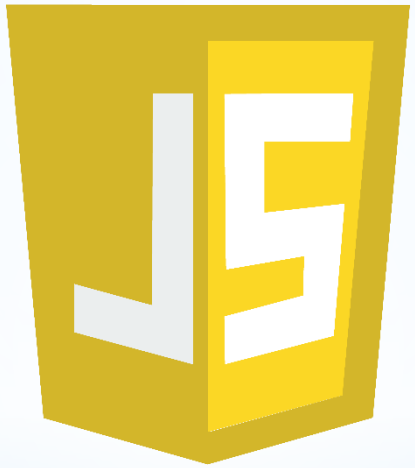
*JavaScript



Ahmed Elashry
aelashry@outlook.com

*What is JavaScript?

* JavaScript is a cross-platform, object-oriented scripting language used to make webpages interactive.

JavaScript was initially created to "make web pages alive".

*What is JavaScript?

*Most popular and widely used client-side scripting language.

- Client-side scripting refers to scripts that run within your web browser.
- Designed to add interactivity and dynamic effects to the web pages by manipulating the content returned from a web server.
- JavaScript was originally developed as LiveScript by Netscape in the mid 1990s.
- It was later renamed to JavaScript in 1995, and became an ECMA standard in 1997.
- Now JavaScript is the standard client-side scripting language for web-based applications, and it is

supported by virtually all web browsers available today, such as Google Chrome, Mozilla Firefox, Apple Safari, etc.

*Difference Between Client-side and Server side Scripting

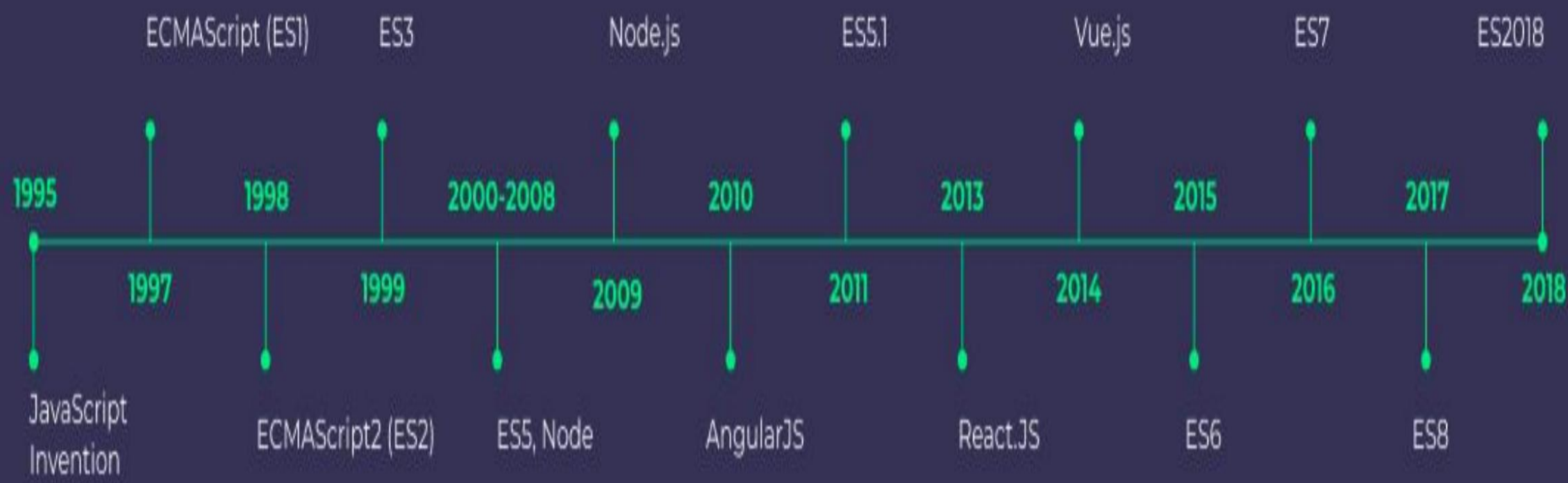
* Client-side scripting languages such as JavaScript, VBScript, etc. are interpreted and executed by the web browser.
* Server-side scripting languages such as PHP, ASP, Java, Python, Ruby, etc. runs on the web server and the output sent back to the web browser in HTML format

* Response from a server-side script is slower as compared to a client side script, because server-side scripts are processed on the remote computer not on the user's local computer

*JS Engine.

* V8 – in Chrome and Opera.
* SpiderMonkey – in Firefox.
* There are other codenames like “Trident” and “Chakra” for different versions of IE, “ChakraCore” for Microsoft Edge
* “Nitro” and “SquirrelFish” for Safari, etc

JavaScript versions timeline



*What JavaScript can Do ?

* Modify the content of a web page by adding or removing elements.
* Change the style and position of the elements on a web page.
* Monitor events like mouse click, hover, etc. and react to it.
* Perform and control transitions and animations.
* Create alert pop-ups to display info or warning messages to the user.
* Perform operations based on user inputs and display the results.
* Validate user inputs before submitting it to the server.
* Send requests over the network to remote servers, download and upload files (so-called AJAX).
* Get and set cookies, ask questions to the visitor, show messages.
* Remember the data on the client-side (“local storage”).

Advantages of JavaScript

* **Less server interaction** – You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
* **Immediate feedback to the visitors** – They don't have to wait for a page reload to see if they have forgotten to enter something.
* **Increased interactivity** – You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
* **Richer interfaces** – You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

*Limitations of JavaScript

* Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.
* JavaScript cannot be used for networking applications because there is no such support available.
* JavaScript doesn't have any multithreading or multiprocessor capabilities.
* Different tabs/windows generally do not know about each other

What You Can Not Do with JavaScript

- May not read/write files on the hard disk, copy them or execute programs. It has no direct access to OS system functions.
- There are ways to interact with camera/microphone and other devices, but they require a user's explicit permission.
- Different tabs/windows generally do not know about each other.

*Languages over JavaScript

* The syntax of JavaScript does not suit everyone's needs. Different people want different features.
* So recently a plethora of new languages appeared, which are transpired (converted) to JavaScript before they run in the browser.
* Examples of such languages:
* CoffeeScript It introduces shorter syntax, allowing us to write clearer and more precise code. Usually, Ruby devs like it.
* TypeScript is concentrated on adding "strict data typing" to simplify the development and support of complex systems. It is developed by Microsoft.
* Flow also adds data typing, but in a different way. Developed by Facebook.
* Dart is a standalone language that has its own engine that runs in non-browser environments (like mobile apps), but also can be transpiled to JavaScript. Developed by Google

*Placement in HTML File

*Script in <head>...</head> section.

*Script in <body>...</body> section.

*Script in <body>...</body> and <head>...</head> sections.

*Script in an external file and then include in <head>...</head> section.

Tip: You can place any number of <script> element in a single document.However, they are processed in the order in which they appear in the document, from top to bottom.

* **Variables** – Represents a named memory block that can store values for the program.
* **Operators** – Symbols that define how the operands will be processed.
* **Keywords** – Words that have a special meaning in the context of a language.
* **Comments** – Used to improve code readability. These are ignored by the JavaScript engine.
* **Identifiers** – These are the names given to elements in a program like variables, functions, etc.

*Syntax

abstract	double	implements	return
arguments	else	in	switch
await	enum	instanceof	synchronized
boolean	eval	int	this
break	export	interface	throw
byte	extends	let	throws
case	false	long	transient
catch	final	native	true
char	finally	new	try
class	float	null	typeof
const	for	package	var

*Reserved Keywords

* JavaScript ignores spaces, tabs, and newlines that appear in programs. You can use spaces, tabs, and newlines freely in your program and you are free to format and indent your programs in a neat and consistent way that makes the code easy to read and understand.

* Whitespace and Line Breaks

* JavaScript is case-sensitive. This means that JavaScript differentiates between the uppercase and the lowercase characters.

* JavaScript is Case-sensitive

* Each line of instruction is called a **statement**.
* Semicolons are optional in JavaScript.

* Semicolons are Optional

Comments in JavaScript

- **Single-line comments (//)** – Any text between a // and the end of a line is treated as a comment.
- **Multi-line comments (/* */)** – These comments may span multiple lines.

* All modern web browsers, Node.js as well as almost every other JavaScript environments support writing messages to a console using a suite of logging methods. The most common of these methods is console.log().
In a browser environment, the console.log() function is predominantly used for debugging purposes

* Using console.log()

*What is Variable?

*Variables are fundamental to all programming languages.

*Variables are used to store data, like string of text, numbers, etc.

*The data or value stored in the variables can be set, updated, and retrieved whenever needed.

*variables are symbolic names for values.

*Declarations

*var Declares a variable, optionally initializing it to a value.

*let Declares a block-scoped, local variable, optionally initializing it to a value.

*const Declares a block-scoped, read-only named constant.

* **variable_name** {Required} The name of the variable: used when calling it.
* = [Optional] Assignment (defining the variable)
* value {Required when using Assignment} The value of a variable [default: undefined]

* JavaScript Variables

*Identifiers

* Identifiers can include both, characters and digits. However, the identifier cannot begin with a digit.
* Identifiers cannot include special symbols except for underscore (_) or a dollar sign ($).
* Identifiers cannot be keywords. They must be unique.
* Identifiers are case sensitive.
* Identifiers cannot contain spaces.

* var declarations, wherever they occur, are processed before any code is executed. This is called hoisting.
* Assigning a value to an undeclared variable implicitly creates it as a global variable
* declaring a variable anywhere in the code is equivalent to declaring it at the top. This also means that a variable can appear to be used before it's declared.

*var

Let and Block Scope

* The block scope restricts a variable's access to the block in which it is declared. The **var** keyword assigns a function scope to the variable. Unlike the var keyword, the **let** keyword allows the script to restrict access to the variable to the nearest enclosing block.

*Variable Scope

* **Global Variables** – A global variable has global scope which means it can be defined anywhere in your JavaScript code.
* **Local Variables** – A local variable will be visible only within a function where it is defined. Function parameters are always local to that function.

* A variable declared using the var or let statement with no assigned value specified has the value of undefined.

* An attempt to access an undeclared variable results in a ReferenceError exception being thrown:

* Evaluating variables

*The const

* This declaration creates a constant whose scope can be either global or local to the block in which it is declared. An initializer for a constant is required; that is, you must specify its value in the same statement in which it's declared (which makes sense, given that it can't be changed later).Constants cannot be reassigned a value.
  * A constant **cannot be re-declared**.
  * The value assigned to a **const** variable is immutable.

JavaScript and Camel Case

* Pascal Case – We can create variables like SmartWatch, MobileDevice, WebDrive, etc. It represents the upper camel case string.

* Lower Camel Case – JavaScript allows developers to use variable names and expression names like smartwatch, mobileDevice, webDriver, etc. Here the first character is in lowercase.

* Underscore – We can use underscore while declaring variables like smart_watch, mobile_device, web_driver, etc.

*Dialog Boxes

*alert() Alert Dialog Box mostly used to send a warning message to the users.

*confirm() Confirmation Dialog Box It displays a dialog box with two buttons: OK and Cancel.

*prompt() Prompt Dialog Box useful when you want to pop-up a text box to get a user input.

*Dynamic Typing

* JavaScript decides the type of variables at runtime. So, we don’t need to care about variable data type while writing the code, providing more flexibility to write code.

* Also, you can assign the values of the different data types to a single variable. For example, if you have stored the number value of a particular variable, you can update the variable’s value with the string.

Variables Data types

- JavaScript allows you to work with three primitive data types –
  - **Numbers**, eg. 123, 120.50 etc.
  - **Strings** of text e.g. "This text string" etc.
  - **Boolean** e.g. true or false.
- trivial data types
  - **null**
  - **Undefined**
- composite data type
  - **Object**
  - Array
  - Date

* JavaScript is a dynamically typed language. That means you don't have to specify the data type of a variable when you declare it, and data types are converted automatically as needed during script execution

* Data type conversion

*Use strict

* The directive looks like a string: "use strict" or 'use strict'.
* When it is located at the top of a script, the whole script works the "modern" way.
* "use strict" switches the engine to the "modern" mode, changing the behavior of some built-in features.
* Strict mode is supported by all modern browsers.
* starting scripts with "use strict" is recommended.