Web Development

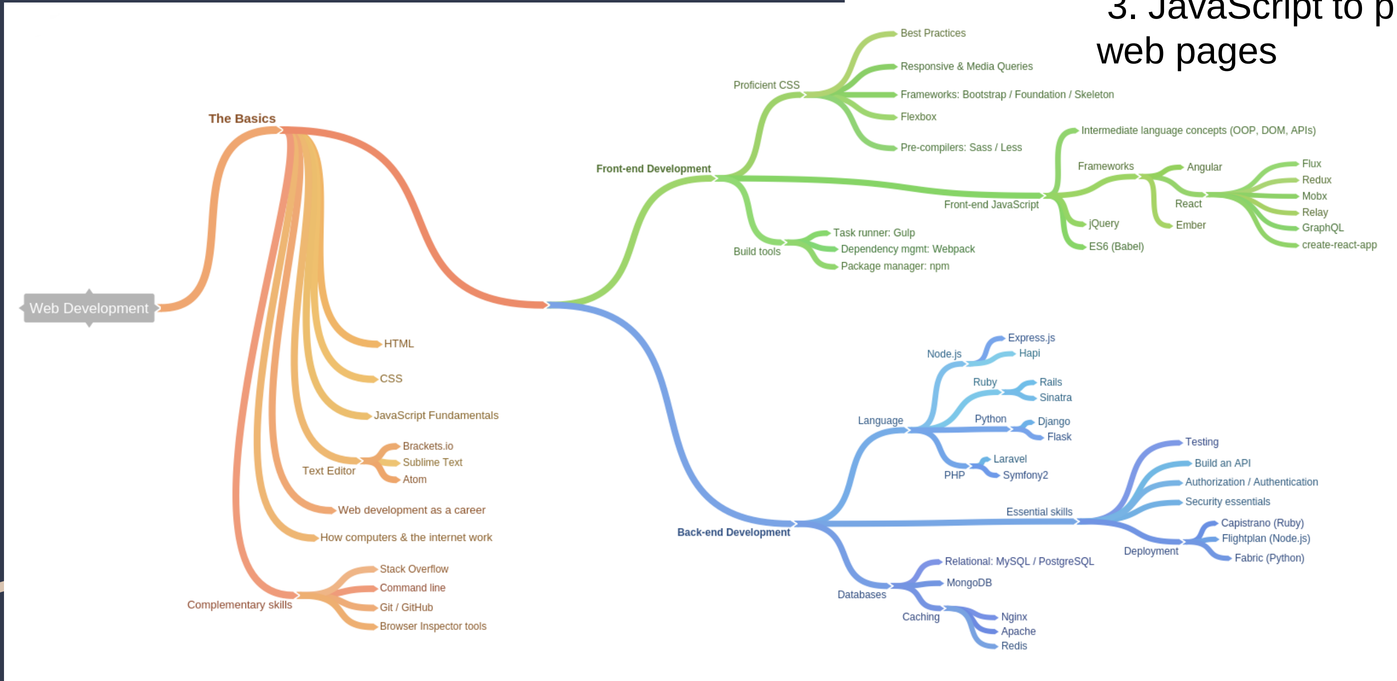
JavaScript- JS

Web Development

1. HTML to define the content of web pages

2. CSS to specify the layout of web pages

3. JavaScript to program the behavior of web pages



JavaScript

- often abbreviated as **JS**,
- a high-level, interpreted programming language
- that conforms to the ECMAScript specification
- It is a programming language that is characterized as
  - dynamic
  - weakly typed
  - prototype-based
  - multi-paradigm

JavaScript

- Alongside HTML and CSS, JavaScript is **one of the core technologies** of the World Wide Web.
- JavaScript enables interactive web pages
- an essential part of web applications.
- The vast majority of websites use it,
- major web browsers have a dedicated **JavaScript engine** to execute it

JavaScript

- As a multi-paradigm language, JavaScript supports
  - event-driven,
  - functional,
  - imperative (including object-oriented and prototype-based) programming styles.

- **Web pages** are not the only place where JavaScript is used.
- **Many desktop and server programs** use JavaScript. (Node.js)
- Some databases, (MongoDB and CouchDB),
- also use JavaScript as their programming language

JavaScript – IS –Not

- It has APIs for working with
  - text,
  - arrays,
  - dates,
  - regular expressions,
  - DOM
- Language itself does not include
- any I/O, such as networking, storage, or graphics facilities.
- It relies upon the host environment in which it is embedded to provide these features.

JavaScript – Now

- Initially only implemented client-side in web browsers,
- JavaScript engines are now embedded in many other types of host software,
  - including server-side in web servers and databases,
  - in non-web programs such as word processors and PDF software,
  - in runtime environments that make JavaScript available for writing
  - mobile and desktop applications, including desktop widgets.

JavaScript - MVC

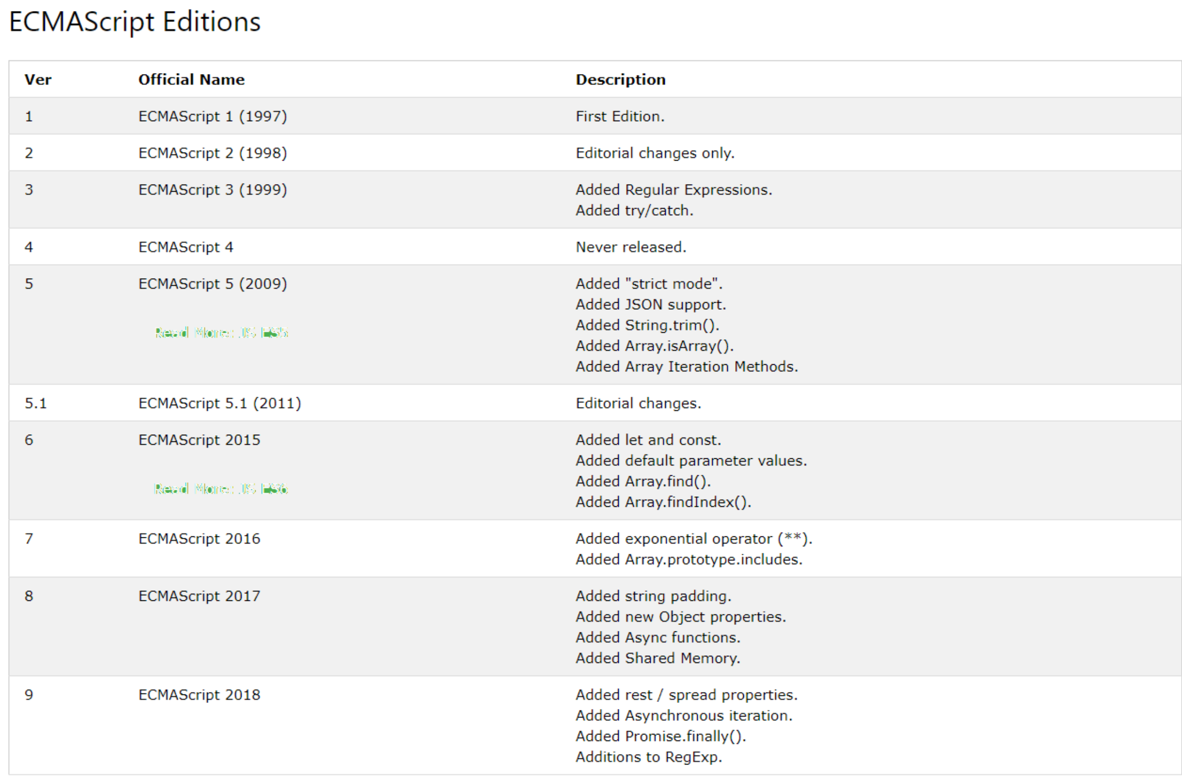
- Some JavaScript frameworks follow the **model–view–controller** paradigm designed to segregate a web application into orthogonal units to improve code quality and maintainability.
- Examples:
  - AngularJS
  - Ember.js
  - Meteor.js

JavaScript

- JavaScript and Java are completely different languages, both in concept and design.

- JavaScript was invented by Brendan Eich in **1995**,
- became an ECMA standard in **1997**.

JavaScript – Version



ECMAScript Editions

Ver	Official Name	Description
1	ECMAScript 1 (1997)	First Edition.
2	ECMAScript 2 (1998)	Editorial changes only.
3	ECMAScript 3 (1999)	Added Regular Expressions. Added try/catch.
4	ECMAScript 4	Never released.
5	ECMAScript 5 (2009) Read More: JS ES5	Added "strict mode". Added JSON support. Added String.trim(). Added Array.isArray(). Added Array Iteration Methods.
5.1	ECMAScript 5.1 (2011)	Editorial changes.
6	ECMAScript 2015 Read More: JS ES6	Added let and const. Added default parameter values. Added Array.find(). Added Array.findIndex().
7	ECMAScript 2016	Added exponential operator (**). Added Array.prototype.includes.
8	ECMAScript 2017	Added string padding. Added new Object properties. Added Async functions. Added Shared Memory.
9	ECMAScript 2018	Added rest / spread properties. Added Asynchronous iteration. Added Promise.finally(). Additions to RegExp.

- ECMAScript 3 is fully supported in all browsers.
- ECMAScript 5 is fully supported in all **modern** browsers.

ECMAScript 5
JavaScript 5

New features released in 2009

- String.trim()
- Array.isArray()
- Array.forEach()
- Array.map()
- Array.filter()
- Array.every()
- Array.some()
- Array.indexOf()
- Array.lastIndexOf()
- JSON.parse()
- JSON.stringify()
- Date.now()
- ….

JavaScript 6
ECMAScript 6

some of the new features in ES6.

- JavaScript let
- JavaScript const
- Exponentiation (**)
- Default parameter values
- Array.find()
- Array.findIndex()

Syntax

- JavaScript code must be inserted between **<script> and </script>** tags
- A JavaScript function is a block of JavaScript code, that can be executed when **called**
- Scripts can be placed in the
  - <body>
  - <head> section of an HTML page
  - in both

in <body>

```
<!DOCTYPE html>
<html>
<body>
<h1>A Web Page</h1>
<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try
it</button>

<script>
function myFunction() {
 document.getElementById("demo").innerHTML =
"Paragraph changed.";
}
</script>

</body>
</html>
```

External File

- Scripts can also be placed in external files

```
function myFunction() {
 document.getElementById("demo").innerHTML =
 "Paragraph changed.";
}
```

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

External File

Placing scripts in external files has some advantages:

- It separates HTML and code
- It makes HTML and JavaScript easier to read and maintain
- Cached JavaScript files can speed up page loads

JS Display

Display data in different ways: Writing

- into an HTML element, using innerHTML
  - document.getElementById(id)
- into the HTML output using document.write()
  - document.write(5 + 6);
  - should only be used for testing
- into an alert box, using window.alert()
  - window.alert(5 + 6);
- into the browser console, using console.log()
  - console.log(5 + 6);

Others

- Comments
- Variables
- Operators
- Arrays
- Array sort
- Random
- Booleans
- Conditions
- Functions
- Debugging
- RegExp
- Forms
- Objects
- …..