Scripting: JavaScript

Roadmap

➤ JavaScript

What is JavaScript?

➢ A **scripting language** that was designed to add **interactivity** to HTML pages

➢ Invented in 1995 at Netscape Corporation (LiveScript)

➢ A JavaScript consists of lines of executable computer code

➢ A JavaScript is usually embedded directly into HTML pages

➢ JavaScript programs are run by an interpreter built into the user's web browser (not on the server)

➢ Java and JavaScript

  • Java and JavaScript are two completely different languages in both concept and design

JAVA *is to* JAVASCRIPT *as* HAM *is to* HAMSTER

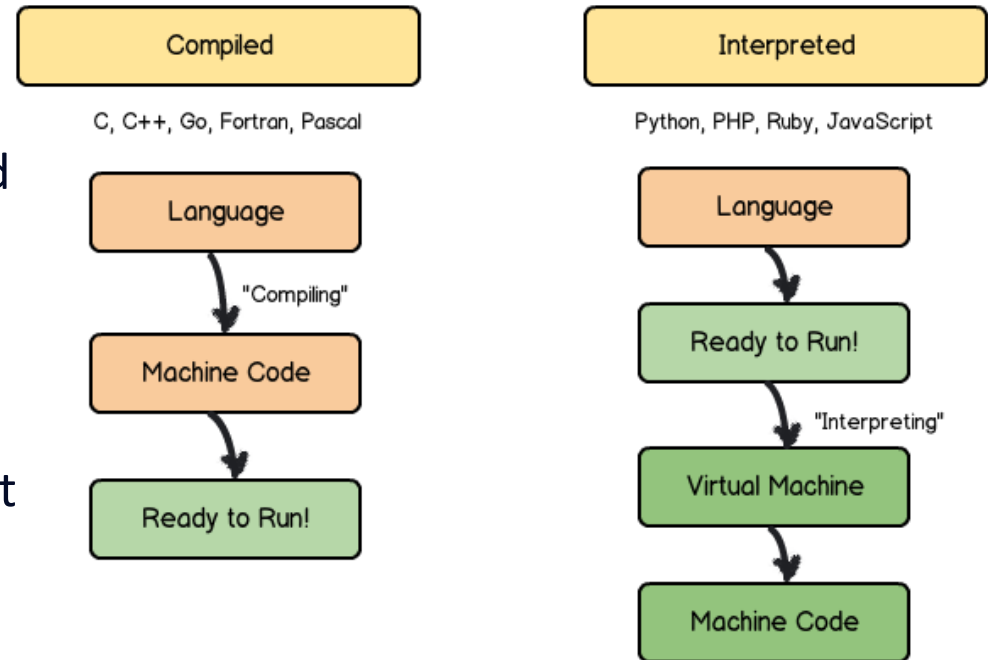That is to say, one has nothing to do with the other.

Compiled Vs Interpreted Languages

➢ Compiled

- Compiled languages are translated into machine code before they are executed

- Compilers

➢ Interpreted

- Interpreted languages are translated into machine code at runtime – without prior compilation.

- Interpreters

Compiled
C, C++, Go, Fortran, Pascal

Language

"Compiling"

Machine Code

Ready to Run!

Interpreted
Python, PHP, Ruby, JavaScript

Language

Ready to Run!

"Interpreting"

Virtual Machine

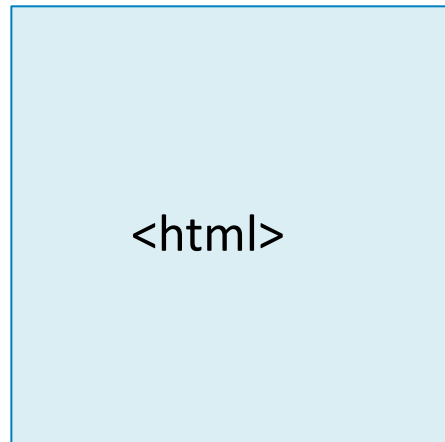| Machine Code |
|---|

Client-side scripting: where does JavaScript Fit In?

➢ Recall

- Client opens a connection to server
- Client sends request to server
- Server sends response to client
- Client and server close connection
- Client renders (displays) the response received from server
    1. Involves displaying HTML and
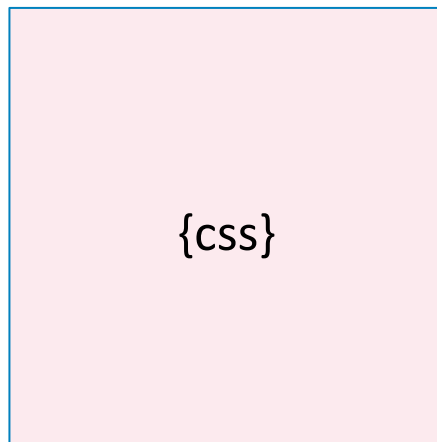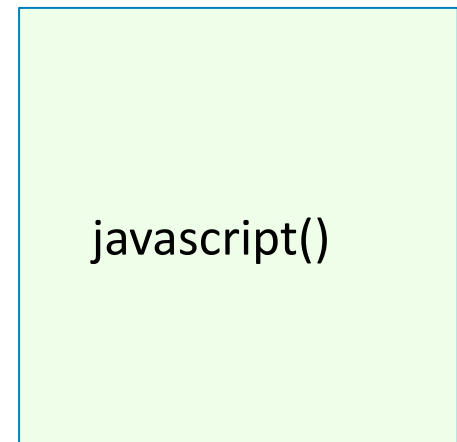   2. Running any JavaScript code within HTML

Why Study JavaScript

- Web development: HTML, CSS, JavaScript
- Separation of content, presentation and behaviour

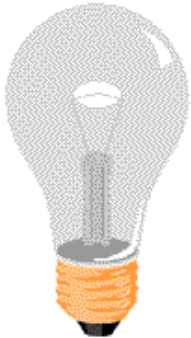| <html>	{css}	javascript()
Content layer	Presentation layer	Behaviour layer

What can JavaScript Do?

JavaScript can

- Dynamically modify an HTML page
- React to user input or to events
- Validate user input
- be used to create or modify cookies

➢ JavaScript is a full-featured programming language

➢ JavaScript user interaction does not require any communication with the server

What Can JavaScript Do?

JavaScript can change HTML attribute values.

In this case JavaScript changes the value of the src (source) attribute of an image.

Turn on the light   Turn off the light

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

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

<h2>What Can JavaScript Do?</h2>

<p>JavaScript can change HTML attribute values.</p>

<p>In this case JavaScript changes the value of the src (source) attribute of an image.</p>

<button onclick="document.getElementById('myImage').src='pic_bulbon.gif'">Turn on the light</button>

<img id="myImage" src="pic_bulboff.gif" style="width:100px">

<button onclick="document.getElementById('myImage').src='pic_bulboff.gif'">Turn off the light</button>

</body>
</html>
```

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

Using JavaScript in your HTML

➢ JavaScript can be inserted into HTML documents by using the **`<script>`** tag

```html
<html>
<head>
<title>Hello World in JavaScript</title>
</head>
<body>
<script type= "text/javascript">
document.write("Hello World!");
</script>
</body>
</html>
```

➢ Produce a page with the string "Hello World!"

➢ **`document.write`** is a standard JavaScript method for writing output to a page

➢ Semicolons are optional!

➢ However, semicolons are required if you want to put more than one statement on a single line

`type= "text/javascript"` … **`You don't need this anymore`**

Where to Put your Scripts

➢ You can have any number of scripts

➢ Scripts can be placed in the `HEAD` or in the `BODY`.

- If any code within a script is not inside a function or object, then
  - In the `HEAD`, scripts are run before the page is displayed
  - In the `BODY`, scripts are run as the page is displayed

➢ The HEAD is the right place to define **functions** and **variables** that are used by scripts within the BODY

JavaScript Functions and Events

➢ A JavaScript function is a block of JavaScript code, that can be executed when "called" for.

➢ For example, a function can be called when an event occurs, like when the user clicks a button.

```
<!DOCTYPE html>
<html>
<head>
<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>
</head>
<body>

<h2>Demo JavaScript in Head</h2>

<p id="demo">A Paragraph.</p>

<button type="button" onclick="myFunction()">Try it</button>

</body>
</html>
```

Demo JavaScript in Head

A Paragraph.

Try it

when the user clicks the button "Try it"

Demo JavaScript in Head

Paragraph changed.

Try it

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

External Scripts

- ➤ Script can also be loaded from an external file
- ➤ This is useful if you have a complicated script or set of subroutines that are used in several different documents

```html
<html>
<head>
<title>External JavaScript</title>
<script src="myJavaScript.js"></script>
</head>
<body>
</body>
</html>
```

External Scripts

➢ External scripts are practical when the same code is used in many different web pages

➢ JavaScript files have the file extension .js

➢ You can place an external script reference in <head> or <body>

➢ External scripts cannot contain <script> tags

➢ An external script can be referenced in 3 different ways:

✓ With a full URL (a full web address)

✓ With a file path (like /js/)

✓ Without any path

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

<h2>Demo External JavaScript</h2>

<p id="demo">A Paragraph.</p>

<button type="button" onclick="myFunction()">Try it</button>

<p>This example links to "myScript.js".</p>
<p>(myFunction is stored in "myScript.js")</p>

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

</body>
</html>
```

Demo External JavaScript

A Paragraph.

Try it

This example links to "myScript.js".

(myFunction is stored in "myScript.js")

External file: myScript.js

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

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

Demo External JavaScript

Paragraph changed.

Try it

This example links to "myScript.js".

(myFunction is stored in "myScript.js")

JavaScript Variables

➢ Variables are Containers for Storing Data

➢ JavaScript Variables can be declared in 4 ways:

    ✓ Automatically

    ✓ Using var

    ✓ Using let

    ✓ Using const

➢ ES6 (2015) introduced the two new JavaScript keywords: let and const

    ✓ These two keywords provided **Block Scope** in JavaScript

```
{
    let x = 2;
}
// x can NOT be used here
```

```
{
    var x = 2;
}
// x CAN be used here
```

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

JavaScript Variables

➢ When to Use var, let, or const?

  ✓ Always declare variables

  ✓ Always use const if the value should not be changed

  ✓ Always use const if the type should not be changed (Arrays and Objects)

  ✓ Only use let if you can't use const

  ✓ Only use var if you MUST support old browsers

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

JavaScript Operators and Constructs

➤ JavaScript has most of the operators used in other programming languages such as JAVA, C++

➤ Arithmetic Operators

   +, -, *, /, %, ++, --

➤ Assignment Operators

   =, +=, -=, *=, /=, %=

➤ Logical Operators

   &&, ||, !

➤ Comparison Operators

   <, >, <=, >=, ==, !=

➤ Constructs

   if, else, while, for, switch, case

Simple User Interaction: JavaScript PopUp

➢ There are three built-in methods of doing user interaction

✓ Alert Box
  - An alert box is often used if you want to make sure information comes through to the user
  - `window.alert()` alerts the user that something has happened

✓ Confirm Box
  - A confirm box is often used if you want the user to verify or accept something
  - `window.confirm()` asks the user to confirm or cancel something

✓ Prompt Box
  - A prompt box is often used if you want the user to input a value before entering a page
  - `window.prompt()` asks the user to enter some text

➢ All methods can be written without the window prefix

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

<h2>JavaScript Alert</h2>

<button onclick="myFunction()">Try it</button>

<script>
function myFunction() {
  alert("I am an alert box!");
}
</script>

</body>
</html>
```

JavaScript Alert

Try it

www.w3schools.com says

I am an alert box!

OK

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

<h2>JavaScript Confirm Box</h2>


<button onclick="myFunction()">Try it</button>

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

<script>
function myFunction() {
  var txt;
  if (confirm("Press a button!")) {
    txt = "You pressed OK!";
  } else {
    txt = "You pressed Cancel!";
  }
  document.getElementById("demo").innerHTML = txt;
}
</script>

</body>
</html>
```

JavaScript Confirm Box

Try it

JavaScript Confirm Box

Try it

You pressed OK!

JavaScript Confirm Box

Try it

You pressed Cancel!

---

www.w3schools.com says

Press a button!

OK   Cancel

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

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

<h2>JavaScript Prompt</h2>

<button onclick="myFunction()">Try it</button>

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

<script>
function myFunction() {
  let text;
  let person = prompt("Please enter your name:", "Harry Potter");
  if (person == null || person == "") {
    text = "User cancelled the prompt.";
  } else {
    text = "Hello " + person + "! How are you today?";
  }
  document.getElementById("demo").innerHTML = text;
}
</script>

</body>
</html>
```

JavaScript Prompt

Try it

JavaScript Prompt

Try it

Hello Harry Potter! How are you today?

JavaScript Prompt

Try it

User cancelled the prompt.

www.w3schools.com says

Please enter your name:

Harry Potter

OK   Cancel

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