JAVA SCRIPT

Dr. Jose Annunziato

What is JavaScript

- JavaScript is the programming language of the Web

- Historically JavaScript executed on browser, but now also on servers

- Use `script` tag to include JavaScript in Web pages

```
<script>
    alert("Hello World!");
</script>
```

Some simple examples

Alert pop ups only work on browsers

```
alert("Hello World!");
```

Writing to the browser's console or command line

```
console.log("Hello World!");
```

Writing to the browser's current HTML document

```
document.write("Hello World!");
```

Some simple examples

Appending content to an existing element on the current HTML document

```
document.getElementById("hello")
    .innerHTML = "Hello World!";
```

Retrieving elements by ID

Consider the following HTML

```
<h1 id="hello">Hello World!</h1>
```

JavaScript defines global ***document*** object referring to current HTML document

The ***document*** object has functions to manipulate the HTML document programmatically

Retrieving elements by ID

Use **document.getElementById()** to refer to an element in HTML

```
<script>

  let theElement =

    document.getElementById("hello");

</script>
```

Manipulating an element's style

Use an elements `style` property to manipulate it's style

Change it's foreground color

```
theElement.style.color = "red"
```

Change it's font size

```
theElement.style.fontSize = "25px";
```

Manipulating an element's style

Hide it

```
theElement.style.display = "none";
```

Show it

```
theElement.style.display = "block";
```

Displaying the current date

Create a new date instance using `new Date()`

```
let today = new Date();
```

Display the month

```
document.write("<h2>The Month</h2>")
document.write(today.getMonth());
```

Display the year

```
document.write("<h2>The Year</h2>")
document.write(today.getFullYear());
```

Display arrays

Use square brackets to declare arrays

```
let scores = [90,89,78,67];
```

Display arrays

```
document.write("<h1>Scores</h1>")
document.write(scores);
```

Display arrays

Calculate total and average

```
let total = 0;
for(let i=0; i<scores.length; i++) {
   total += scores[i];
}
let average = total / scores.length;
```

Display total and average

```
document.write("total = " + total + "<br/>");
document.write("average = "+average+"<br/>");
```

Creating dynamic lists

```
document.write("<ul>");
for(let i in scores) {
    document.write("<li>scores["+i+"] = " +
                        scores[i]);
    document.write("</li>");
}
document.write("</ul>");
```

Creating dynamic tables

```
let table = "<table>";
for(var j in scores) {
    let row = "<tr>";              let col = "<td>"
    col += "scores["+j+"]";  col += "</td>";
    row += col;                    col = "<td>"
    col += scores[j];              col += "</td>";
    row += col;                    row += "</tr>";
    table += row;
}
table += "</table>";
document.write(table);
```

Binding events

Use `onclick` attribute to bind a click event on an element to a function

```
<button onclick="sayHello()">
    Say Hello
</button>

<script type="text/javascript">
    function sayHello() {
        alert("Hello!")
    }
</script>
```

JavaScript in HEAD, BODY, external file

Declare *script* elements in *HEAD* tag

```
<head>
  <script type="text/javascript">
    function sayHello() {
      alert("Hello!")
    }
  </script>
</head>
```

JavaScript in HEAD, BODY, external file

Best practice to declare in separate file. Can reuse

```
<head>
    <script src="hello.js"></script>
</head>
```

If big, slow to load file, often declared at bottom of BODY

```
<body>
    ...
    <script src="hello.js"></script>
</body>
```

InnerHTML

Consider the following HTML

```
<h1>Simple Math</h1>
2 + 3 = <span id="result"></span>
```

Use `innerHTML` property to modify existing content

```
let result = document.getElementById("result");
result.innerHTML = 2 + 3;
```

Reading an input field's value

Consider the following HTML with default value 123

```
<input id="a" value="123"/>
```

Use `value` property to retrieve an input's current value

```
let theElement = document.getElementById("a");
let theValue = theElement.value;
alert(theValue);
```

Addition calculator

Consider the following HTML

```
<h1>Input Fields</h1>
<input id="a"/> +
<input id="b"/> =
<span  id="c"></span>
<br/>
<button onclick="addAplusB()">
   Add A and B
</button>
```

Addition calculator

The function `addAplusB` adds #a and #b and displays result in #c

```
function addAplusB() {
    let a = document.getElementById("a").value;
    a = parseInt(a);
    let b = document.getElementById("b").value;
    b = parseInt(b);
    document.getElementById("c").innerHTML = a + b;
}
```

JavaScript Object Notation (JSON)

Objects group various properties and values to represent a single thing / object. Consider describing a house with properties rooms, baths, and square feet

Use {} to declare object literals

```
const house = {
    rooms: 3,
    baths: 3,
    sqf: 1300
};
```

Syntax referred to as JSON or JavaScript Object Notation

Dot notation

Use dot notation retrieve values of object properties, e.g., `object.property`

Consider the following HTML

```
<h1>House</h1>
<ul>
    <li id="rooms"></li>
    <li id="baths"></li>
    <li id="squareFeet"></li>
</ul>
```

Dot notation

Use dot notation retrieve values of object properties

```
const house = {
    rooms: 3,
    baths: 3,
    sqf: 1300
};
document.getElementById("rooms").innerHTML = house.rooms;
document.getElementById("baths").innerHTML = house.baths;
document.getElementById("squareFeet").innerHTML =
house.sqf;
```

Use strict

Enforce some best practices by configuring strict language validation

Browser will then disallow things such as global variables

```
"use strict";
var abc = 123;
```

IIFE

Functions and variables not in functions are global

All variables, not declared in a function, are globally accessible

Best declared in IIFE: ***Immediately Invoked Function Expression***

```
(function() {
 ...
})()
```

EXAMPLE

Jose Annunziato

ToDo list in JavaScript

todo.html

```html
<input id="title"/>
<br/>
<textarea id="description"></textarea>
<br/>
<button id="addBtn">Add</button>
<ul id="todos">
</ul>
<script src="todo.js"></script>
```

ToDo list in JavaScript

```javascript
var titleFld = document.getElementById("title");
var descriptionFld =
document.getElementById("description");
var addBtn = document.getElementById("addBtn");
var todosUl = document.getElementById("todos");

addBtn.onclick = addBtnHandler;

function addBtnHandler() { ... }
```

ToDo list in JavaScript

```javascript
function addBtnHandler() {
    var titleStr = titleFld.value;
    var descriptionStr = descriptionFld.value;
    var id = (new Date()).getTime();
    var note = "<li id='"+id+"'>"+titleStr
        +"<span
onclick=\"deleteNote('"+id+"')\">x</span></li>";
    todosUl.insertAdjacentHTML("beforeend", note);
}
```

ToDo list in JavaScript

```javascript
function deleteNote(id) {
  var todo = document.getElementById(id);
  todo.remove();
}
```