
ACIT 1620 - FUNDAMENTAL WEB TECHNOLOGIES

WEEK 10

NEDA CHANGIZI



Today's Learning Outcomes

- Use `<script>` tag to load and run JS code from HTML documents
- Declare and use arrays in JavaScript.
- Declare and use objects in JavaScript.
- Write different kinds of loops in JavaScript.

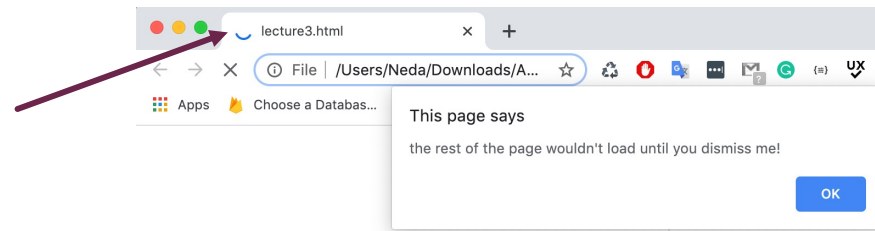
Internal JavaScript (inside HTML file)

`<script>...</script>`

- How do you create an alert or write to the console from your HTML file?

```
<body>
  <h1>Intro to DOM</h1>
  <script>
    alert("The rest of the page wouldn't load until you dismiss me!")
    console.log("Hello");
  </script>
  <p>First paragrph</p>
</body>
```

What's happening?



External JavaScript

`<script src=...></script>`

```
<body>
  <h1>Intro to DOM</h1>
  <script src="scripts/script.js"></script>
  <p>First paragraph</p>
</body>
```

JS script.js ×

Users > Neda > Downloads > ACIT1620 > week8 > scripts > JS script.js

```
1  alert("The rest of the page wouldn't load until you dismiss me!");
2  console.log("Hello");
```

JavaScript Render Blocking

- HTML rendering is stopped until the JavaScript code is executed:

How JavaScript Loads: Right away

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

- HTML parsing
- JS download
- JS execution



- Should the script tag go in the head or body tag?

Script loading strategies - async and defer

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

- HTML parsing
- JS download
- JS execution



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

- HTML parsing
- JS download
- JS execution



Loop – Activity

- Write a piece of code to log to console powers of 2 (from 2 to 1024).
 - Using for loop
 - Using while loop

do/while Loop

- How is do/while loop different from while loop?
- Write a piece of code using do/while loop to get user's name and show an alert with a welcome message including user's name. Keep asking user for an input if a number is entered instead of a string.

Watch out for Infinite loops

Note the exit condition is missing!

- When will these loops finish?

```
for (let current = 20; ; current = current + 1) {  
  if (current % 7 == 0) {  
    console.log(current);  
  }  
}
```

```
let number = 0;  
while (number <= 12) {  
  console.log(number);  
}
```

Accessing Array Elements

- Each element is stored at an index (starting at zero).
- How do we access “cheese”?

```
let shopping = ['bread', 'milk', 'cheese', 'hummus', 'noodles'];
```

- How can I access 1?
 - multidimensional array

```
let random = ['tree', 795, [0, 1, 2]];
```

- What if I access an index that doesn't have anything stored?

```
shopping[10];
```

Adding/Removing Array Elements

```
let fruits = ['Apple', 'Banana'];
```

- Use a built-in array method to
 - Add 'Orange' to the end of the array
 - Remove the last element of the array
 - Add 'Mango' to the front of the array
 - Remove the first element of the array
- Look up some other array methods like splice(), indexOf(), slice(), etc.

Array – Activity

- Write a loop in 2 different ways to print only odd numbers from this array:

```
let sequence = [1, 1, 2, 3, 5, 8, 13];
```

```
for (let i = 0; i < arr.length; i = i + 1) {
```

```
for (let item of arr) {
```

JavaScript Objects

- In JavaScript, almost "everything" is an object.
 - **Properties** of an object are stored in **name(key) : value** pairs
- To access values of object properties:

objectName.propertyName e.g. person.age

This is useful when the
propertyName is a variable

objectName["propertyName"] e.g. person["age"]

```
const person = {  
  name: ['Bob', 'Smith'],  
  age: 32,  
  gender: 'male',  
  interests: ['music', 'skiing'],  
};
```

- Console.log this statement using the person object defined above:
"my name is Bob Smith. I am interested in music and skiing"

Loop through object's properties

- Write a loop to print out all the keys and values of this object:
 - e.g. person's name is Celeste

```
for (let key in obj) {
```

```
let person = {  
  name: 'Celeste',  
  age: 31,  
  height: '170cm'  
};
```

Array of Objects – Activity

- Write a piece of code to iterate through the “students” array and print out the number of student’s in CIT program with GPA above 3.

```
let student1 = {  
  ID: "A00022",  
  GPA: 3.0,  
  program: "CIT"  
};
```

```
let student2 = {  
  ID: "A01000",  
  GPA: 3.1,  
  program: "CST"  
};
```

```
let student3 = {  
  ID: "A00114",  
  GPA: 3.2,  
  program: "CIT"  
};
```

```
let students = [student1, student2, student3];
```

Lab 7 – Personal Website (JavaScript Basic)

- Start where you left off in the previous lab (personal website). Do the following tasks in a new branch:
 - Create a JS file and save it in a separate folder. Add a line in course.html page to load and run the JS file (don't forget the appropriate attribute). In your JS file:
 - Define a constant array called courseList. This array should have at least 3 items in it. Each item should be in the format of a JS object: with code and name properties. e.g. { code: "ACIT 1620" , name: "Web Fundamental Technologies"}.
 - Get a 4-digit number from the user e.g. "1620". Prompt the user again if invalid (e.g. a string or a number with fewer digits) data is entered.
 - Write a loop to iterate through courseList array and check if the "code" property of any of the items in the array contains the provided number by the user.
 - If yes, log to console "Yes I am taking the course: ACIT 1620 – Web Fundamental Technologies" (use template literal to generate this string)
 - If not, add a new object to the array with code property equal to the value user entered, and null for the name property. Log a success message to the console.
- Merge your code back to the main branch and push the changes to your remote repo.
- Submit your GitHub repo link along with your zipped folder.