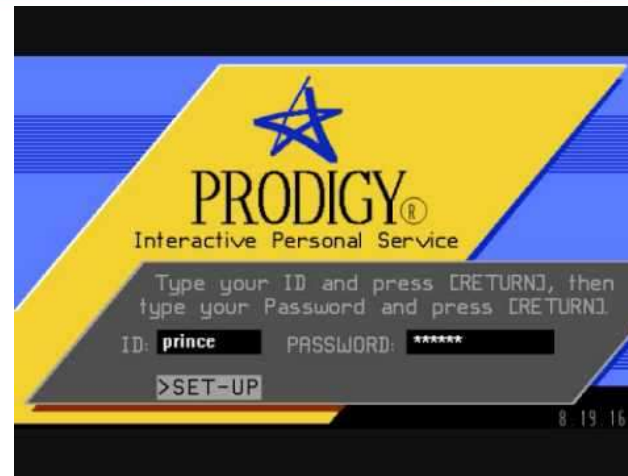


MODULE 4: CLIENT SIDE PROGRAMMING

Introduction to Client Side Scripting with JavaScript



Days of Dial-Up



Static is booooring

- Client side scripting
 - Executes code on the user's browser
 - Allows us to interact with the HTML rendered and the CSS sent by the server
 - Creates less stress on the server
 - More interactive engaging experiences for users

Uses for client side scripting

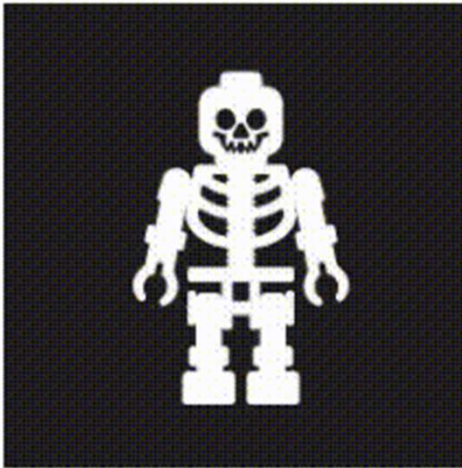
- Responding to events (click, keypress, scroll, resize)
- Can interact with other web services / APIs to dynamically update the page
- Can manipulate the loaded page without refresh

Separation of Concerns

- What provides the content on a web page
 - HTML
- What provides the look of the web page
 - CSS
- What provides the interaction/behaviors of the web page
 - Client Side Scripting -- Javascript

The parts...

HTML
structure



Javascript

- C# requires a runtime to execute.
- Javascript requires a browser to execute.
- C# is compiled code.
- Javascript is interpreted code.
- C# is statically typed language.
- Javascript is dynamically typed language.

Language Types

- A **statically** typed language enforces the data type constraints at compile-time
- A **dynamically** typed language infers the data type of what the variable holds at run-time

LET'S CODE!



Variable Declaration

- var – old style declaration.
- const
- let

```
{  
    var x = 2;  
}  
Console.log(x);
```

```
{  
    let x = 2;  
}  
Console.log(x);
```

Strict Equality vs. Loose Equality

- **Strict Equality** compares two operands for type and value equality (===)
- **Loose Equality** compares two operands for value only after converting to a common type. It will consider two values equal if they have the same value (==)

All our old friends are back!



Comparison Operators

- A **boolean expression** is an expression that produces a boolean value (true or false) when evaluated

Operator	Meaning
==	Equals To
!=	Not Equal To
>	Greater Than
<	Less Than
>=	Greater Than or Equal To
<=	Less Than or Equal To

Logical Operators

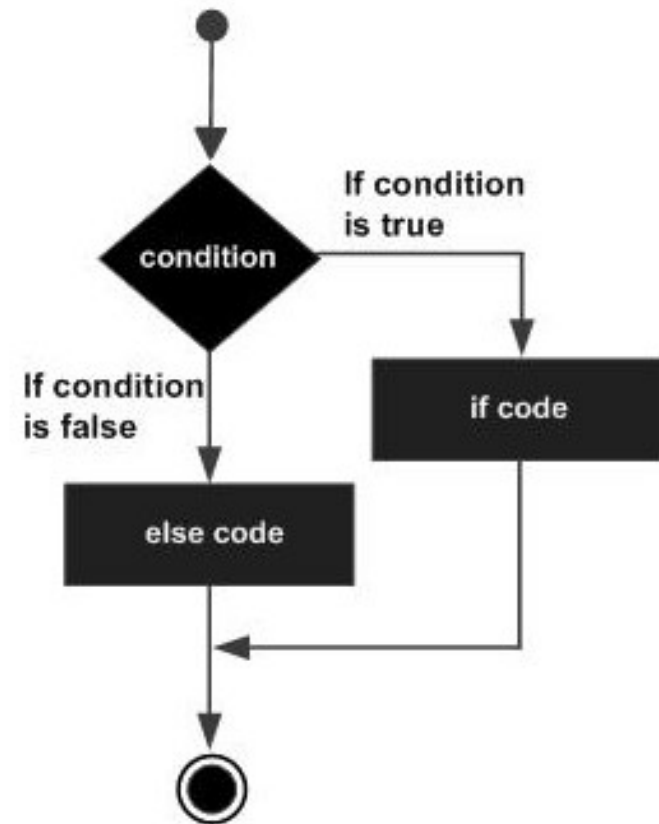
A	B	!A	A && B	A B	A^B
True	True				
True	False				
False	True				
False	False				

Logical Operators

A	B	!A	A && B	A B	A^B
True	True	False	True	True	False
True	False	False	False	True	True
False	True	True	False	True	True
False	False	True	False	False	False

If Statement

```
If (condition)
{
    <if code>
} else {
    <else code>
}
```



Code Blocks and Scope

- Code that needs to belong together as a single unit can be written in **blocks**.

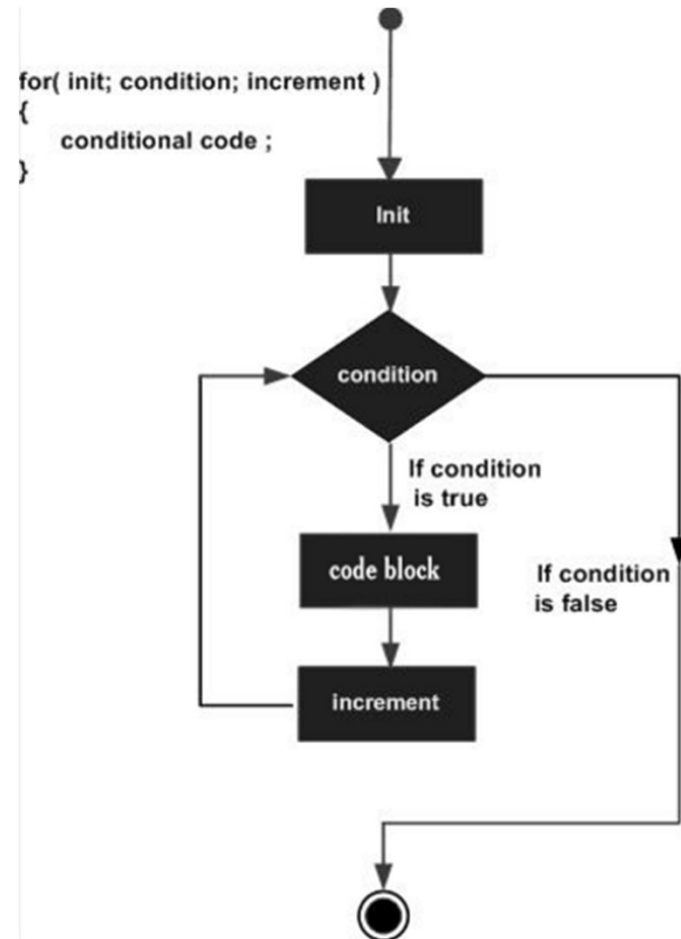
```
{  
  let length;  
  let width;  
  let area;  
  area = length * width;  
}
```

Arrays

- not fixed in size
- methods:
 - .concat()
 - .push()
 - .pop()

Accessing Elements in an Array

- For loop allows you to check each element in an array.



Memorize this code segment!!

```
for(let i = 0; i < scores.length; i++) {  
    ...  
}
```

While and Do While

while (condition)

{

 loop to execute

}

do

{

 loop to execute

} while (condition)

Object Literals

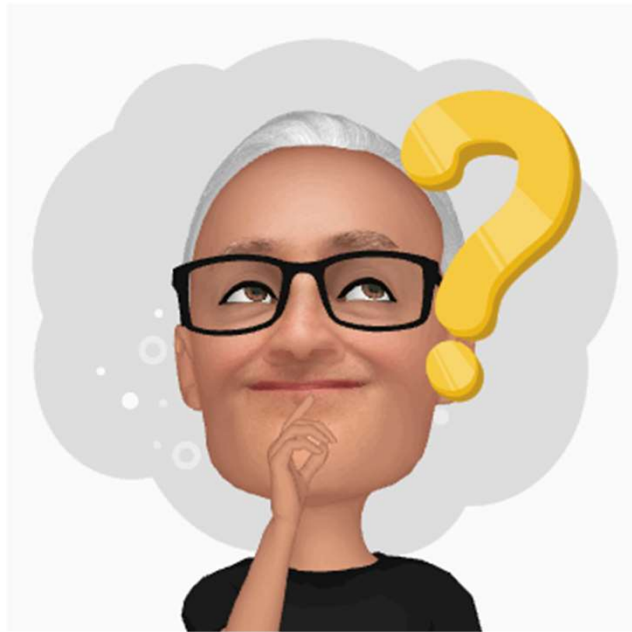
```
function createObject() {  
  const obj = {  
    firstName: "John",  
    lastName: "Smith",  
    age: 40  
  };  
  
  return obj;  
}
```

LET'S CODE!



ELEVATE  YOURSELF

WHAT QUESTIONS DO
YOU HAVE?



Reading for tonight:

Functions

