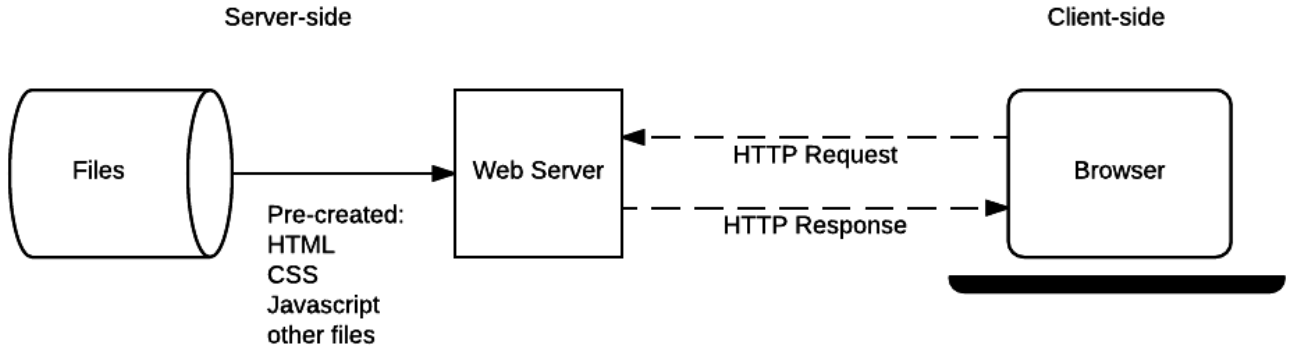
Communication between clients and servers is done by **requests** and **responses**:

1. A client (a browser) sends an **HTTP request** to the web
2. An web server receives the request
3. The server runs an application to process the request
4. The server returns an **HTTP response** (output) to the browser
5. The client (the browser) receives the response

HTTP Request

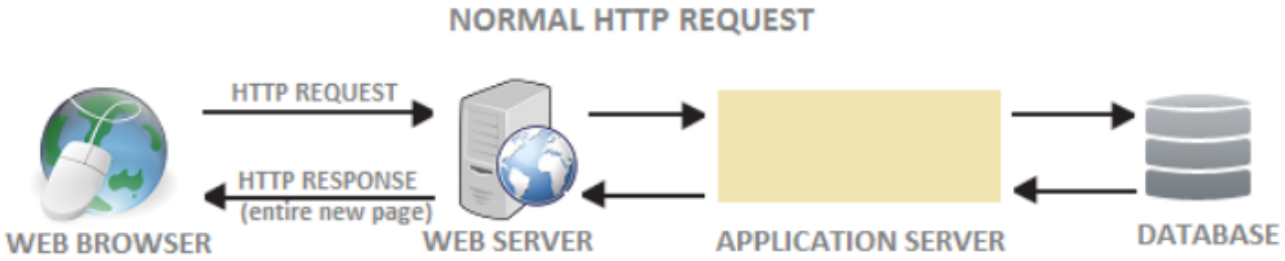
A typical HTTP request / response circle:

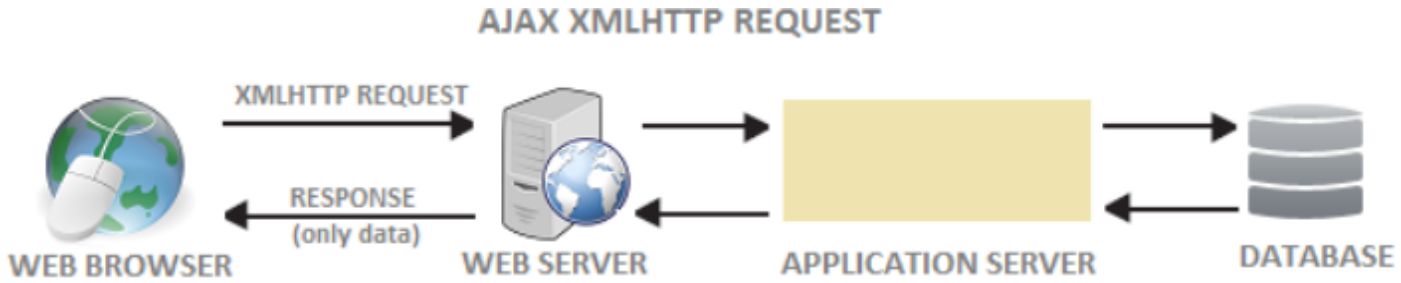
- The browser requests an HTML page. The server returns an HTML file.
- The browser requests a style sheet. The server returns a CSS file.
- The browser requests an JPG image. The server returns a JPG file.
- The browser requests JavaScript code. The server returns a JS file
- The browser requests data. The server returns data (in XML or JSON).



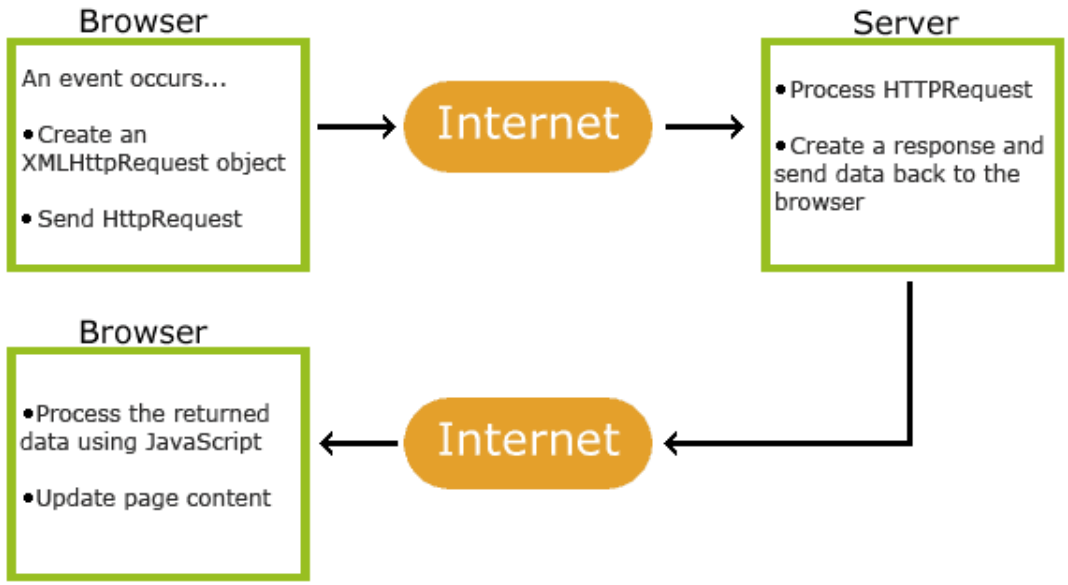
XHR

- All browsers have a built-in **XMLHttpRequest Object (XHR)**.
- XHR is a JavaScript object that is used to transfer data between a web browser and a web server.
- XHR is often used to request and receive data for the purpose of modifying a web page.
- Despite the XML and Http in the name, XHR is used with other protocols than HTTP, and the data can be of many different types like HTML, CSS, XML, JSON, and plain text.





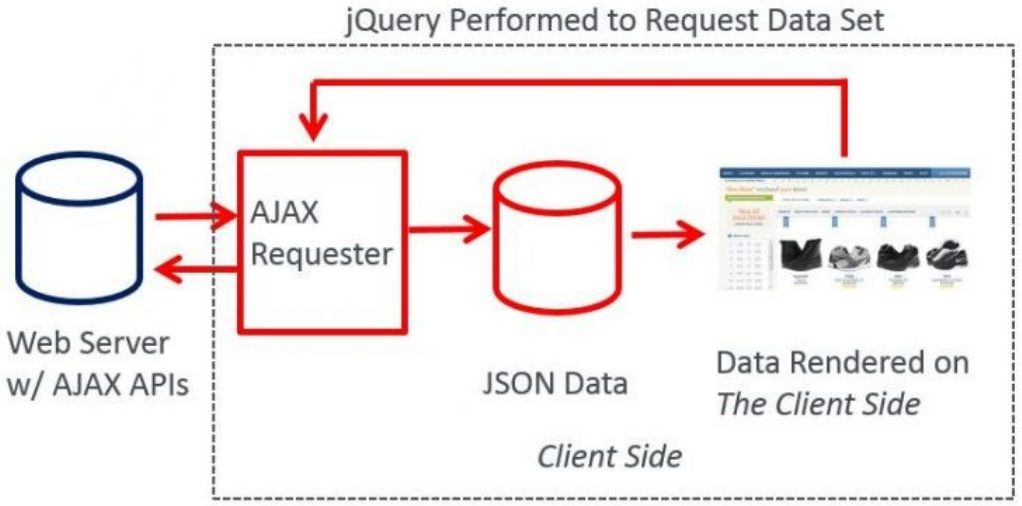
XHR



The XHR Object is a **Web Developers Dream**, because you can:

- Update a web page without reloading the page
- Request data from a server - after the page has loaded
- Receive data from a server - after the page has loaded
- Send data to a server - in the background

- The XHR Object is the underlying concept of **AJAX** and **JSON**:



Html

- HTML stands for **H**yper **T**ext **M**arkup **L**anguage
- HTML is the **standard markup** language for Web pages
- HTML **elements** are the building blocks of HTML pages
- HTML elements are represented by **<> tags**

Html Elements

- An HTML element is a **start** tag and an **end** tag with content in between:
- Es. <h1>This is a Heading</h1>

Html Attributes

- HTML elements can have **attributes**
- Attributes provide **additional information** about the element
- Attributes come in name/value pairs like **charset="utf-8“**

```
<!DOCTYPE html>
<html lang="en">

<meta charset="utf-8">
<title>Page Title</title>

<body>
  <h1>This is a Heading</h1>
  <p>This is a paragraph.</p>
  <p>This is another paragraph.</p>
</body>

</html>
```

- The <!DOCTYPE html> declaration defines this document to be HTML5
- The <html> element is the root element of an HTML page
- The lang attribute defines the language of the document

- The <meta> element contains meta information about the document
- The charset attribure defines the character set used in the document
- The <title> element specifies a title for the document

- The <body> element contains the visible page content
- The <h1> element defines a large heading
- The <p> element defines a paragraph

Html Documents

- All HTML documents must start with a document type declaration: <!DOCTYPE html>.
- The HTML document itself begins with <html> and ends with </html>.
- The visible part of the HTML document is between <body> and </body>.

Html Document Structure

```
<html>
<head>
<title>Page title</title>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
</body>
</html>
```

- **Note:** Only the content inside the <body> section (the white area above) is displayed in a browser.

Html Headings and paragraph

- HTML **headings** are defined with <h1> to <h6> tags.
- <h1> defines the most important heading. <h6> defines the least important heading:
- **esempio heading**

- HTML **paragraphs** are defined with tags:
  - Es: This is another paragraph.

Html links

- HTML **links** are defined with <a> tags:
- Es.: This is a link

- HTML **images** are defined with <img> tags.
- The source file (src), alternative text (alt), width, and height are provided as attributes:
- Es. <img src="img_w3schools.jpg" alt="W3Schools" style="width:120px;height:150px"

Html links

- HTML **links** are defined with <a> tags:
- Es.: This is a link

- HTML **images** are defined with <img> tags.
- The source file (src), alternative text (alt), width, and height are provided as attributes:
- Es. <img src="img_w3schools.jpg" alt="W3Schools" style="width:120px;height:150px"

Html

HTML **buttons** are defined with <button> tags:

- Example

<button>Click me</button>

- HTML **lists** are defined with <ul> (unordered/bullet list) or <ol> (ordered/numbered list) tags, followed by <li> tags (list items):

Example

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Html

An HTML **table** is defined with a <table> tag.

- Table rows are defined with <tr> tags.
- Table headers are defined with <th> tags. (bold and centered by default).
- Table cells (data) are defined with <td> tags.
- esempio tabella + css

Html

Every HTML element can have attributes.

For web development and programming, the most important attributes are id and class. These attributes are often used to address program based web page manipulations.

Attribute	Example
id	<table **id**="table01"
class	<p **class**="normal">
style	<p **style**="font-size:16px">
data-	<div **data**-id="500">
onclick	<input **onclick**="myFunction()">
onmouseover	<a **onmouseover**="this.set:Attribute('css','color:red')">

Html

Every HTML element can have attributes.

For web development and programming, the most important attributes are id and class. These attributes are often used to address program based web page manipulations.

Attribute	Example
id	<table **id**="table01"
class	<p **class**="normal">
style	<p **style**="font-size:16px">
data-	<div **data**-id="500">
onclick	<input **onclick**="myFunction()">
onmouseover	<a **onmouseover**="this.set:Attribute('css','color:red')">

Html

- Tag reference html
- Esercizi
- Quiz
- Tutorial HTML

CSS

- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed

- esempio CSS

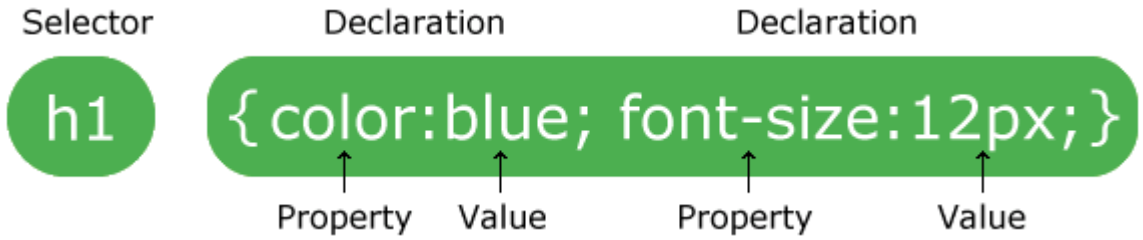
```
<style>

body {background-color:lightblue; text-align:center;}
h1 {color:blue; font-size:40px;}
p {font-family:verdana; font-size:20px;}

</style>
```

CSS

- A **CSS rule** consists of a **selector** and a **declaration** block:



- The selector points to the HTML element to style (h1).
- The declaration block (in curly braces) contains one or more declarations separated by semicolons.
- Each declaration includes a CSS **property name** and a **value**, separated by a colon.
- In the following example all <p> elements will be 32px wide, center-aligned, and with red:

```
<style>
p {
   font-size: 32px;
   color: red;
   text-align: center;}
</style>
```

CSS

- A CSS style sheet can be stored in an **external file** (es.mystyle.css)

```
body {background-color: orange; font-family: verdana}
h1 {color: white;}
p {font-size: 20px;}
```

- External style sheets are linked to HTML pages with <link> tags:

```
<!DOCTYPE html>
<html>
<link rel="stylesheet" href="mystyle.css">
<body>
<h1>My First CSS Example</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

- Esempio

CSS

If different styles are specified for HTML elements, the styles will **cascade** into new styles with the following priority:

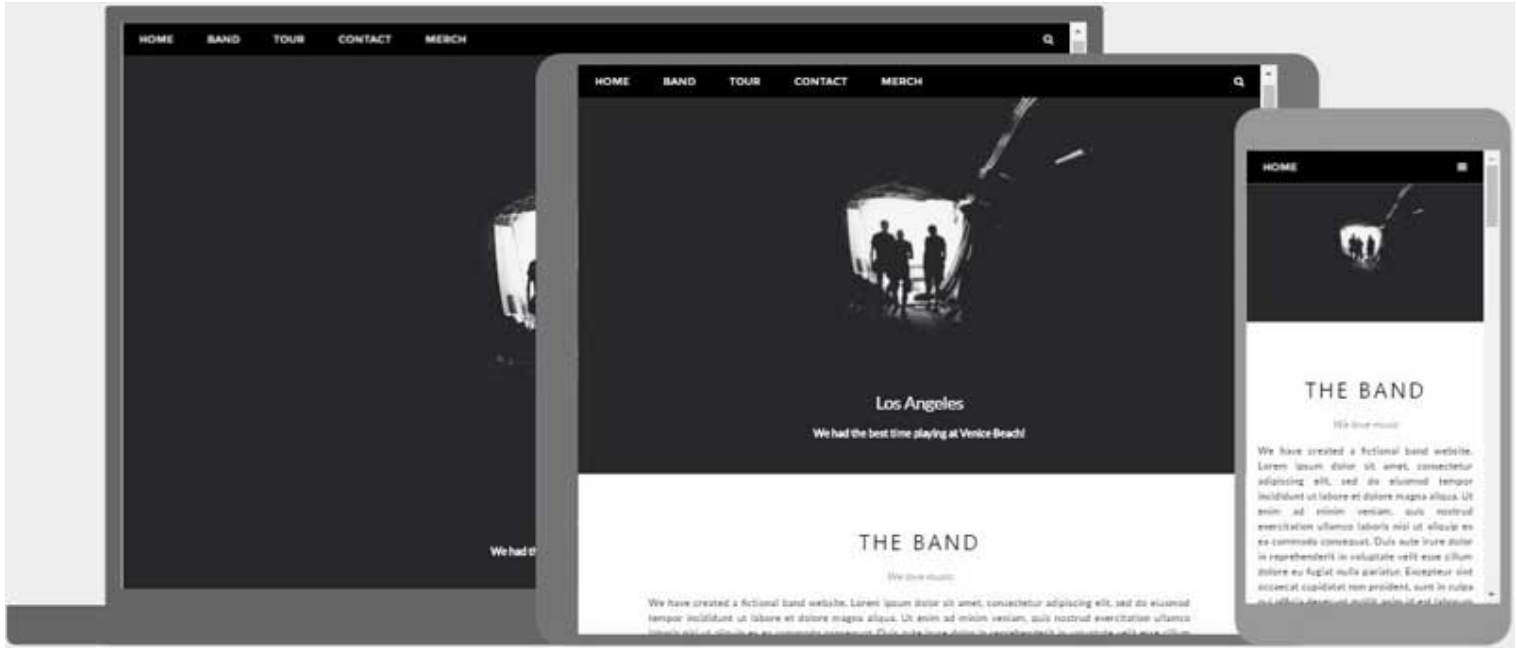
- Priority 1: Inline styles
- Priority 2: External and internal style sheets
- Priority 3: Browser default
- If different styles are defined on the same priority level, the last one has the highest priority.

- Esempio

CSS

- Tutorial: https://www.w3schools.com/css/default.asp
- Full Css reference : W3Schools CSS Reference

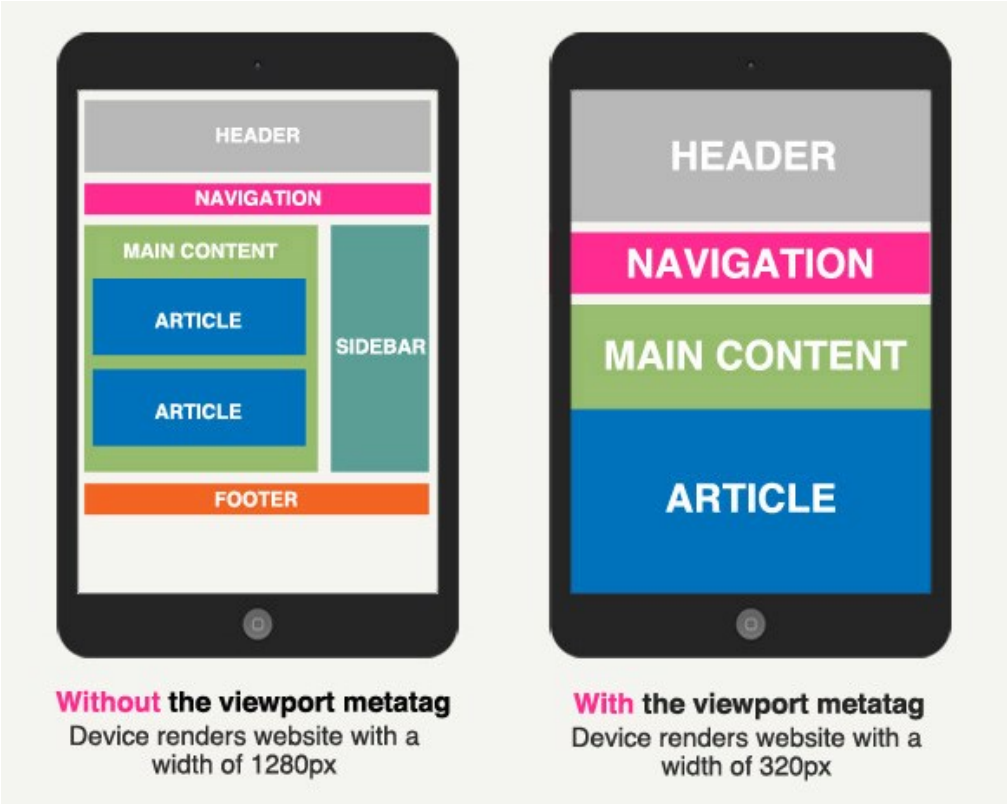
Responsive web design



- Responsive Web Design is about using HTML and CSS to automatically resize a website.
- Responsive Web Design is about making a webside look good on all devices (desktops, tablets, and phones):

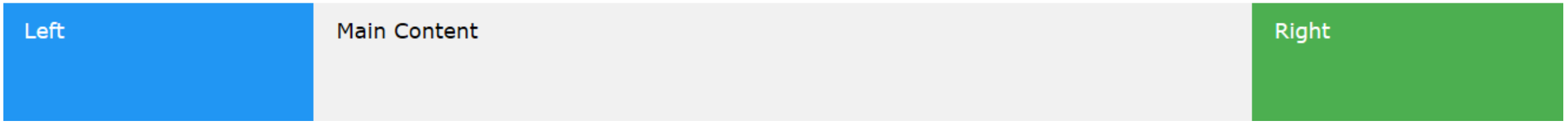
Responsive Web Design

- Setting The Viewport
- When making responsive web pages, add the following <meta> element to all your web pages:
- Esempio
- <meta name="viewport" content="width=device-width, initial-scale=1.0">



Responsive Web Design

- Media Queries plays an important role in responsive web pages.
- With media queries you can define different styles for different browser sizes.
- Example:
  Resize the browser window to see that the three elements below will display horizontally on large screens and vertically on small screens:



- Esempio

Responsive Web Design

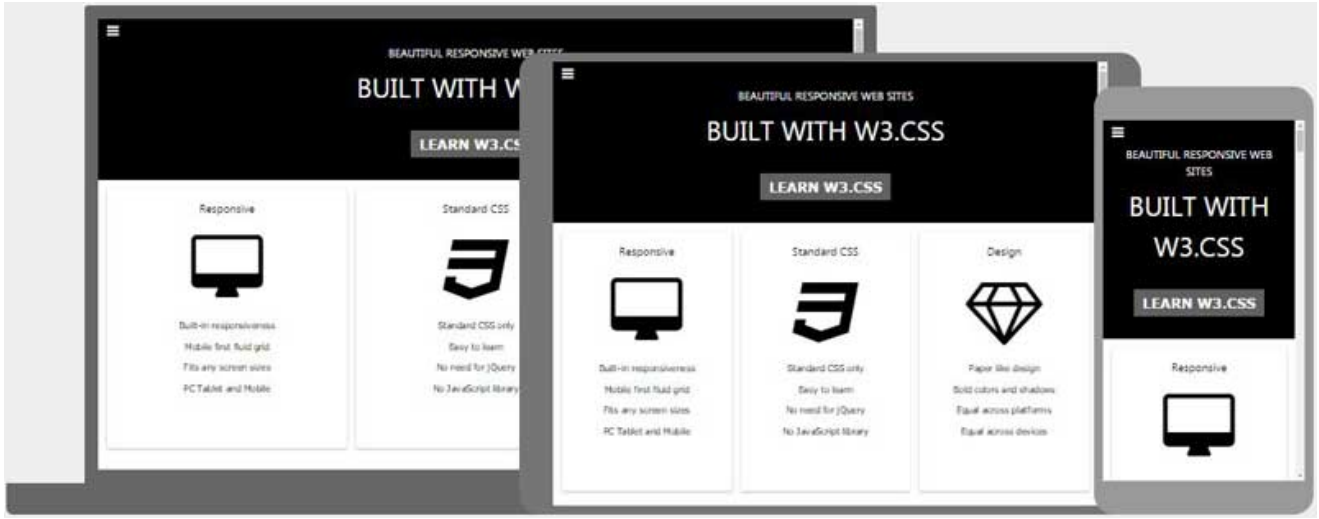
- Responsive images are images that scale nicely to fit any browser size.
- When the CSS width property is set to a percentage value, an image will scale up and down when resizing the browser window.
- This image is responsive:

- Esempio1

- If the max-width property is set to 100%, the image will scale down if it has to, but never scale up to be larger than its original size:
- Esempio2

Responsive Web Design

- The HTML <picture> element allows you to define different images for different browser window sizes.
- Esempio 1


BUILT WITH W3.CSS
LEARN W3.CSS
Responsive
Standard CSS
Design
BUILT WITH
W3.CSS
LEARN W3.CSS
Responsive

[Esempio]

Responsive Web Design

- Bootstrap is a very popular framework that uses HTML, CSS and jQuery to make responsive web pages.
- Esempio 1

Javascript

- JavaScript is the Programming Language for the Web
- JavaScript can update and change both HTML and CSS
- JavaScript can calculate, manipulate and validate data

Data Types
JavaScript variables can be:

1. Numbers
2. Strings
3. Objects
4. Arrays
5. Functions

Javascript

- JavaScript variables are containers for storing data values.
- In this example, x, y, and z, are variables:

```
var x = 5;
var y = 6;
var z = x + y;
```

Esempio

- JavaScript has only **one type** of number. Numbers can be written with or without decimals.

```
var x = 3.14;    // A number with decimals
var y = 3;       // A number without decimals
```

- All numbers are stored as double precision floating point numbers.
- The maximum number of decimals is 17, but floating point is not always 100% accurate:

Example

```
var x = 0.2 + 0.1;         // x will be 0.30000000000000004
```

Javascript

Strings **store text**. Strings are written inside quotes. You can use **single** or double **quotes**:
Example
```
var carname = "Volvo XC60";  // Double quotes
var carname = 'Volvo XC60';  // Single quotes
```
Esempio

The length of a string is found in the built in property **length**:
Example
```
var txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
var sln = txt.length;
```
Try it Yourself

Javascript Objects

- This code assigns a simple value (Fiat) to a variable named car:

```
var car = "Fiat";
```

- Objects are variables too. But objects can contain many values.

This code assigns **many values** (Fiat, 500, white) to a **variable** named car:

```
var car = {type:"Fiat", model:"500", color:"white"};
```

https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis_js_objects_object

- JavaScript **arrays** are used to store multiple values in a single variable.

```
var cars = ["Saab", "Volvo", "BMW"];
```

https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis_js_array

JavaScript Functions

Javascript functions

- A JavaScript function is a block of code designed to perform a particular task.
- A JavaScript function is executed when "something" invokes it (calls it).

```
function myFunction(p1, p2) {
   return p1 * p2;              // The function returns the product of p1 and p2
}
```

Esempio

Javascript

- JavaScript Can Change HTML Content
- JavaScript Can Change HTML Attribute Values
- JavaScript Can Change HTML Styles CSS)
- JavaScript Can Hide HTML Elements
- JavaScript Can Show HTML Elements

- One of many JavaScript HTML methods is **getElementById()**.

This example uses the method to "find" an HTML element (with id="demo") and changes the element content (**innerHTML**) to "Hello JavaScript":

```
document.getElementById("demo").innerHTML = "Hello JavaScript";
```

Try it Yourself

- In this example JavaScript changes the value of the src (source) attribute of an <img> tag:
- Esempio

- Changing the style of an HTML element, is a variant of changing an HTML attribute:

```
document.getElementById("demo").style.fontSize = "35px";
```

Try it Yours

Javascript

- Hiding HTML elements can be done by changing the display style:

document.getElementById("demo").style.display = "none";

Try it Yoursel

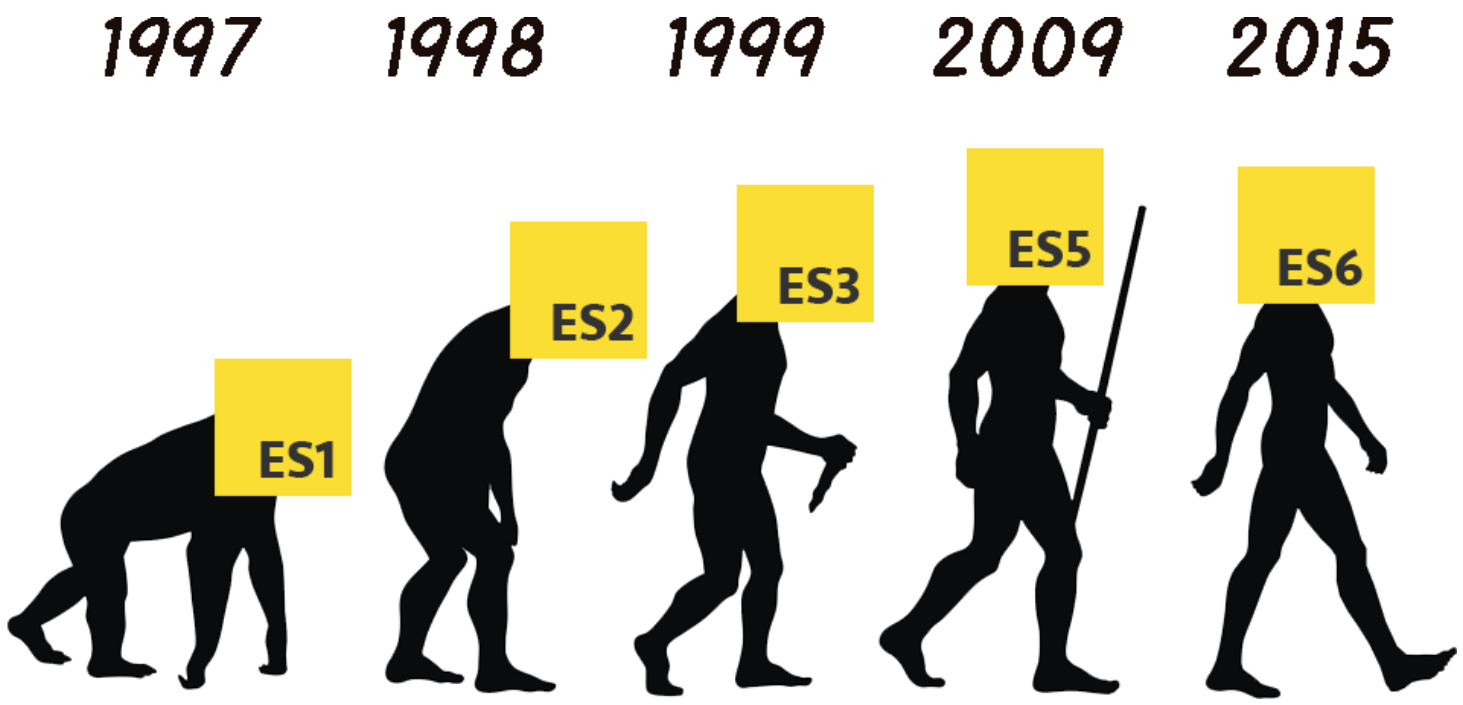
- Showing hidden HTML elements can also be done by changing the display style:

Example

document.getElementById("demo").style.display = "block";

Try it Yourself

- ES5 is a shortcut for ECMAScript 5
- ECMAScript 5 is also known as JavaScript 5
- ECMAScript 5 is also known as ECMAScript 2009 because released in 2009

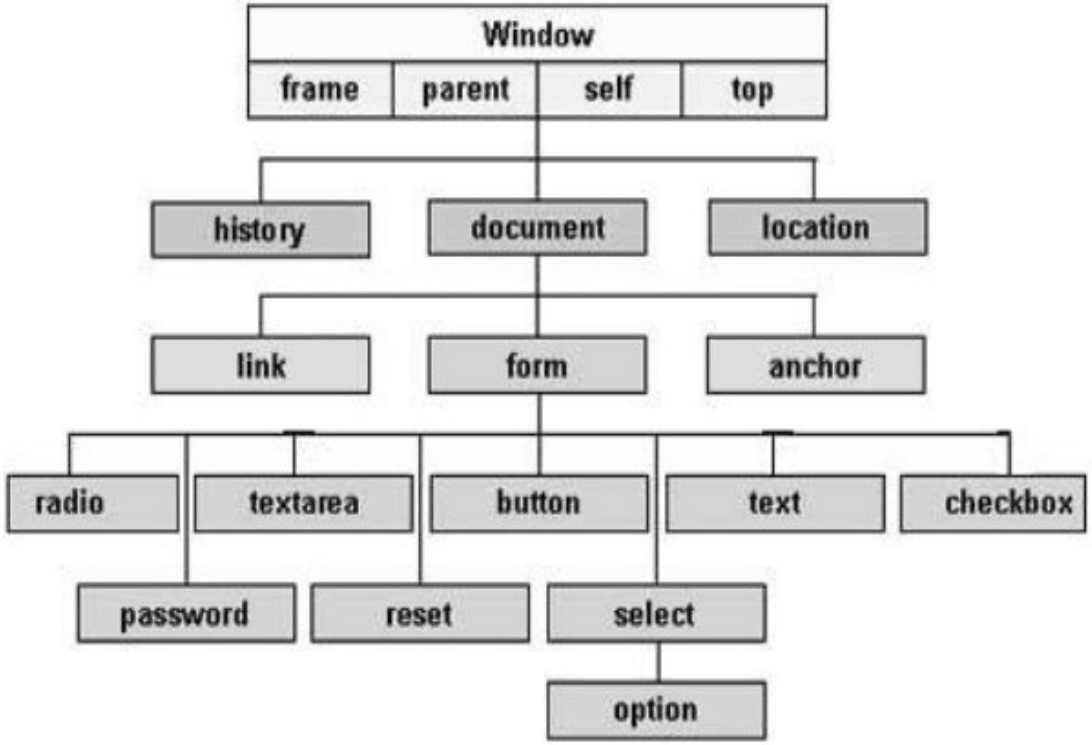


Html Dom

- The HTML DOM is an Object Model for HTML. It defines:
- HTML elements as objects
- Properties for all HTML elements
- Methods for all HTML elements
- Events for all HTML elements

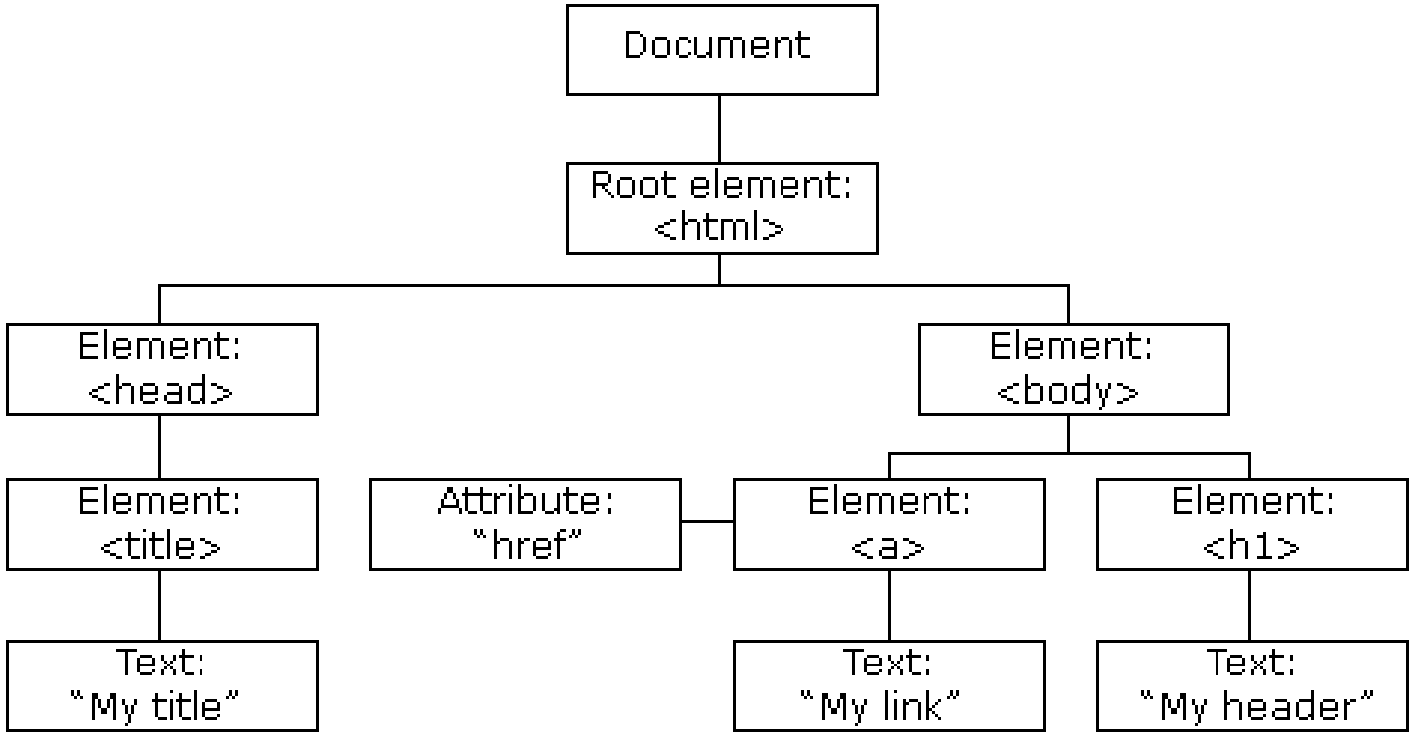
- The HTML DOM is an API (Programming Interface) for JavaScript:

- JavaScript can add/change/remove HTML elements
- JavaScript can add/change/remove HTML attributes
- JavaScript can add/change/remove CSS styles
- JavaScript can react to HTML events
- JavaScript can add/change/remove HTML events



Html Dom

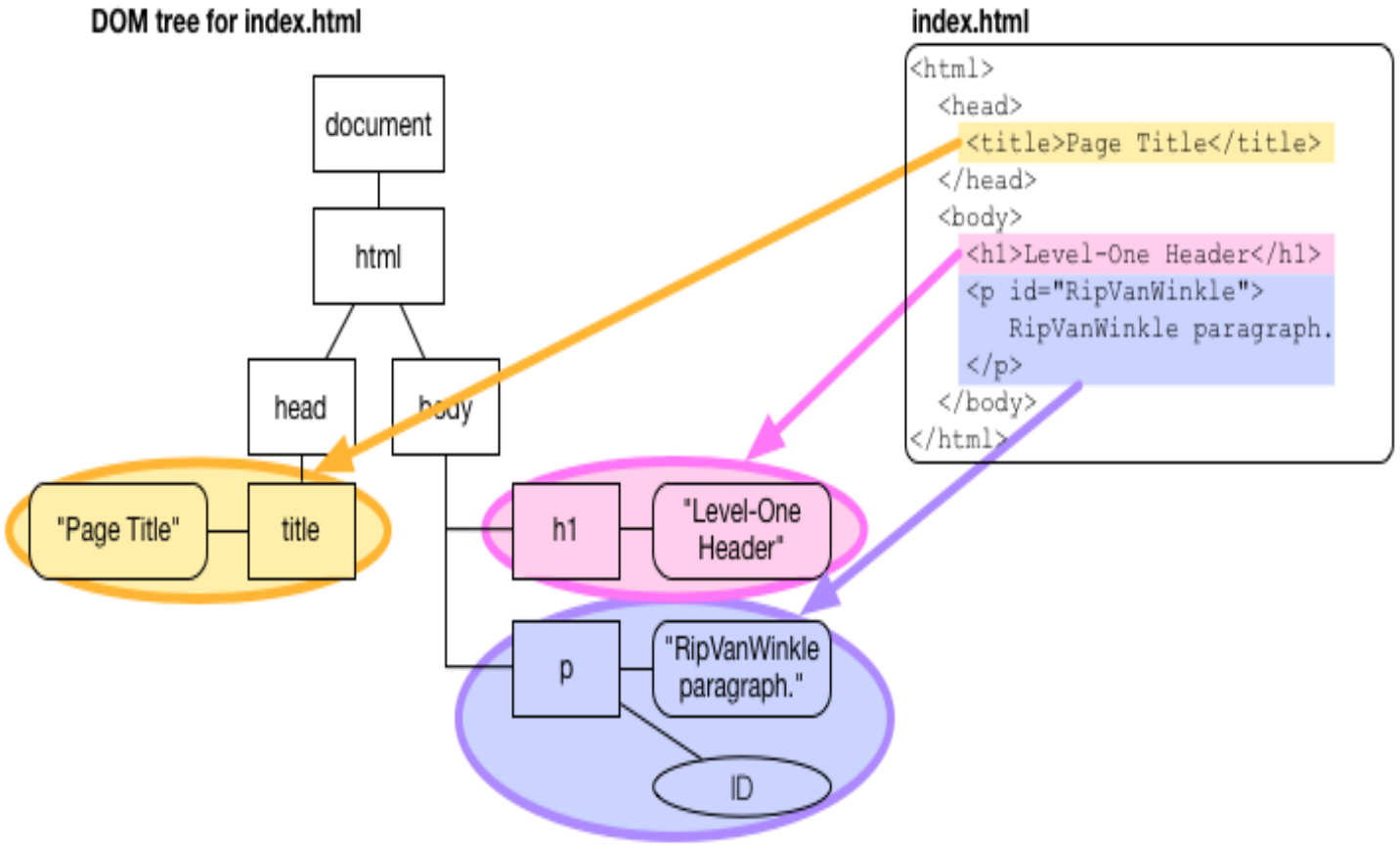
- When a web page is loaded, the browser creates a **D**ocument **O**bject **M**odel of the page.
- The **HTML DOM** model is constructed as a tree of **Objects**:



HTML Dom: Find elements

- When you want to access HTML elements with JavaScript, you have to find the elements first.
- There are a couple of ways to do this:

- Finding HTML elements by id
- Finding HTML elements by tag name
- Finding HTML elements by class name
- Finding HTML elements by CSS selectors
- Finding HTML elements by HTML object collections



Html Dom: Find elements

- This example finds the element with id="intro":

var myElement = document.getElementById("intro");
Try it Yourself »

If the element is found, the method will return the element as an object (in myElement).
If the element is not found, myElement will contain null.

- This example finds all <p> elements:

var x = document.getElementsByTagName("p");
Try it Yourse

- This example finds the element with id="main", and then finds all <p> elements inside "main":

var x = document.getElementById("main");
var y = x.getElementsByTagName("p");
Try it Y

Html Dom: Find elements

- If you want to find all HTML elements with the same class name, use getElementsByClassName().
- This example returns a list of all elements with class="intro".

- var x = document.getElementsByClassName("intro");
- Esempio

Finding elements by class name does not work in Internet Explorer 8 and earlier versions.

- If you want to find all HTML elements that matches a specified CSS selector (id, class names, types, attributes, values of attributes, etc), use the querySelectorAll() method.

This example returns a list of all <p> elements with class="intro".
var x = document.querySelectorAll("p.intro");
Try it Yourself »
The querySelectorAll() method does not work in Internet Explorer 8 and earlier versions.

Html Dom: Find elements

- HTML object collections are also accessible:

- document.anchors
- https://www.w3schools.com/js/tryit.asp?filename=tryjs_doc_anchors

- document.forms
- https://www.w3schools.com/js/tryit.asp?filename=tryjs_doc_forms

- document.images
- https://www.w3schools.com/js/tryit.asp?filename=tryjs_doc_images

- document.links
- https://www.w3schools.com/js/tryit.asp?filename=tryjs_doc_links

- document.scripts
- https://www.w3schools.com/js/tryit.asp?filename=tryjs_doc_scripts

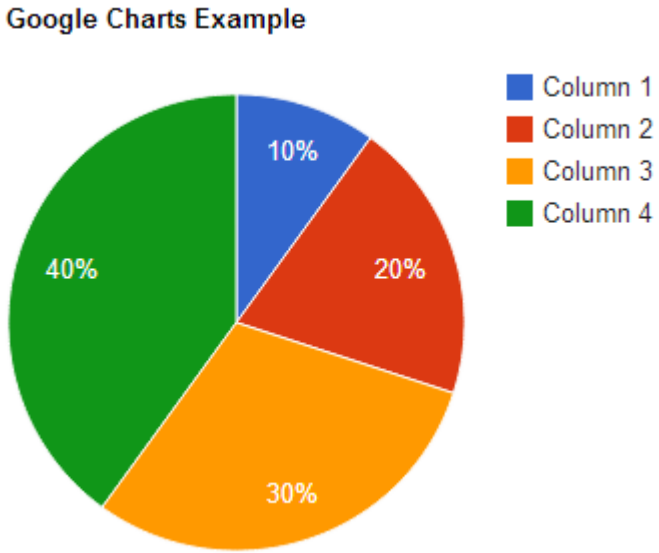
Google Maps APIs

- https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis_google_maps

*The **mapCanvas** variable is the map's HTML element.*
*The **mapOptions** variable defines the properties for the map.*
*The **center** property specifies where to center the map (using latitude and longitude coordinates).*
*The **zoom** property specifies the zoom level for the map (try to experiment with the zoom level).*
*The **google.maps.Map** object is created with mapCanvas and mapOptions as parameters.*

- https://www.w3schools.com/graphics/google_maps_intro.asp

- 900 fonts available
- https://www.w3schools.com/howto/tryit.asp?font=Sofia
- https://www.w3schools.com/whatis/tryit.asp?font=Anton



- https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis_google_chart_pie

XML plays an important role in many different IT systems
XML is often used for distributing data over the Internet
It is important for all web developers to have a good understanding of XML

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

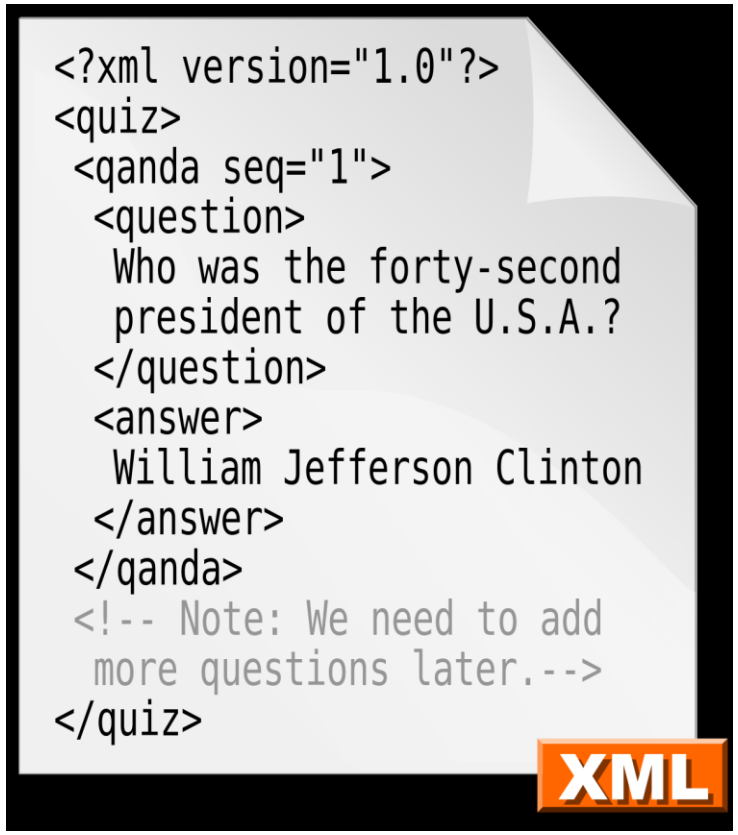
To: Tove
From: Jani
Heading: Reminder
Body: Don't forget me this weekend!

XML:

```
<breakfast_menu>
  <food>
    <name>Belgian Waffles</name>
    <price>$5.95</price>
    <description>Two of our famous Belgian Waffles with plenty of real maple syrup</description>
    <calories>650</calories> </food>
  <food>
    <name>Strawberry Belgian Waffles</name>
    <price>$7.95</price>
    <description>Light Belgian waffles covered with strawberries and whipped cream</description>
    <calories>900</calories> </food>
  <food>
    <name>Berry-Berry Belgian Waffles</name>
    <price>$8.95</price>
    <description>Light Belgian waffles covered with an assortment of fresh berries and whipped
cream</description>
    <calories>900</calories> </food>
  <food>
    <name>French Toast</name>
    <price>$4.50</price>
    <description>Thick slices made from our homemade sourdough bread</description>
    <calories>600</calories> </food>
  <food>
    <name>Homestyle Breakfast</name>
    <price>$6.95</price>
    <description>Two eggs, bacon or sausage, toast, and our ever-popular hash browns</description>
    <calories>950</calories> </food>
</breakfast_menu>
```

Xslt:

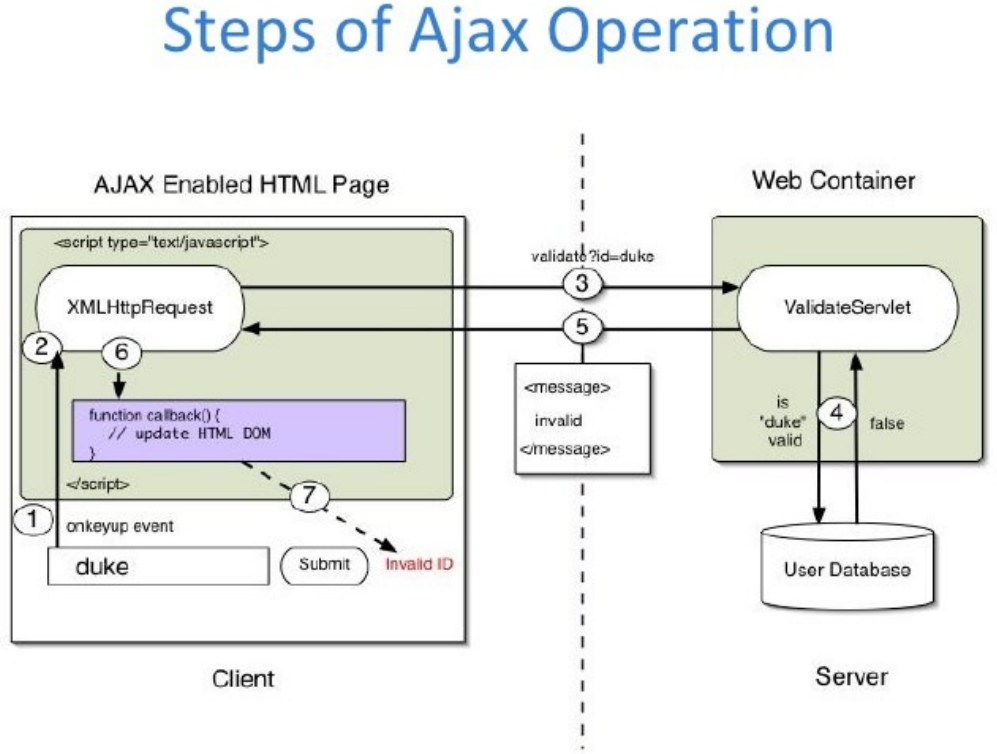
https://www.w3schools.com/whatis/simplexsl.xml



https://www.w3schools.com/xml/default.asp

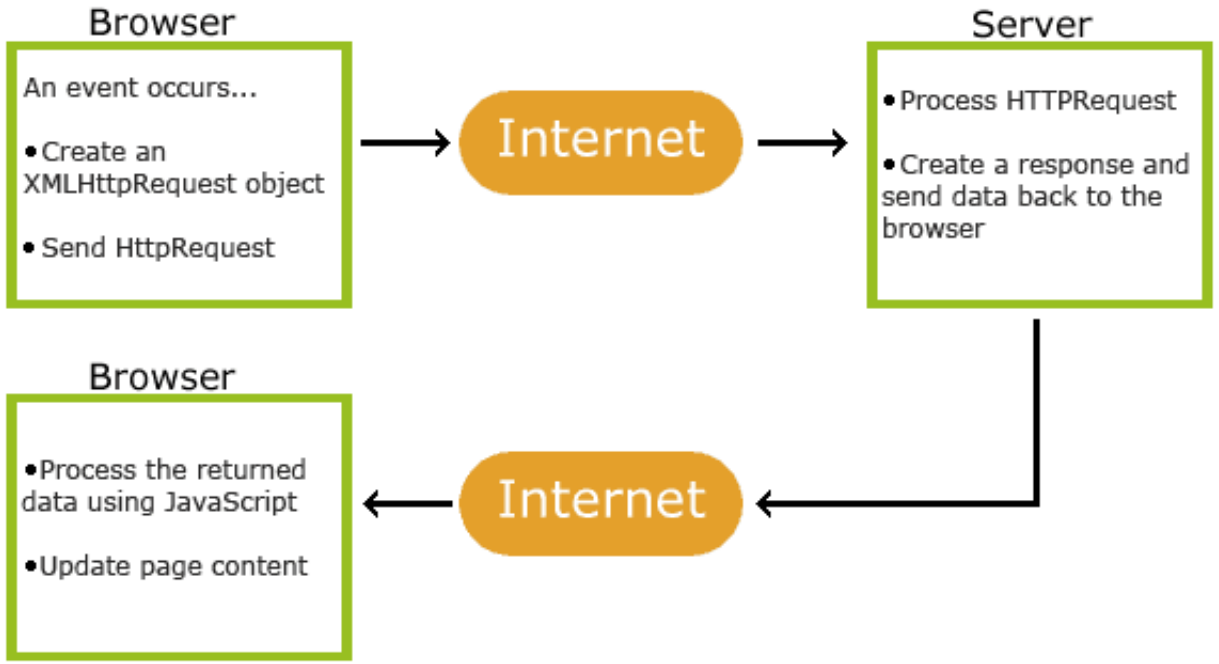
AJAX is a developer's dream, because you can:

- Read data from a web server - after a web page has loaded
- Update a web page without reloading the page
- Send data to a web server - in the background



https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis_ajax

The HTML page contains a section and a <button>.
The section is used to display information from a server.
The <button> calls a function (if it is clicked).
The function requests data from a web server and displays it:

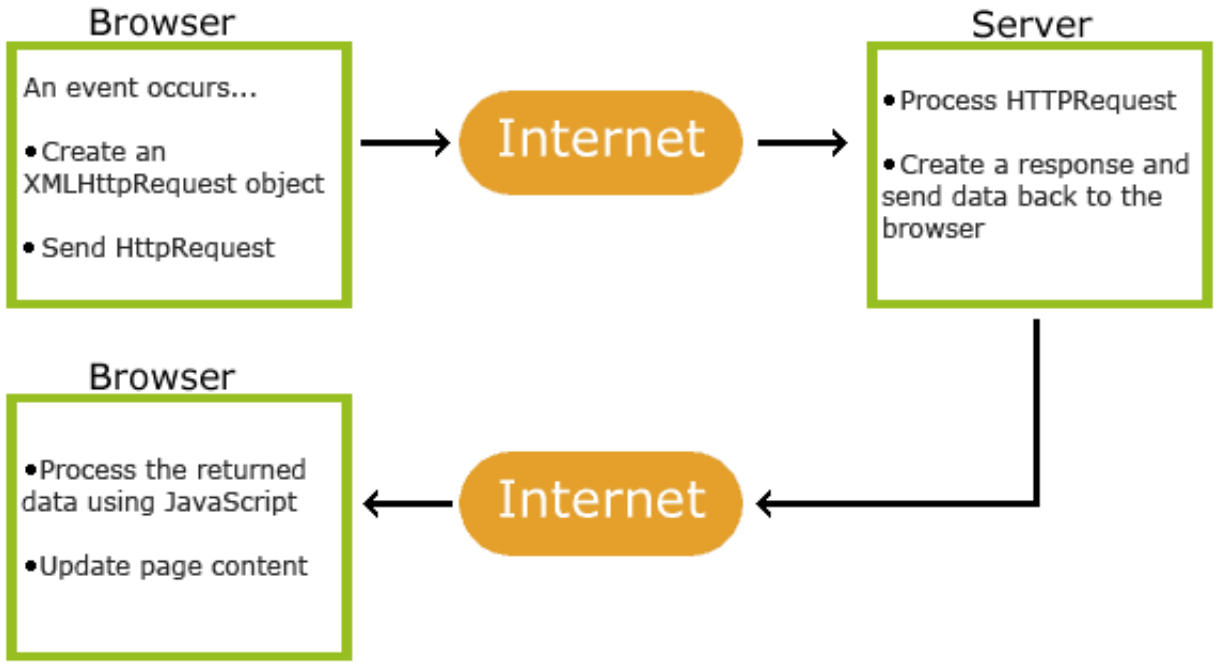


AJAX = Asynchronous JavaScript And XML.
AJAX is not a programming language.
AJAX just uses a combination of:

- A browser built-in XMLHttpRequest object (to request data from a web server)
- JavaScript and HTML DOM (to display or use the data)

AJAX is a misleading name. AJAX applications might use XML to transport data, but it is equally common to transport data as plain text or JSON text.

AJAX allows web pages to be updated asynchronously by exchanging data with a web server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.



An event occurs in a web page (the page is loaded, a button is clicked)
2. An XMLHttpRequest object is created by JavaScript
3. The XMLHttpRequest object sends a request to a web server
4. The server processes the request
5. The server sends a response back to the web page
6. The response is read by JavaScript
7. Proper action (like page update) is performed by JavaScript

For a full AJAX tutorial go to W3Schools AJAX Tutorial.
https://www.w3schools.com/js/js_ajax_intro.asp

JSON stands for JavaScript Object Notation
JSON is a lightweight format for storing and transporting data
JSON is often used when data is sent from a server to a web page
JSON is "self-describing" and easy to understand

This example defines an employees object: an array of 3 employee records (objects):

```
{
"employees":[
    {"firstName":"John", "lastName":"Doe"},
    {"firstName":"Anna", "lastName":"Smith"},
    {"firstName":"Peter", "lastName":"Jones"}
]
}
```

{JSON}

JavaScript Object Notation

```
{                                                   Object Starts
    "Title": "The Cuckoo's Calling"
    "Author": "Robert Galbraith",
    "Genre": "classic crime novel",
    "Detail": {                                     Object Starts
        "Publisher": "Little Brown"                 Value string
        "Publication_Year": 2013,                   Value number
        "ISBN-13": 9781408704004,
        "Language": "English",
        "Pages": 494
    }                                               Object ends
    "Price": [                                      Array starts
        {                                           Object Starts
            "type": "Hardcover",
            "price": 16.65,
        }                                           Object ends
        {                                           Object Starts
            "type": "Kindle Edition",
            "price": 7.03,
        }                                           Object ends
    ]                                               Array ends
}                                                   Object ends
```

JSON Syntax Rules

- Data is in name/value pairs
- Data is separated by commas
- Curly braces hold objects
- Square brackets hold arrays

JavaScript Object Notation

The JSON format is syntactically identical to the code for creating JavaScript objects.
Because of this similarity, a JavaScript program can easily convert JSON data into native JavaScript objects.
The JSON syntax is derived from JavaScript object notation syntax, but the JSON format is text only.
Code for reading and generating JSON data can be written in any programming language.

- **JSON data** is written as name/value pairs, just like JavaScript object properties.

A name/value pair consists of a field name (in double quotes), followed by a colon, followed by a value:
"firstName":"John“

JSON names require double quotes. JavaScript names do not.

- **JSON objects** are written inside curly braces.

Just like in JavaScript, objects can contain multiple name/value pairs:
{"firstName":"John", "lastName":"Doe"}

- **JSON arrays** are written inside square brackets. Just like in JavaScript, an array can contain objects:

```
"employees":[
   {"firstName":"John", "lastName":"Doe"},
   {"firstName":"Anna", "lastName":"Smith"},
   {"firstName":"Peter", "lastName":"Jones"}
]
```

In the example above, the object "employees" is an array. It contains three objects.
Each object is a record of a person (with a first name and a last name).

Convert JSON

Esempio

- A common use of JSON is to read data from a web server, and display the data in a web page.
- For simplicity, this can be demonstrated using a string as input.
- First, create a JavaScript string containing JSON syntax:

```
var text = '{ "employees" : [' +
'{ "firstName":"John" , "lastName":"Doe" },' +
'{ "firstName":"Anna" , "lastName":"Smith" },' +
'{ "firstName":"Peter" , "lastName":"Jones" } ]}';
```

- Then, use the JavaScript built-in function JSON.parse() to convert the string into a JavaScript object:

```
var obj = JSON.parse(text);
```

- Finally, use the new JavaScript object in your page:

```
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML =
obj.employees[1].firstName + " " + obj.employees[1].lastName;
</script>
```

CSS Icon

Esempio

Icons come in scalable vector libraries that can be customized with CSS
Common libraries are:

- Font Awesome Icons
- Bootstrap Icons
- Google Icons

To use icons, just add a link to the icon library in the <head> section of your HTML page:
No downloads or installations required!
To insert an icon, add the name of the icon class to any inline HTML element like <i> or .

Esempio Icons

Examples:

- Bootstrap Icon
- Google Icon

Css Icon
icons_reference

Bootstrap Framework

- Bootstrap is is the most popular CSS Framework for developing responsive and mobile-first websites.
- Bootstrap 4 is the newest version of Bootstrap

- Esempio

Bootstrap 4 supports all major browsers except Internet Explorer 9.
If you require support for IE9 or IE8, you must use Bootstrap 3.

The container class is one of the most important Bootstrap classes.
It provides margins, padding, alignments, and more, to HTML elements.
Example:

```
<div class="container">
   <h1>This is a paragraph</h1>
   <p>This is a paragraph</p>
   <p>This is a paragraph</p>
   <p>This is a paragraph</p>
   <p>This is a paragraph</p>
</div>
```

Esempio container

Bootstrap Framework

- Colors
- https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis_bootstrap_colors

- Text colors
- https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis_bootstrap_txt_colors&stacked=h

- Three equal-width columns, on all devices and screen widths:
- https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis_bootstrap_grid_fixed&stacked=h

- Responsive columns. Three equal-width columns scaling to stack on top of each other on small screens:
- https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis_bootstrap_grid_responsive&stacked=h

- A boredered zebra-striped table:
- https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis_bootstrap_table&stacked=h

- Bootstrap provides an easy way to create predefined alert messages:
- https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis_bootstrap_alerts&stacked=h

Bootstrap Framework

- Bootstrap provides different styles of buttons:
- https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis_bootstrap_buttons&stacked=h

- Cards
- https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis_bootstrap_cards&stacked=h

For a full Bootstrap tutorial go to W3Schools Bootstrap Tutorial.
For a full Bootstrap reference go to W3Schools Bootstrap Reference.

- Esempi dal sito ufficiale bootstrap
- https://getbootstrap.com/docs/4.1/layout/overview/

W3.css

- Modern Responsive CSS
- Equality for all browsers: Chrome. Firefox Edge. IE. Safari. Opera.
- Equality for all devices: Desktop. Laptop. Tablet. Mobile.
- Standard CSS only (No jQuery or JavaScript library).

W3.CSS is a modern CSS framework with built-in responsiveness. It supports responsive mobile first design by default, and it is smaller and faster than similar CSS frameworks.
W3.CSS can also speed up and simplify web development because it is easier to learn, and easier to use than other CSS frameworks.

- Esempio con W3.css:
- https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis_w3css&stacked=h
- Esempio con bs
- https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis_bootstrap&stacked=h

W3.css

- The **w3-container** class is one of the most important W3.CSS classes.

It provides correct margins, padding, alignments, and more, to most HTML elements.
https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis_w3css_container

The **w3-color** classes are inspired by modern colors:
https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis_w3css_colors

The **w3-panel** class can display all kinds of **allerts** and **notes** and **quotes**:
https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis_w3css_alerts
https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis_w3css_panels

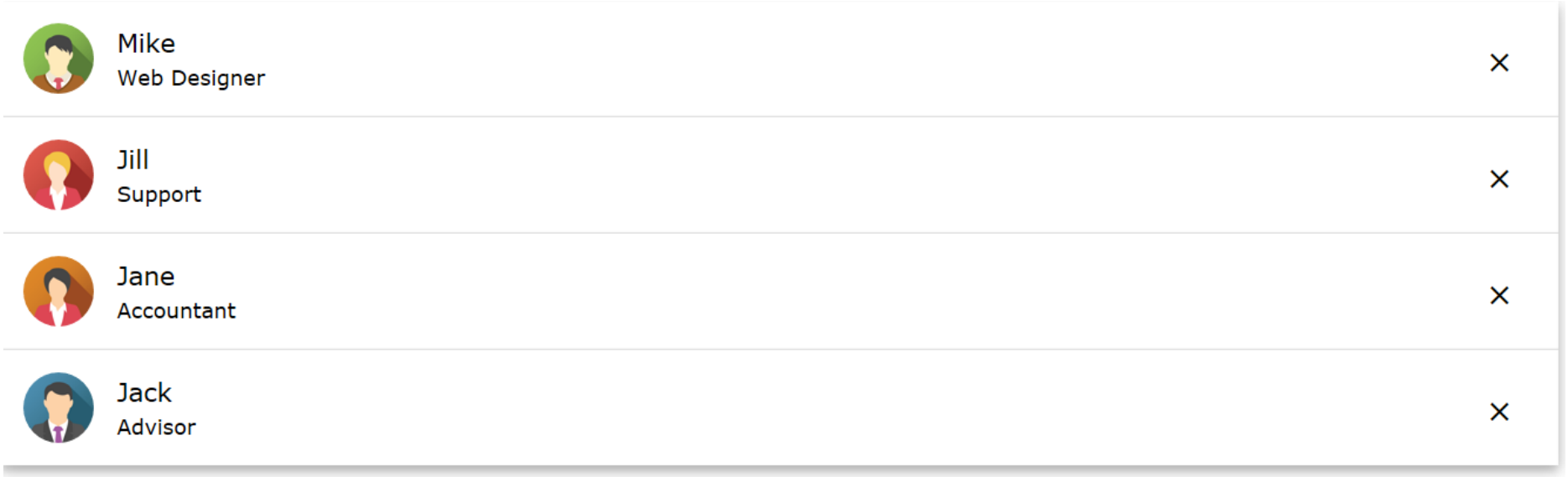
The **w3-card** classes are suitible for both images and notes:
https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis_w3css_cards

The **w3-table** classes can handle all kinds of tables:
https://www.w3schools.com/whatis/tryit.asp?filename=trywhatis_w3css_tables

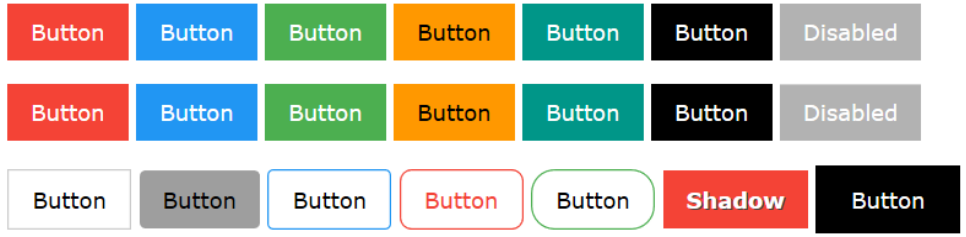
W3.css

- The **w3-ul** class can handle all kinds of **lists**:

- https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_lists_border
- https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_lists_red
- https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_lists_hoverable2
- https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_lists_avatar



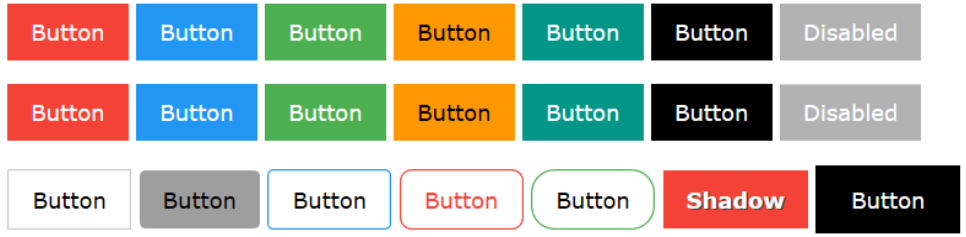
W3.css



- The **w3-button** and **w3-btn** class provides buttons of all sizes and types:
- W3.CSS provides the following classes for buttons:

Class	Defines
w3-btn	A rectangular button with a shadow hover effect. Default color is black.
w3-button	A rectangular button with a gray hover effect. Default color is light-gray in W3.CSS version 3. Default color is inherited from parent element in version 4.
w3-bar	A horizontal bar that can be used to group buttons together. (Perfect for horizontal navigation menus)
w3-block	Class that can be used to define a full width (100%) button.
w3-circle	Can be used to define a circular button.
w3-ripple	Can be used to create a ripple effect.

W3.css



- The **w3-button** and **w3-btn** class provides buttons of all sizes and types:
- W3.CSS provides the following classes for buttons:

Class	Defines
w3-btn	A rectangular button with a shadow hover effect. Default color is black.
w3-button	A rectangular button with a gray hover effect. Default color is light-gray in W3.CSS version 3. Default color is inherited from parent element in version 4.
w3-bar	A horizontal bar that can be used to group buttons together. (Perfect for horizontal navigation menus)
w3-block	Class that can be used to define a full width (100%) button.
w3-circle	Can be used to define a circular button.
w3-ripple	Can be used to create a ripple effect.

W3.css - Bottons

Both the **w3-button** class and the **w3-btn** class add button-behavior to any HTML elements.
The most common elements to use are <input type="button">, <button>, and <a>:
Buttons all

Hover effects also come in all colors. The **w3-hover-*color*** classes is used to add hover color to buttons:
Buttons hoover

https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_buttons_shapes
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_buttons_sizes
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_buttons_borders
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_buttons_text_effects3

The size of the a block can be defined using **style="width:"**.
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_buttons_width_size

W3.css Tags

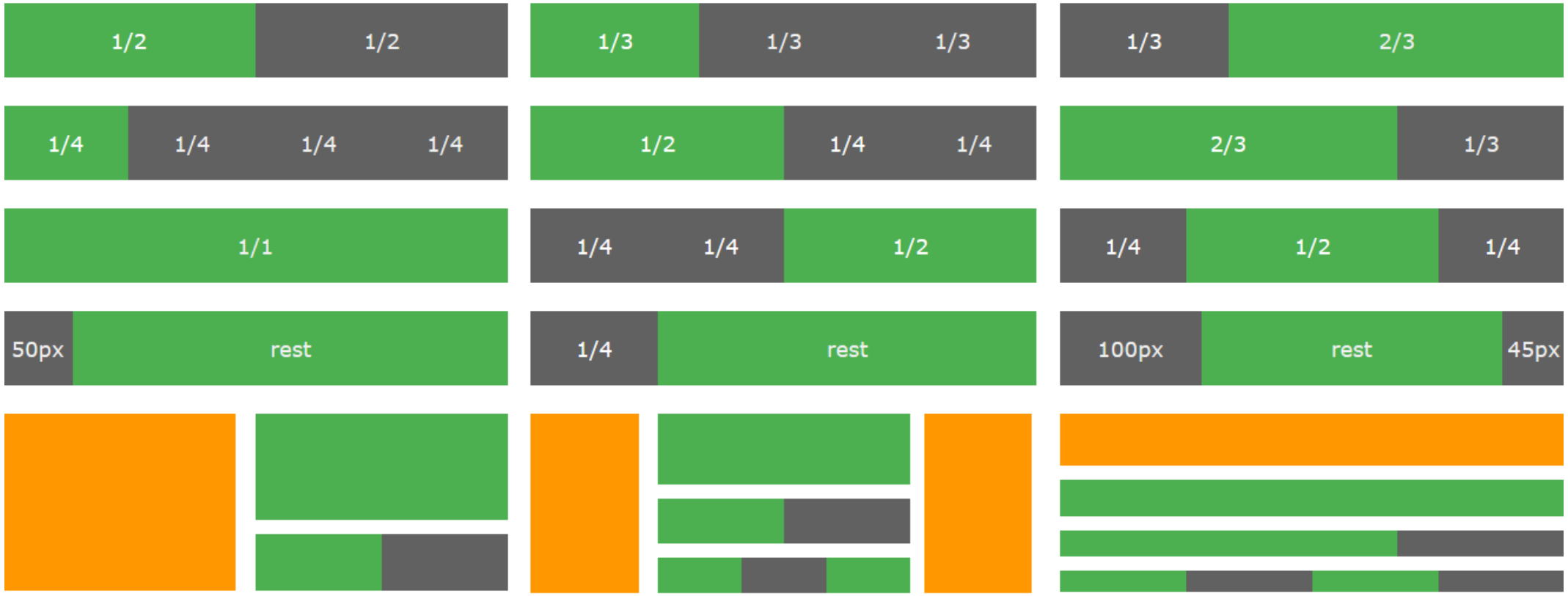
The **w3-tag** and the **w3-badge** classes are capable of displaying all kinds of tags, labels, badges and signs:
Use a **w3-*color*** class to change the color of a tag:
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_tags_color

By default, a tag will inherit the size of its container.
The **w3-*size*** classes (w3-tiny, w3-small, w3-large, w3-xlarge, w3-xxlarge, w3-xxxlarge, w3-jumbo) can be used to modify the size of a tag:
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_tags_large

The **w3-spin** class can be used to let a sign spin 360 degrees:
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_tags_spin

W3.css Grids

The **Responsive Grid** classes provide responsiveness for all device types: PC, laptop, tablet, and mobile.
W3.CSS also supports a **12 column mobile-first fluid grid** with small, medium, and large classes



W3.css Grids

The **Responsive Grid** classes provide responsiveness for all device types: PC, laptop, tablet, and mobile.
W3.CSS also supports a **12 column mobile-first fluid grid** with small, medium, and large classes

W3.CSS's grid system is responsive, and the columns will re-arrange automatically depending on the screen size:

Class	Description
w3-half	Occupies 1/2 of the window (on medium and large screens)
w3-third	Occupies 1/3 of the window (on medium and large screens)
w3-twothird	Occupies 2/3 of the window (on medium and large screens)
w3-quarter	Occupies 1/4 of the window (on medium and large screens)
w3-threequarter	Occupies 3/4 of the window (on medium and large screens)
w3-rest	Occupies the rest of the column width
w3-col	Defines one column in a 12-column responsive grid
w3-mobile	Adds mobile-first responsiveness to a cell (column). Displays elements as block elements on mobile devices.

W3.css Grids

The responsive classes grid must be placed inside a **w3-row** class (or **w3-row-padding** class) to be fully responsive.

Class	Description
w3-row	Container for responsive classes, with no padding
w3-row-padding	Container for responsive classes, with 8px left and right padding
w3-content	Container for fixed size centered content
w3-hide-small	Hides content on small screens (less than 601px)
w3-hide-medium	Hides content on medium screens
w3-hide-large	Hides content on large screens (larger than 992px)
l1 - l12	Responsive sizes for large screens
m1 - m12	Responsive sizes for medium screens
s1 - s12	Responsive sizes for small screens

W3.css Grids

The width of the **w3-half** class is 1/2 of the parent element (style="width:50%").
On screens smaller than 601 pixels it resizes to 100%.

https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_responsive_half&stacked=h

The **w3-col** class defines one column in a 12-column responsive grid.
The **w3-rest** class will occupy the rest of the width:
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_grid_rest&stacked=h

he **w3-content** class defines a container for fixed size centered content. Use the CSS max-width property to override the default width (980px).
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_responsive_content&stacked=h

The **w3-hide-small**, **w3-hide-medium**, and **w3-hide-large** classes hide elements on specific screen sizes.
Note: Resize the browser window to understand how it works:
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_responsive_hide&stacked=h

W3.css Mobile

The **w3-mobile** class adds mobile-first responsiveness to any element.
It adds display:block and width:100% to an element on screens that are less than 600px wide.
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_w3-mobile

The built-in responsiveness of W3.CSS uses the DP size of a screen.
W3.CSS will treat an iPhone 6 with a resolution of 750 x 1334 pixels as a small screen of 375 x 667 pixels DP.
Small screens are less than 601 pixels DP, medium screens are less than 993 pixels DP.

W3.css Layout

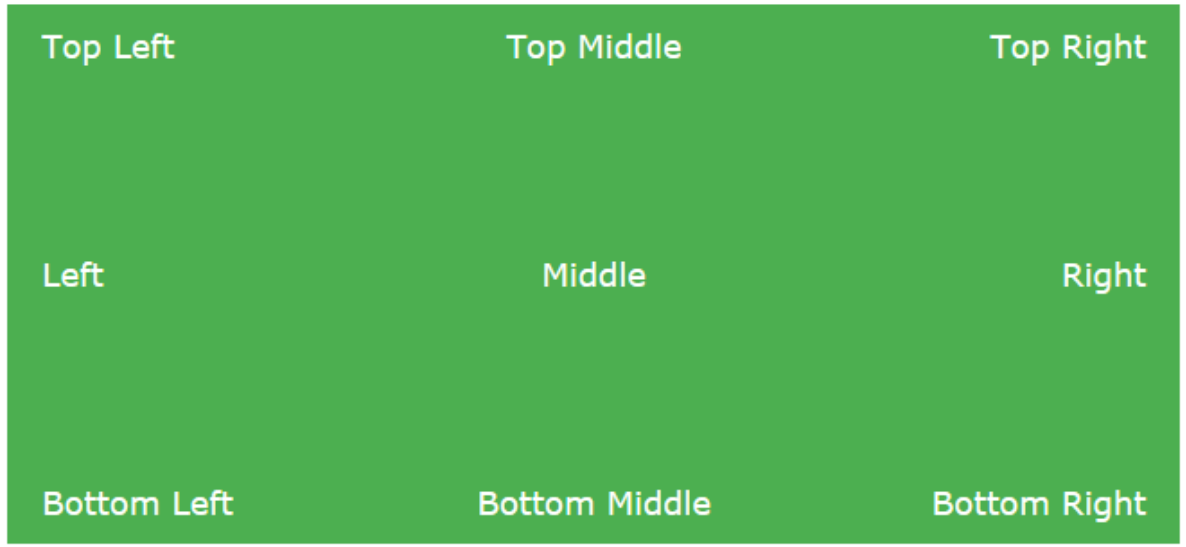
The **w3-cell** class redefines block elements to display horizontally (like table cells):
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_layout_cell

The **w3-cell-row** is a container for cells (columns).
The width of the w3-cell-row container defines the total width of all the contained cells.
The default width is 100%:
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_layout_container

he **w3-cell** class has a very nice built-in self adjusting standard. Side-by-side elements will automatically adjust to the necessary width:
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_layout_cell_width
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_layout_col

W3.css Display

The **w3-display** classes allow you to display HTML elements in specific positions:



Class	Defines
w3-display-container	Container for w3-display-*classes*
w3-display-topleft	Displays content at the top left corner of the w3-display-container
w3-display-topright	Displays content at the top right corner of the w3-display-container
w3-display-bottomleft	Displays content at the bottom left corner of the w3-display-container
w3-display-bottomright	Displays content at the bottom right corner of the w3-display-container
w3-display-left	Displays content to the left (middle left) of the w3-display-container
w3-display-right	Displays content to the right (middle right) of the w3-display-container
w3-display-middle	Displays content in the middle (center) of the w3-display-container
w3-display-topmiddle	Displays content at the top middle of the w3-display-container
w3-display-bottommiddle	Displays content at the bottom middle of the w3-display-container
w3-display-position	Displays content at a specified position in the w3-display-container
w3-display-hover	Displays content on hover inside the w3-display-container
w3-left	Floats an element to the left (float: left)
w3-right	Floats an element to the right (float: right)
w3-show	Shows an element (display: block)
w3-hide	Hides an element (display: none)
w3-mobile	Adds mobile-first responsiveness to any element. Displays elements as block elements on mobile devices

Esempio div
Esempio padded
Esempio img
esmpio effetto hoover
Esempio animation

W3.css Display

Force an element to be shown or hidden with the **w3-show** or **w3-hide** class.
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_display_show
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_display_show_toggle

W3.css Animation

W3.CSS provides the following classes for animations:

Class	Defines
w3-animate-top	Slides in an element from the top (-300px to 0)
w3-animate-bottom	Slides in an element from the bottom (-300px to 0)
w3-animate-left	Slides in an element from the left (-300px to 0)
w3-animate-right	Slides in an element from the right (-300px to 0)
w3-animate-opacity	Animates an element's opacity from 0 to 1 in 1.5 seconds
w3-animate-zoom	Animates an element from 0 to 100% in size
w3-animate-fading	Animates an element's opacity from 0 to 1 and 1 to 0 (fades in + fade out)
w3-spin	Spins an element 360 degrees

W3.css Animation

- The **w3-animate-top** class slides in an element from the top (from -300px to 0):

https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_animate-top

- The **w3-animate-opacity** class animates an element's opacity from 0 to 1 in 0.8 seconds.

Fade in an element with the **w3-animate-opacity** class:
Example: ..
Try It Yourse

- The **w3-animate-zoom** class animates an element from 0 to 100% in size.

Zoom in an element with the **w3-animate-zoom** class:
Example: ..
Try It You

The **w3-spin** class spins an element 360 degrees:
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_animate-spin

W3.css Animation

The **w3-animate-fading** class fades in and out an element every 10 seconds (continuously):
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_animate_fading

W3.css Modals

The **w3-modal** class provides modal dialog in pure HTML:
The **w3-modal** class defines a container for a modal.
The **w3-modal-content** class defines the modal content.
Modal content can be any HTML element (divs, headings, paragraphs, images, etc.).

https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_modal

Use **w3-container** classes to create different sections inside the modal content:
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_modal2

W3.css Progress bar

A progress bar can be used to show how far along a user is in a process:
A normal element can be used for a progress bar.
The CSS width property can be used to set the height and width of a progress bar.
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_progressbar
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_progressbar_labels

Use JavaScript to create a dynamic progress bar:
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_progressbar_js
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_progressbar_labels_js
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_progressbar_labels_js3
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_progressbar_labels_js4

W3.css - DropDown

W3.CSS provides the following dropdown classes:

Class	Defines
w3-dropdown-hover	A hoverable dropdown element
w3-dropdown-content	The dropdown part to be displayed
w3-dropdown-click	A clickable dropdown element

The **w3-dropdown-hover** class defines a hoverable dropdown element.
The **w3-dropdown-content** class defines the dropdown content to be displayed.
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_dropdown_hover

Both the hoverable element and the dropdown content element can be any HTML element.
In the example above the hover element was a <button>, and the dropdown content a .
In the next example the hover element is a <p>, and the dropdown content is a :
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_dropdown_hover_p

W3.css DropDown

The **w3-dropdown-hover** class is perfect for creating navigation bars with dropdown menues:
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_navbar_dropdown

The **w3-dropdown-click** class creates a clickable dropdown element.
With this class, the dropdown is opened by JavaScript:
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_dropdown_click
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_dropdown_pic
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_dropdown_card
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_dropdown_animate

W3.css Accordion

An accordion is used to show (and hide) HTML content.
Use the **w3-hide** class to hide the accordion content.
Use any kind of button to open and close the content:

https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_accordion
Both the element that is used to open the accordion and the accordion's content can be any HTML element.
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_accordion_links
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_accordion_animate

Accordion	**Dropdown**
Content pushes the page content down	Content lays over existing the page content
Content is often 100% wide	Content is 160px wide (default)
Often used to open multiple sections	Often used to open one section

W3.css Tabs

Tabbed navigation is a way to navigate around a website.
Normally, tabbed navigation uses navigation buttons (tabs) arranged together with the selected tab highlighted.
This example uses elements with the same class name ("city" in our example) , and changes the style between **display:none** and **display:block** to hide and display different content:
Styles, buttos, and javascript function:
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_tabulators
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_tabulators_animate
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_tabulators_image

W3.css Navigation Bar

W3.CSS provides the following classes for navigation bars:

Class	Defines
w3-bar	Horizontal container for HTML elements
w3-bar-block	Vertical container for HTML elements
w3-bar-item	Container bar elements
w3-sidebar	Vertical sidebar for HTML elements
w3-mobile	Makes any bar element mobile-first responsive
w3-dropdown-hover	Hoverable dropdown element
w3-dropdown-click	Clickable dropdown element (instead of hover)

W3.css Navigation Bar

The w3-bar class is a container for displaying HTML elements horizontally.
The w3-bar-item class defines the container elements.
It is a perfect tool for creating navigation bars:

https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_navigation_bar

The **w3-mobile** class makes any bar elements responsive (horizontal on large screens and vertical on small).
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_navigation_bar_mobile

Add a ***w3-color*** class to a link to highlight it:
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_navbar_active

W3.css Slideshow

Displaying a manual slideshow with W3.CSS is very easy.
Just create many elements with the same class name, 2 buttons, and js function:

https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_slideshow_self

*First, set the **slideIndex** to 1. (First picture)*
*Then call **showDivs()** to display the first image.*
*When the user clicks one of the buttons call **plusDivs()**.*
*The plusDivs() function **subtracts** one or **adds** one to the slideIndex.*
*The **showDiv()** function hides (**display="none"**) all elements with the class name "mySlides", and displays (**display="block"**) the element with the given slideIndex.*
*If the slideIndex is **higher than** the number of elements (x.length), the slideIndex is set to zero.*
*If the slideIndex is **less than** 1 it is set to number of elements (x.length).*

Automatic Slider
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_slideshow_rr

W3.css- Input Forms

https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_input_top
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_input_bottom
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_input_card
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_input_label_colored

https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_input_bordered

https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_input_rounded
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_input_border-none
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_input_colors
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_input_hoverable

https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_input_animate
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_input_check
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_input_radio

https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_input_select
https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_input_contact

W3.css- Input Forms

For a full W3.CSS tutorial go to W3Schools W3.CSS Tutorial.
For a full W3.CSS reference go to W3Schools W3.CSS Reference.

Command line Interface - CLI

CLI is a command line program that accepts text input to execute operating system functions.
In the 1960s, using only computer terminals, this was the only way to interact with computers.
In the 1970s an 1980s, command line input was commonly used by Unix systems and PC systems like MS-DOS and Apple DOS.
Today, with graphical user interfaces (GUI), most users never use command-line interfaces (CLI).
However, CLI is still used by software developers and system administrators to configure computers, install software, and access features that are not available in the graphical interface.

The software package manager npm uses command line input to install software:
Windows Example: C:\>npm install mysoftware
Mac OS Example: >npm install mysoftware
You can navigate your folders (directories) with command line commands:
Windows Example:

```
C:\Users\myuser>cd ..
C:\Users\>cd ..
C:\>
```

Command line Interface – Linux commands

Command	Description
ls	List the directory (folder) system.
cd *pathname*	Change directory (folder) in the file system.
cd ..	Move one level up (one folder) in the file system.
cp	Copy a file to another folder.
mv	Move a file to another folder.
mkdir	Creates a new directory (folder).
rmdir	Remove a directory (folder).
clear	Clears the CLI window.
exit	Closes the CLI window.
man *command*	Shows the manual for a given command.

Command line Interface – Windows commands

Command	Description
dir	List the directory (folder) system.
cd *pathname*	Change directory (folder) in the file system.
cd \	Move to the root folder of the file system.
cd ..	Move one level up (one folder) in the file system.
copy	Copy a file to another folder.
move	Move a file to another folder.
type *filename*	Type a file.
mkdir or md	Creates a new directory (folder).
rmdir or rd	Removes a directory (folder).
cls	Clears the CLI window.
exit	Closes the CLI window.
help *command*	Shows the manual for a given command.

NPM - Node Package Manager

- **npm** is the world's largest **Software Library** (Registry)
- **npm** is also a software **Package Manager** and **Installer**

- **npm** includes a **CLI** (Command Line Client) that can be used to download and install software:
- Windows Example
- `C:\>npm install`

- **npm** is installed with **Node.js**
- This means that you have to install Node.js to get npm installed on your computer
  https://nodejs.org

- All **npm** packages are defined in files called **package.json**.
- The content of package.json must be written in **JSON**.
- At least two fields must be present in the definition file: **name** and **version**

NPM - Node Package Manager

Example

```
{
"name" : "foo",
"version" : "1.2.3",
"description" : "A package for fooing things",
"main" : "foo.js",
"keywords" : ["foo", "fool", "foolish"],
"author" : "John Doe",
"licence" : "ISC"
}
```

- npm can manage dependencies.
- npm can (in one command line) install all the dependencies of a project.
- Dependencies are also defined in package.json.

NPM - Node Package Manager

- If you want to share your own software in the npm registry, you can sign in at: https://www.npmjs.com
- You can publish any directory from your computer as long as the directory has a package.json file.

- Check if npm is installed:

C:\>npm

- Check if you are logged in:

C:\>npm whoami

- If not, log in:

C:\>npm login
Username: <your username>
Password: <your password>

- Navigate to your project and publish your project:

C:\Users\myuser>cd myproject
C:\Users\myuser\myproject>npm publish

GitHub

- A GitHub repository can be used to store a development project.
- It can contain folders and any type of files (HTML, CSS, JavaScript, Documents, Data, Images).
- A GitHub repository should also include a licence file and a README file about the project.
- A GitHub repository can also be used to store ideas, or any resources that you want to share.
- **Branch**
- A GitHub branch is used to work with different versions of a repository at the same time.
- By default a repository has a master branch (a production branch).
- Any other branch is a copy of the master branch (as it was at a point in time).
- New Branches are for bug fixes and feature work separate from the master branch. When changes are ready, they can be merged into the master branch. If you make changes to the master branch while working on a new branch, these updates can be pulled in.

GitHub

- **Commits**

At GitHub, changes are called commits.

- Each commit (change) has a description explaining why a change was made.

- **Pull Requests**
- Pull Requests are the heart of GitHub collaboration.
- With a pull request you are proposing that your changes should be merged (pulled in) with the master.
- Pull requests show content differences, changes, additions, and subtractions in colors (green and red).
- As soon as you have a commit, you can open a pull request and start a discussion, even before the code is finished.
- A a great way to learn GitHub, before working on larger projects, is to open pull requests in your own repository and merge them yourself.
- You merge any changes into the master by clicking a "Merge pull request" button.