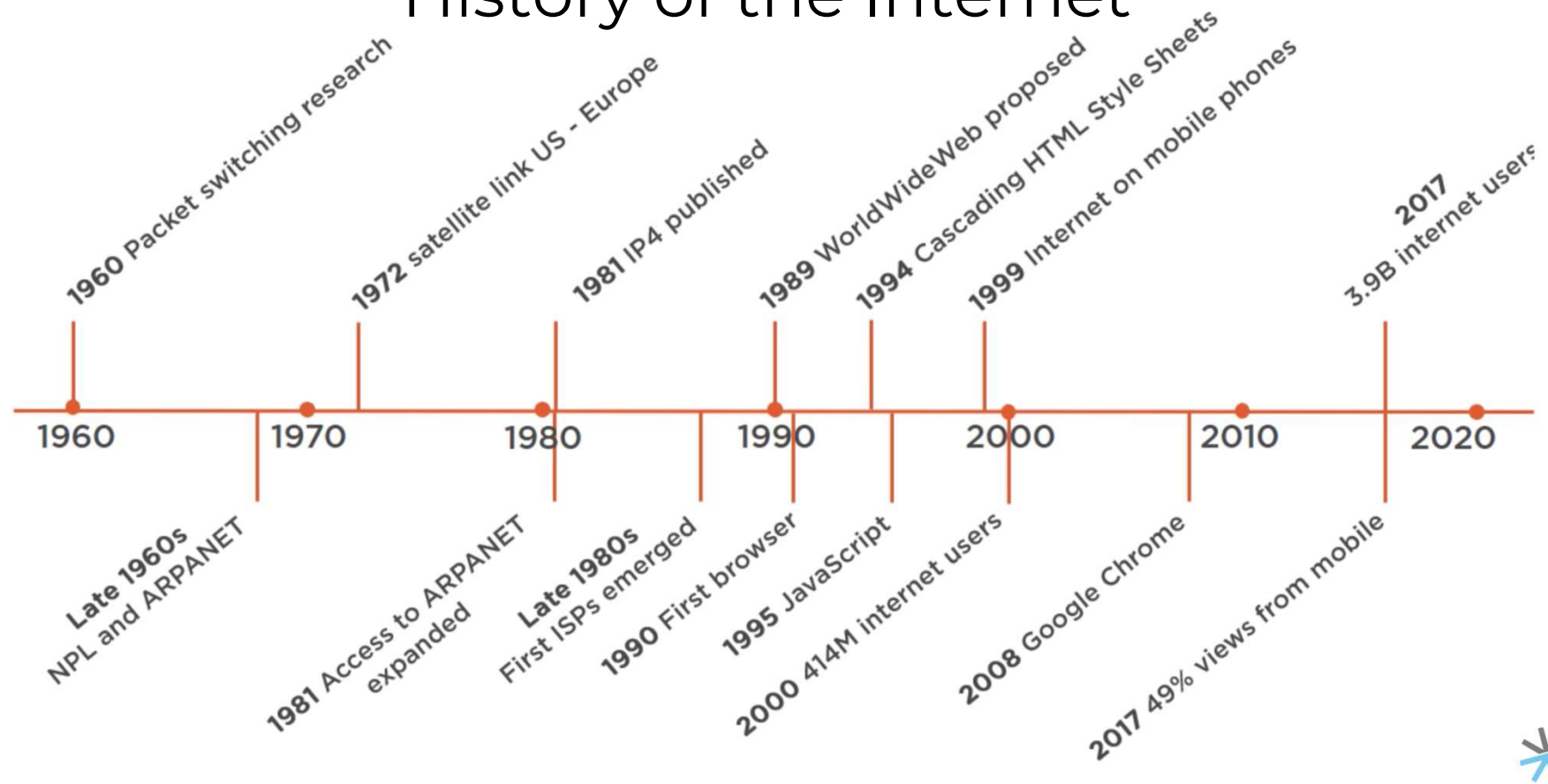


HTML, CSS, and JavaScript

The Big Picture



History of the Internet



What is the World Wide Web

The **World Wide Web**

is an **information space**

where **documents** and **other web resources**

are identified by **Uniform Resource Locators (URLs)**,

which may be interlinked by **hypertext**,

and are accessible over **the Internet**.

World Wide Web, Wikipedia, 2020
https://en.wikipedia.org/wiki/World_Wide_Web



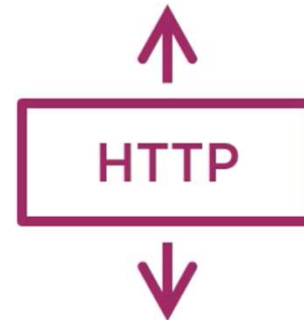
Ingredients for the web



Resources
(documents, images,
other files)



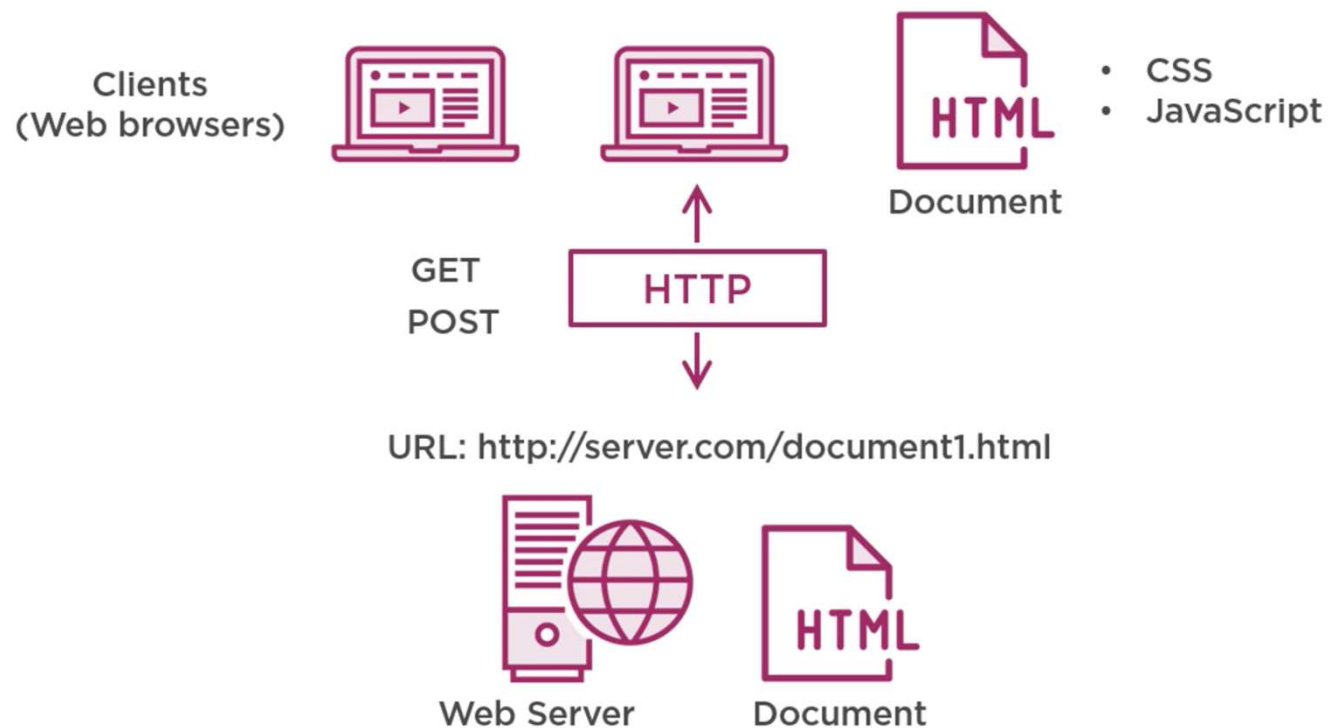
Uniform Resource
Locators (URLs)



Hypertext Transfer
Protocol (HTTP)



Ingredients for the web



Why Learn About HTML, CSS and Javascript

New to these topics



- What the technologies are
- Why they are relevant
- How people use them

Already experienced



- Reaffirm the concepts
- Put the technology in context



Thing to Remember



The web is possible because of the internet

- The internet has been around since the 80s
- The web has been around since the 90s

To use the web, you need:

- A web server, that hosts resources like documents
- URLs that identify those resources on that web server
- Protocols that can be used to transfer resources
- A web browser, to view documents and use resources

Why should you learn about HTML, CSS, and Javascript?

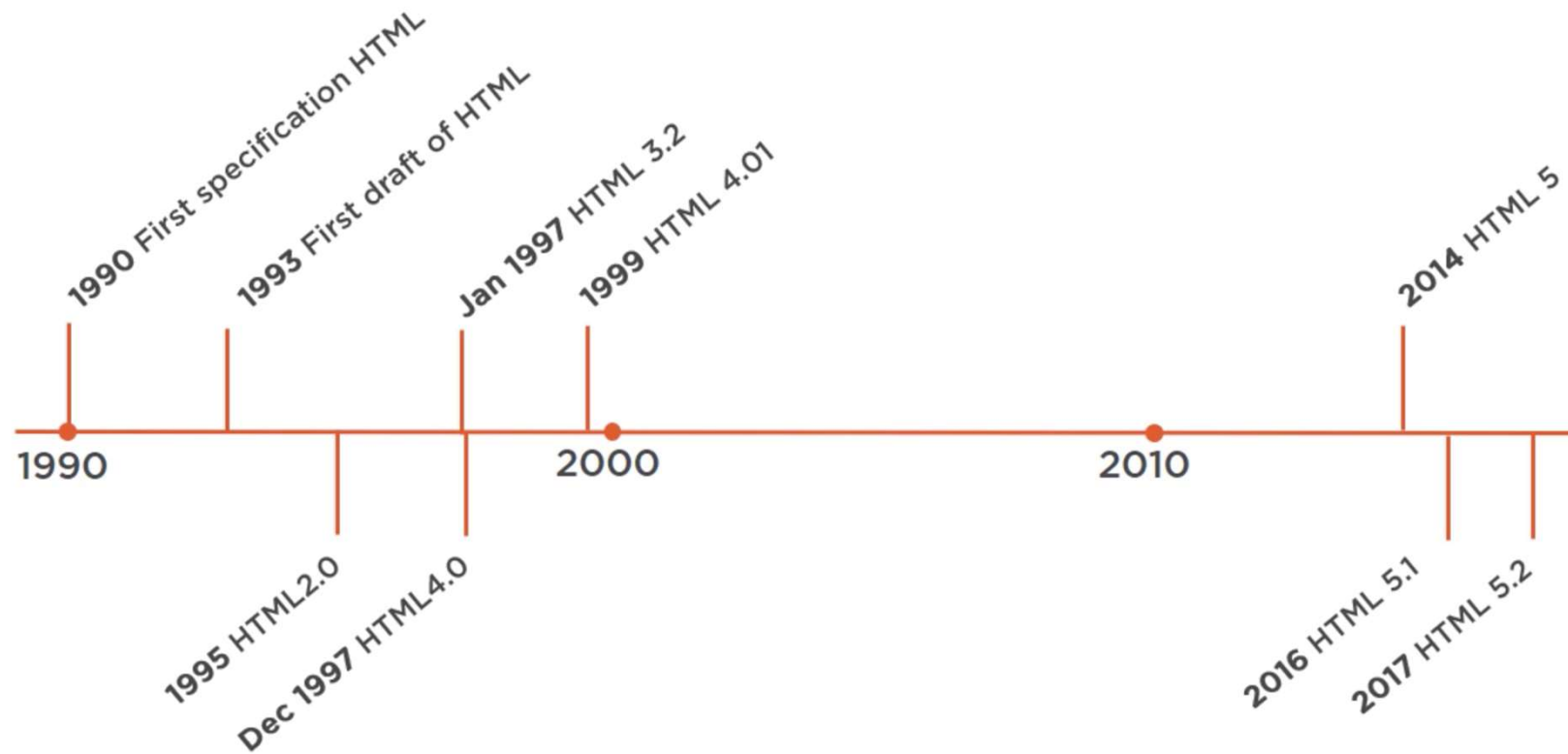
- **What** the technologies are, **Why** they are relevant and **HOW** people use them



Hypertext Markup Language (HTML) is the standard **markup** language for creating web pages and web applications



History of HTML



Who Manages the HTML Specification?

- 1990 Tim Berners-Lee
- 1993 Internet Engineering Task Force (IETF)
- 1996 World Wide Web Consortium (W3C)
- 2000 International Organization for Standardization (ISO)
International Electrotechnical Commission (IEC)
(ISO/IEC 15445:2000)
- 2004 Web Hypertext Application Technology Working Group
(WHATWG)



Structure of HTML Documents

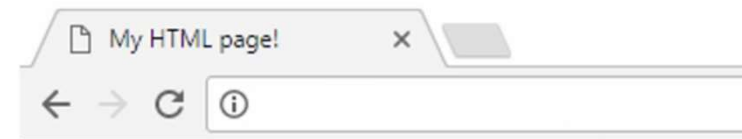
.HTM or .HTML

```
<!DOCTYPE html>
<html>
  <head>
    <!--meta information-->
  </head>
  <body>
    <!--the content of the page-->
  </body>
</html>
```



Example of HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My HTML page!</title>
  </head>
  <body>
    <div>
      <ul>
        <li>This</li>
        <li>is an</li>
        <li>unordered
          list</li>
      </ul>
    </div>
  </body>
</html>
```



- This
- is an
- unordered list



More Expressive HTML Elements

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My HTML page!</title>
  </head>
  <body>
    <header>
      <nav />
    </header>
    <main>
      <article />
      <aside />
    </main>
    <footer>
      <address />
    </footer>
  </body>
</html>
```



More Expressive HTML Elements

```
<audio controls>
  <source src="music.mp3" type="audio/mpeg">
  Your browser does not support the audio tag.
</audio>
```

```
<canvas id="myDrawing"></canvas>
```

```
<form action="/action_page.php" method="post">
  First name: <input type="text" name="fname"><br>
  Last name: <input type="text" name="lname"><br>
  <input type="submit" value="Submit">
</form>
```

```
<video width="150" height="200" controls>
  <source src="movie.mp4" type="video/mp4">
  Your browser does not support the video tag.
</video>
```



Browser Support of HTML Elements

<https://caniuse.com>
m

Video element  - LS

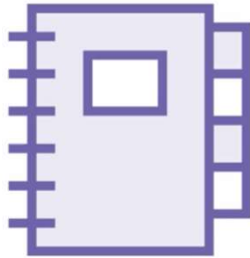
Usage % of all users ?
Global 97.24% + 0.12% = 97.36%

Method of playing videos on webpages (without requiring a plug-in). Includes support for the following media properties:
currentSrc, currentTime, paused, playbackRate, buffered,
duration, played, seekable, ended, autoplay, loop, controls, volume
& muted

Current aligned	Usage relative	Date relative	Apply filters	Show all	?													
IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Opera Mobile	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baidu Browser	KaiOS Browser		
		2-3		3.1-3.2														
6-8		3.5-19		4-10.1	10.1	3.2-10.3		2.1-2.2										
9-10	12-81	20-77	4-81	11-13	11.5-68	11-13.3		2.3-4.4.4	12-12.1				4-11.2					
11	83	78	83	13.1	69	13.5	all	81	46	81	68	12.12	12.0	10.4	7.12	2.5		
		79-80	84-86	14-TP		14.0												

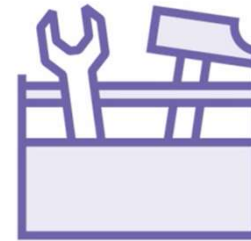


How to Write HTML from Scratch



Code editor

- Notepad++
- Atom
- Visual Studio Code

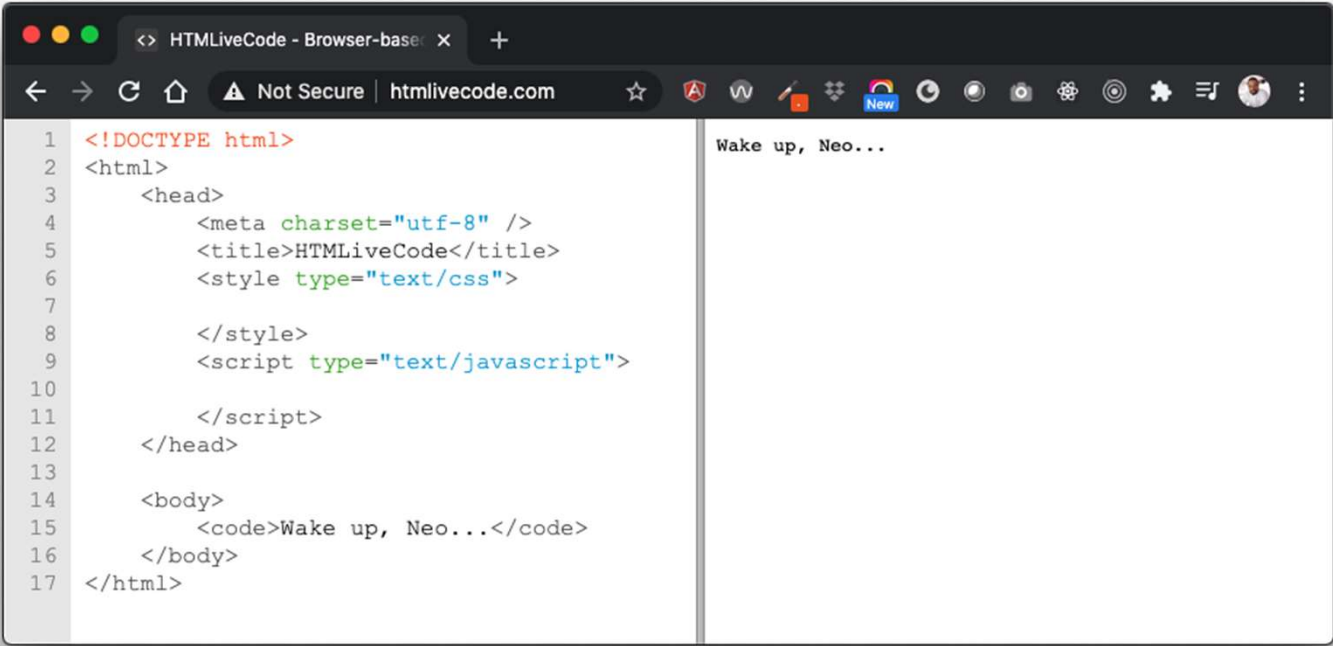


Integrated Development Environment (IDE)

- Eclipse
- Visual Studio
- Webstorm



How to Write HTML from Scratch



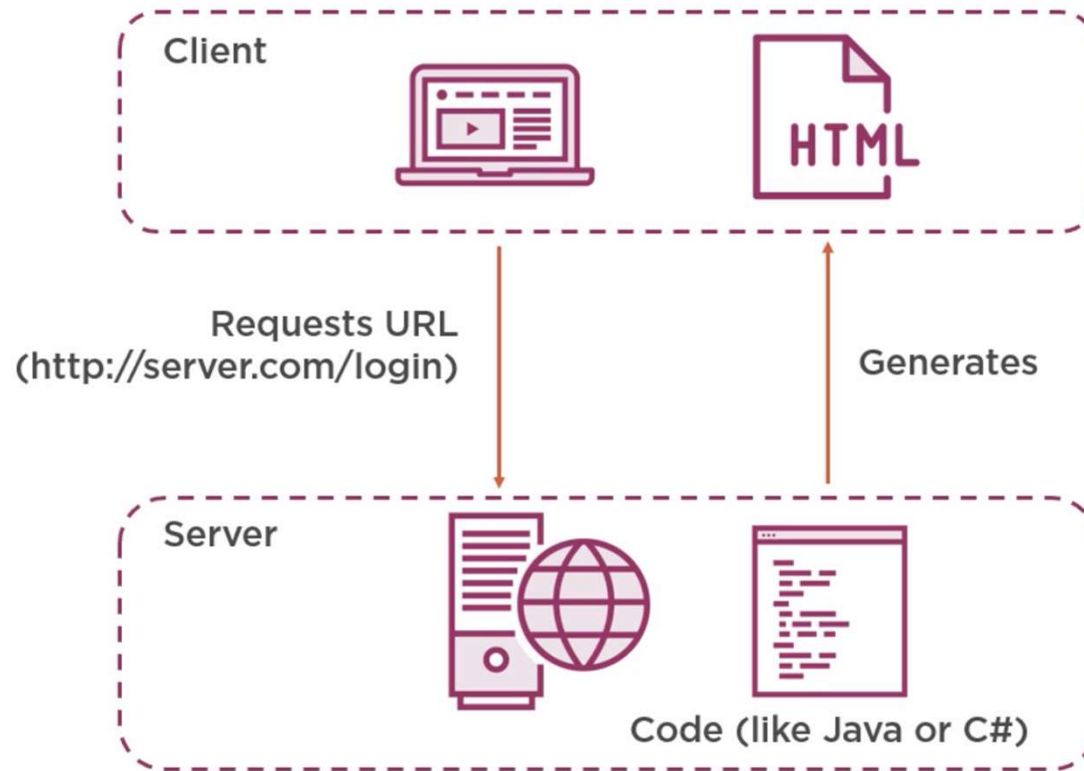
The screenshot shows a web browser window titled "HTMLLiveCode - Browser-based". The address bar shows "Not Secure | htmllivecode.com". The main content area is split into two panels. The left panel displays HTML code with line numbers 1 through 17. The right panel shows the rendered output of the code.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title>HTMLLiveCode</title>
6     <style type="text/css">
7
8     </style>
9     <script type="text/javascript">
10
11    </script>
12  </head>
13
14  <body>
15    <code>Wake up, Neo...</code>
16  </body>
17 </html>
```

The rendered output in the right panel is "Wake up, Neo...".



Write HTML from a Server



Lenguaje/Frameworks for Writing HTML

PHP

JAVA

ASP.NET MVC

Ruby

Python

Node.js

GO

Wordpress.com

Ghost.org



Thing to Remember



HTML is the basic markup for the web

It is a standard of elements and attributes

- Interpreted by web browsers

Developed by organizations and working groups

- IETF, W3C, ISO, IEC, WHATWG

Many elements and attributes

- Evolving towards more interactive elements

You can write HTML

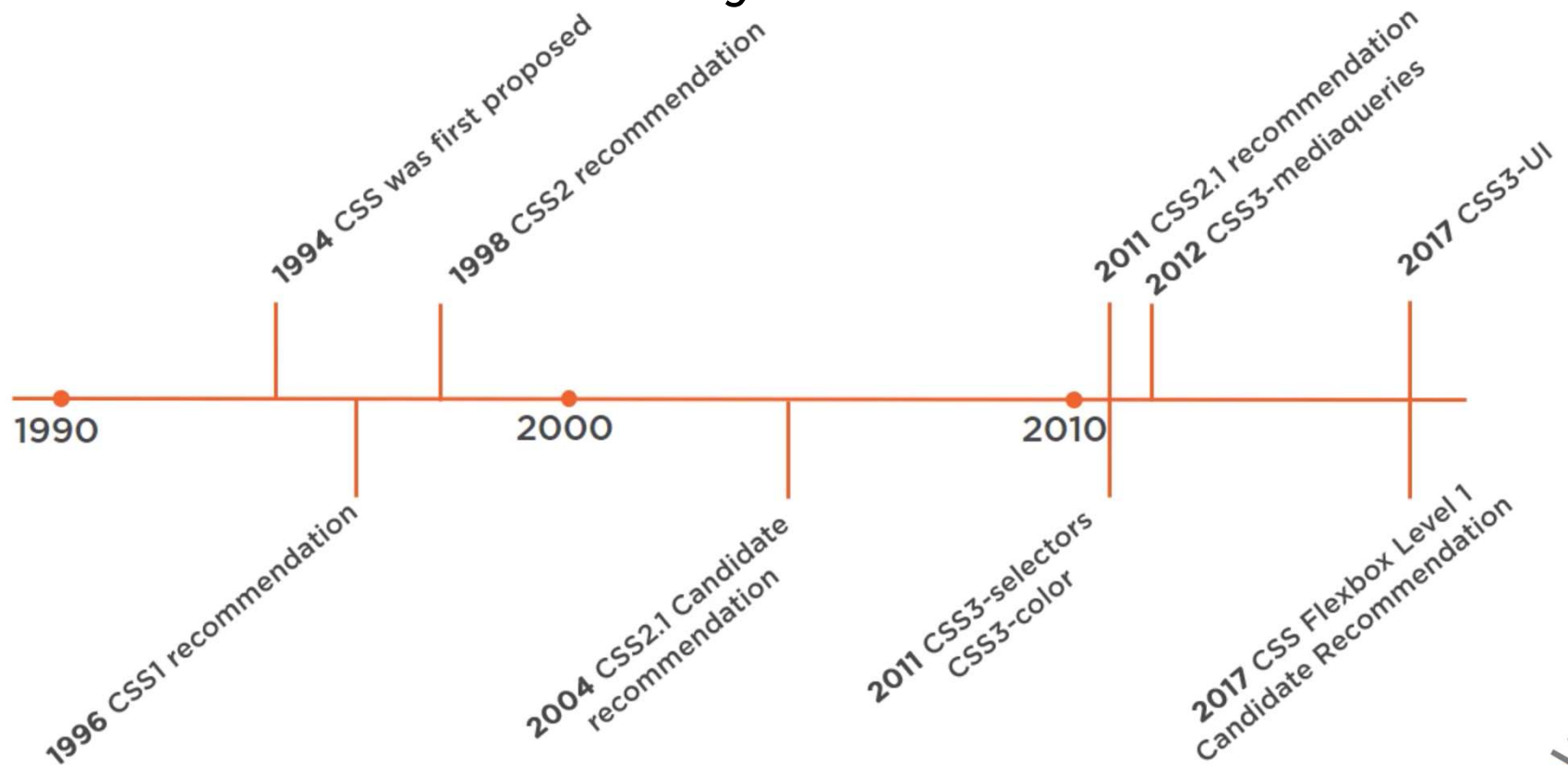
- From scratch
- With a framework that generates HTML



Cascading Style Sheets (CSS)
is a style sheet language used
for describing the
presentation of a document
written in a markup language



History of CSS

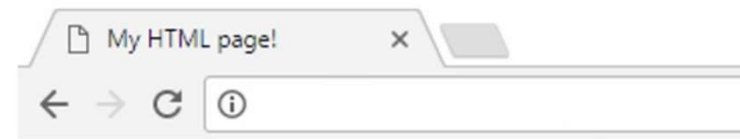


CSS
IS
AWESOME



HTML Styling

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My HTML page!</title>
  </head>
  <body>
    <div>
      <ul>
        <li>This</li>
        <li>is an</li>
        <li>unordered
          list</li>
      </ul>
    </div>
  </body>
</html>
```

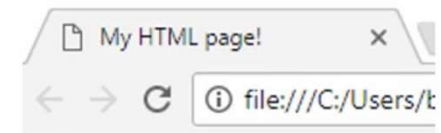


- This
- is an
- unordered list



HTML Styling

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My HTML page!</title>
  </head>
  <body>
    <div style="color:blue">
      <ul>
        <li>This</li>
        <li style="font-style:italic">is an</li>
        <li>unordered list</li>
      </ul>
    </div>
  </body>
</html>
```



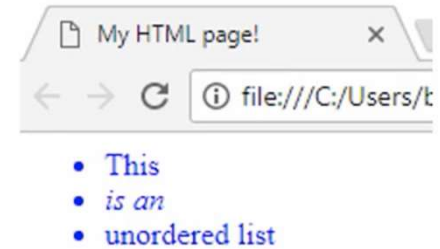
- This
- *is an*
- unordered list



```

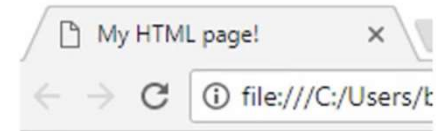
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My HTML page!</title>
    <style>
      div {
        color:blue
      }
      .italic {
        font-style:italic"
      }
    </style>
  </head>
  <body>
    <div>
      <ul>
        <li>This</li>
        <li class="italic">is an</li>
        <li>unordered list</li>
      </ul>
    </div>
  </body>
</html>

```



HTMLPage.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My HTML page!</title>
    <link rel="stylesheet" type="text/css" href="StyleSheet.css"
  </head>
  <body>
    <div>
      <ul>
        <li>This</li>
        <li class="italic">is an</li>
        <li>unordered list</li>
      </ul>
    </div>
  </body>
</html>
```



- This
- *is an*
- unordered list

StyleSheet.css

```
<style>
  div {
    color:blue
  }
  .italic {
    Font-style:italic"
  }
</style>
```



CSS Syntax Selectors

```
div {  
  color: blue;  
}
```

```
.class1 {  
  color: blue;  
}
```

```
#para1 {  
  color: blue;  
}
```

```
div p {  
  text-align: center;  
  color: red;  
}
```

```
h1, h2, p {  
  text-align: center;  
  color: red;  
}
```



CSS Priority

```
<p class="specific">Paragraph.</p>  
<p class="specific" id="morespecific">Paragraph with id!</p>
```

```
#morespecific {  
  background-color: red;  
  border: 1px solid black;  
}
```

```
.specific {  
  background-color: gray;  
  border: none !important;  
}
```

```
p {  
  background-color: blue;  
  color: white;  
  padding: 5px;  
}
```

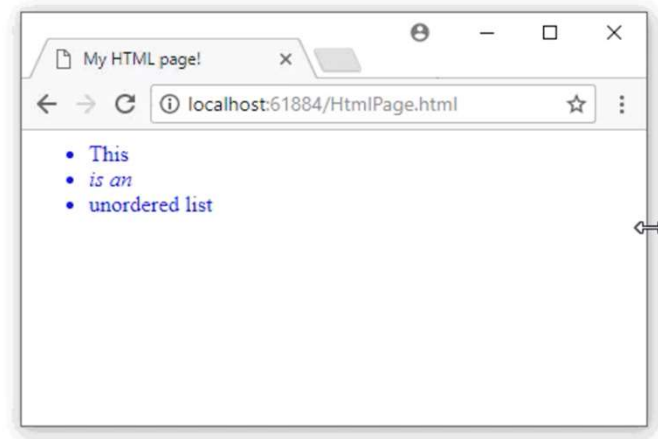
Paragraph.

Paragraph with id!

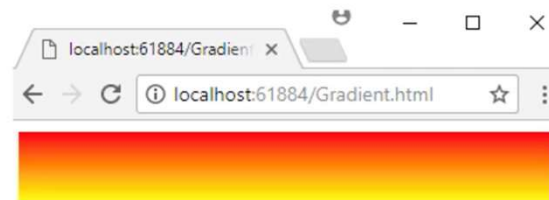


Advanced CSS Syntax

```
@media screen and (min-width: 480px) {  
  body {  
    background-color: lightgreen;  
  }  
}
```

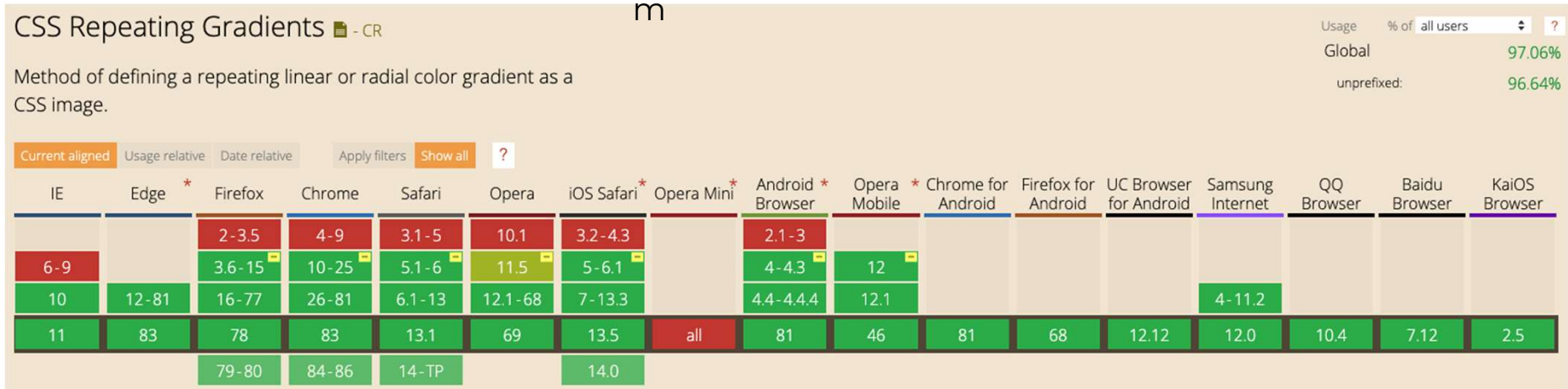


```
#grad {  
  background: linear-gradient(red, yellow);  
}
```

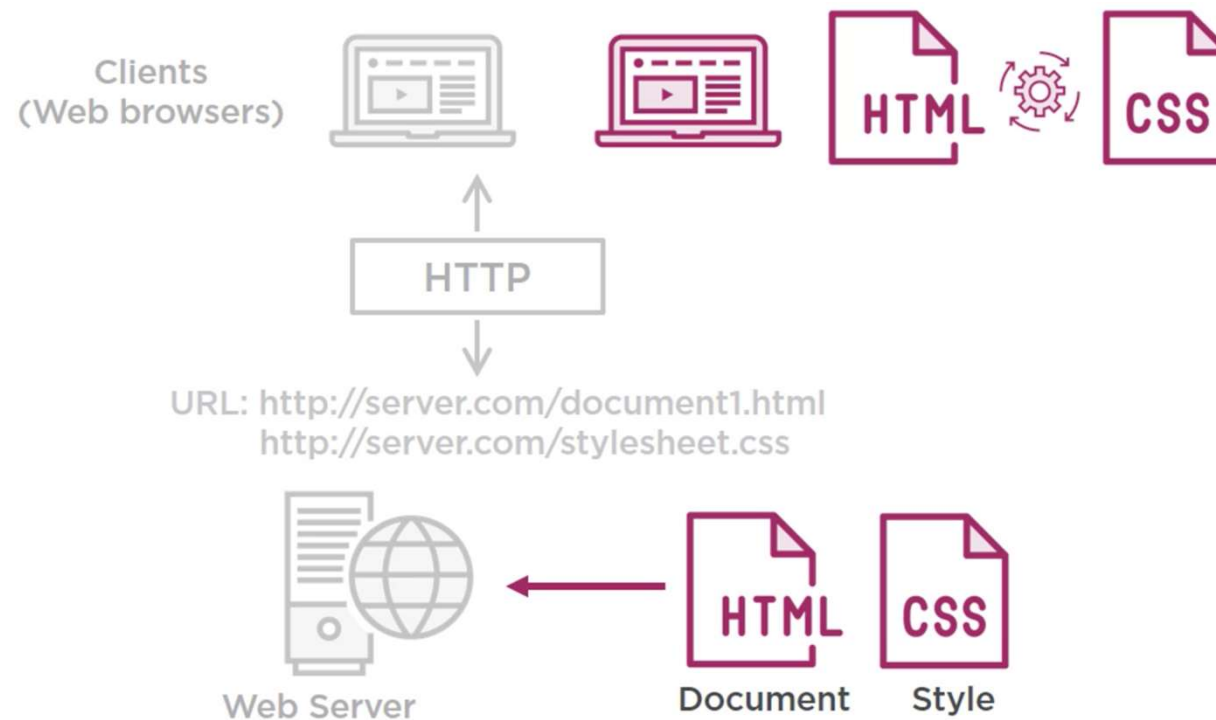


Browser Support of CSS Syntax

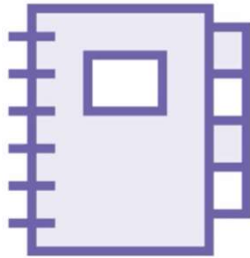
<https://caniuse.com>



Ingredients for the web

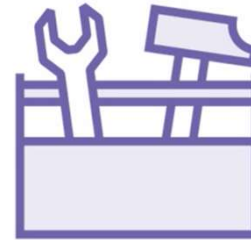


How to Write CSS from Scratch



Code editor

- Notepad++
- Atom
- Visual Studio Code



Integrated Development Environment (IDE)

- Eclipse
- Visual Studio
- Webstorm



CSS Libraries

Bootstrap

Hamburgers.css

Font-Awesome

Ionic

Materialize



CSS Libraries Example (Bootstrap)

```
<html>
<head>
  <meta charset="utf-8" />
  <link rel="stylesheet"
    href="https://ajax.aspnetcdn.com/ajax/bootstrap/3.3.6/css/bootstrap.min.css" />
```

```
<div class="row">
  <div class="col-md-4">

  </div>
  <div class="col-md-4">

  </div>
</div>
```

Cookies!

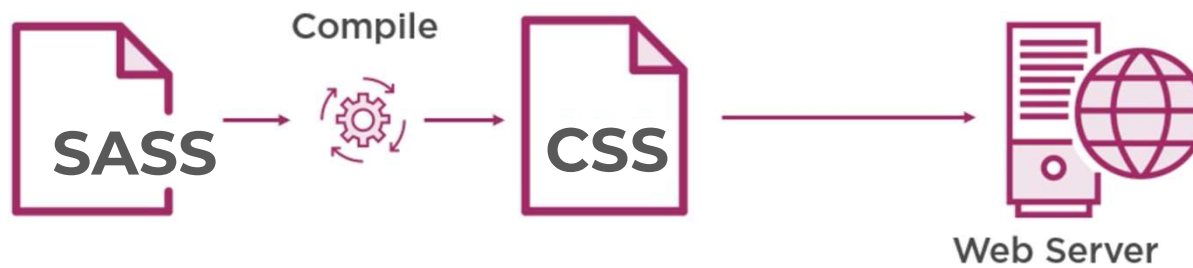
- We love all cookies!
- Chocolate Cookie,
- Butter Cookie,
- Macaroons
- You name it, we love them!

Register

- Register your store to order cookies



CSS Preprocessors



```
$font-stack: Helvetica, sans-serif;  
$primary-color: #333;
```

```
body {  
  font: 100% $font-stack;  
  color: $primary-color;  
}
```

```
body {  
  font: 100% Helvetica, sans-serif;  
  color: #333;  
}
```



Thing to Remember



CSS is the styling mechanism for the web

It is a standard of selectors, properties and attributes

- Interpreted by web browsers

CSS Separates style from content

- Maintainable
- Reusable

CSS provides many properties

- Evolving towards mobile and animations

You can use CSS

- By writing it from scratch
- By using a library (Bootstrap)
- By writing it using a preprocessor (SASS)



Tryit Editor v3.6

←

→

↺

🏠

🔒

w3schools.com/js/tryit.asp?filename=tryjs_while

🔍

☆

🛡️

🌐

🔧

📄

📷

🔗

🔍

⚙️

🔊

👤

⋮

🏠

☰

💾

🔄

🌙

Run »

Result Size: 500 x 400

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript While Loop</h2>

<p id="demo"></p>

<script>
var text = "";
var i = 0;
while (i < 10) {
  text += "<br>The number is " + i;
  i++;
}
document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

JavaScript While Loop

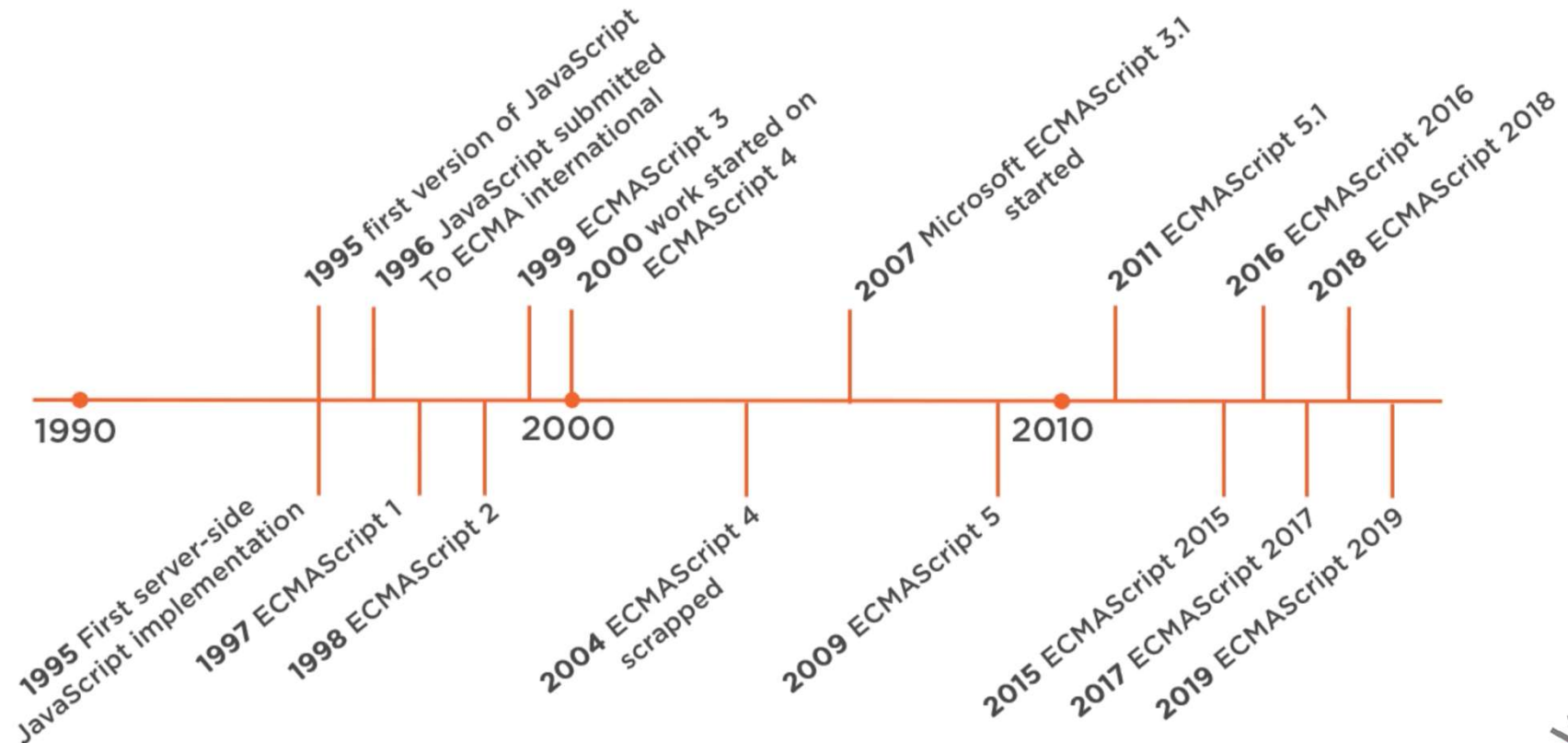
The number is 0
The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
The number is 6
The number is 7
The number is 8
The number is 9



JavaScript is a high-level,
interpreted programming
language

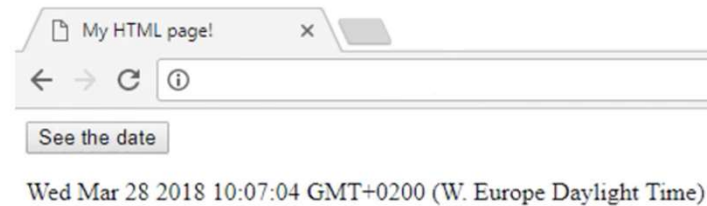


History of JavaScript



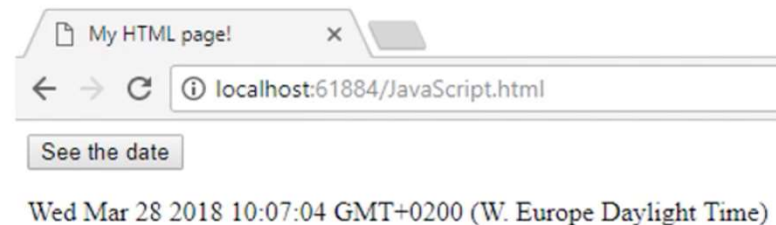
Dynamic Values with JavaScript

```
<!DOCTYPE html>
<html>
<head>
  <title>My HTML page!</title>
</head>
<body>
  <button type="button"
    onclick="document.getElementById('demo').innerHTML = Date()">
    See the date
  </button>
  <p id="demo"></p>
</body>
</html>
```



Dynamic Values with JavaScript

```
<!DOCTYPE html>
<html>
<head>
  <title>My HTML page!</title>
  <script>
    function ShowDate() {
      document.getElementById('demo').innerHTML = Date();
    }
  </script>
</head>
<body>
  <button type="button"
    onclick="ShowDate();">
    See the date
  </button>
  <p id="demo"></p>
</body>
</html>
```



Dynamic Values with JavaScript

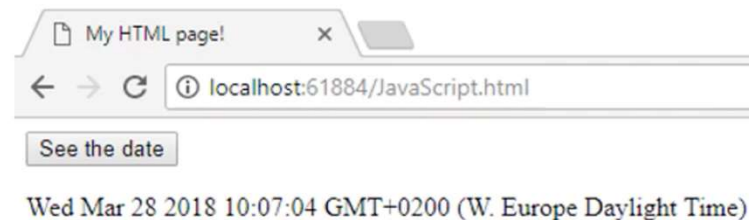
```
<!DOCTYPE html>
<html>
<head>
  <title>My HTML page!</title>
</head>
<body>
  <button type="button"
    onclick="ShowDate();">
    See the date
  </button>
  <p id="demo"></p>

  <script src="ScriptFile.js"></script>

</body>
</html>
```

ScriptFile.js

```
function ShowDate() {
  document.getElementById('demo')
    .innerHTML = Date();
}
```



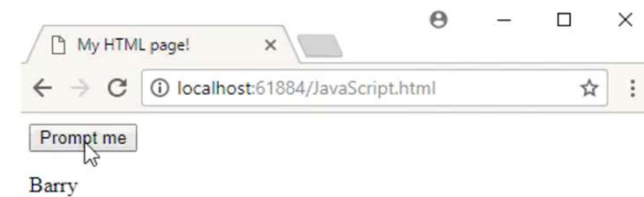
Interacting with the Browser

```
<script>

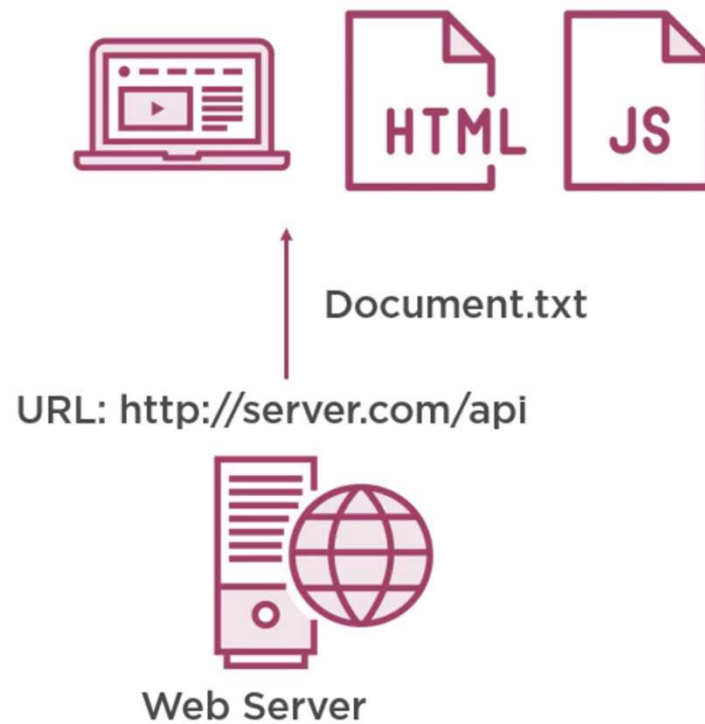
function PromptUser() {
    var txt;
    var name = window.prompt("Enter your name");
    if (name != null || name != "") {
        txt = "Hello " + name;
    }

    document.getElementById("name").innerHTML = txt;
}

</script>
```



Making AJAX Calls



Tryit Editor v3.6

w3schools.com/xml/tryit.asp?filename=tryajax_first

Run »

Result Size: 550 x 500

```
<!DOCTYPE html>
<html>
<body>

<div id="demo">
<h1>The XMLHttpRequest Object</h1>
<button type="button" onclick="loadDoc()">Change
Content</button>
</div>

<script>
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML =
        this.responseText;
    }
  };
  xhttp.open("GET", "ajax_info.txt", true);
  xhttp.send();
}
</script>

</body>
</html>
```

The XMLHttpRequest Object

Change Content



Browser Support of JavaScript

<https://caniuse.co>

XMLHttpRequest advanced features - LS

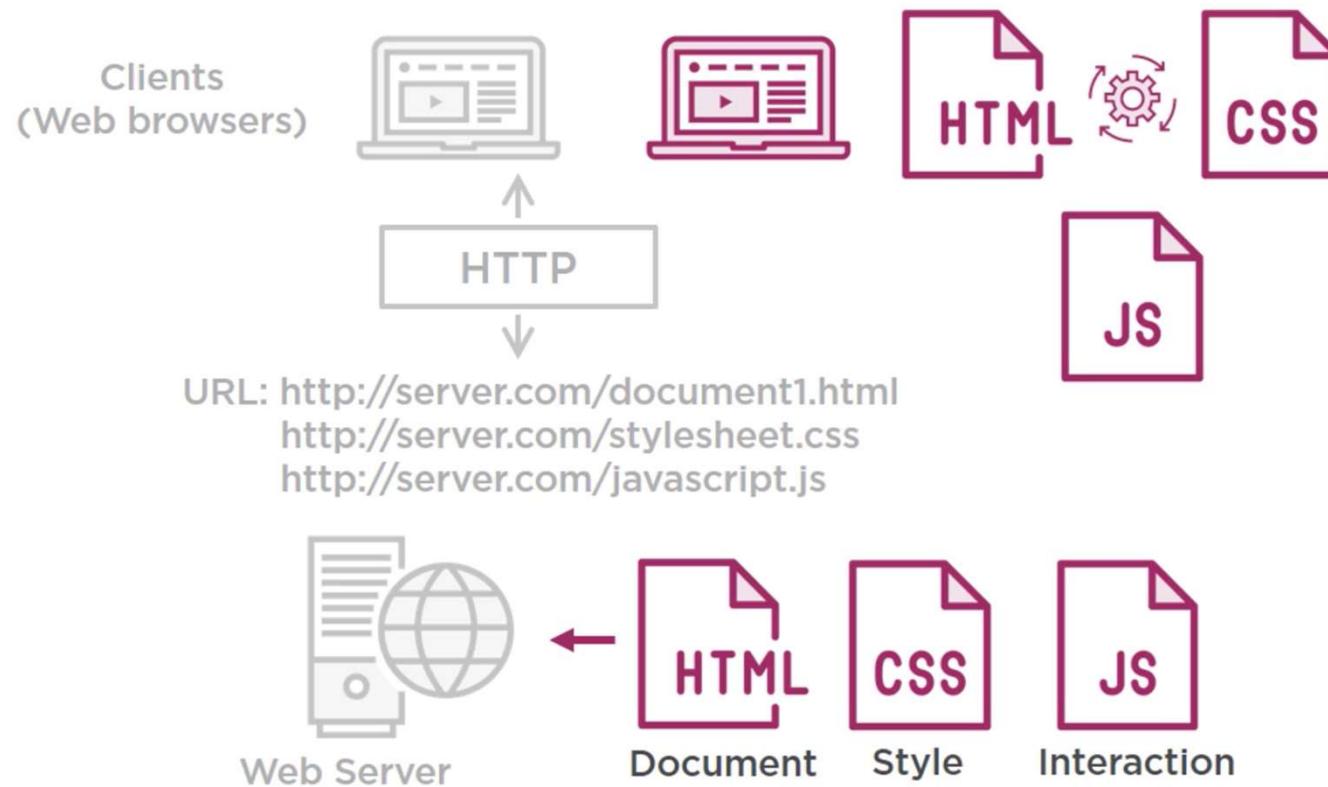
Adds more functionality to XHR (aka AJAX) requests like file uploads, transfer progress information and the ability to send form data. Previously known as **XMLHttpRequest Level 2**, these features now appear simply in the XMLHttpRequest spec.

Usage % of all users Global 95.2% + 1.89% = 97.09%

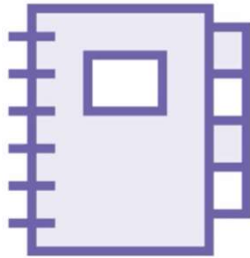
Current aligned	Usage relative	Date relative	Apply filters	Show all	?												
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baidu Browser	KaiOS Browser	
		2-3															
		1 2 3 3.5-5	4-6	3.1-4	10-11.5	3.2-4.3		2.1-2.3									
		1 2 6-9	1 2 7-28	1 2 5-6	12.1	1 2 5-6.1		1 2 3 3-4.3									
6-9		2 10-11	1 29-30	1 6.1-7	1 15-17	1 7.1		1 2 4.4									
1 10	12-81	12-77	31-81	7.1-13	18-68	8-13.3		4.4.4	12-12.1				4-11.2				
1 11	83	78	83	13.1	69	13.5	all	81	46	81	68	12.12	12.0	10.4	7.12	2.5	
		79-80	84-86	14-TP		14.0											



Ingredients for the web

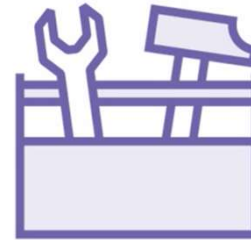


How to Write JavaScript from Scratch



Code editor

- Notepad++
- Atom
- Visual Studio Code



Integrated Development Environment (IDE)

- Eclipse
- Visual Studio
- Webstorm



JavaScript Libraries and Frameworks

JQuery

Dojo Toolkit

Prototype.js

Bootstrap

Velocity.js

D3.js

Angular

Backbone.js

Ember.js

Knockout

Node.js

Vue.js

React



Angular Example

```
import { Component, OnInit } from
 '@angular/core';
import { Hero } from '../hero';

@Component({
  selector: 'app-heroes',
  templateUrl: './heroes.component.html',
  styleUrls: ['./heroes.component.css']
})
export class HeroesComponent implements OnInit
{
  hero: Hero = {
    id: 1,
    name: 'Windstorm'
  };

  constructor() { }

  ngOnInit() {
  }
}
```

```
<h2>{{hero.name | uppercase}} Details</h2>
<div><span>id: </span>{{hero.id}}</div>
<div>
  <label>name:
    <input [(ngModel)]="hero.name"
placeholder="name"/>
  </label>
</div>
```



Thing to Remember



JavaScript is the “glue” of the web

JavaScript is not derived from Java

JavaScript is a language that you can use to

- Interact with HTML
- Interact with the web browser
- Interact with other systems (like APIs)

The language is a standard (ECMAScript)

- Interpreted by web browsers

You can use JavaScript

- By writing it from scratch
- By using a library (React)
- By using a framework (Angular)



Links to Learning Resources

- General
 - <https://www.w3schools.com/> - Free HTML, CSS, JavaScript (and more) tutorials
 - <https://caniuse.com/> - Find out which browsers support which features
- HTML
 - <https://app.pluralsight.com/paths/skills/html5> HTML skill path
- CSS
 - <https://app.pluralsight.com/paths/skills/css> CSS skill path
- JavaScript
 - <https://app.pluralsight.com/paths/skill/javascript> JavaScript skill path

