JAVASCRIPT TOPIC 3

InfoTech 37
Jomar B. Colao
College Instructor

Intro

- JavaScript is a programming language. The Syntax rules define how the language is constructed.

JavaScript Syntax

- JavaScript is a scripting language. It is a lightweight, but powerful, programming language.
- Syntax definition: "The principles by which sentences are constructed in a language."
- The sentences of a programming language are called computer statements, or just statements.

JavaScript Literals

- In a programming language, a literal is a constant value, like 3.14.
- Number literals can be written with or without decimals, and with or without scientific notation (e):

```
 3.14
1001
123e5
```

Demo 1

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

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

          <script>
          document.getElementById("demo").innerHTML = 123e5;
          </script>

          </body>
</html>
```

Demo 2

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

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

    <script>
    document.getElementById("demo").innerHTML = 'John Doe';
    </script>

    </body>
</html>
```

String literals can be written with double or single quotes:

"John Doe"

'John Doe'

Demo 3

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

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

    <script>
    document.getElementById("demo").innerHTML = 5 * 10;
    </script>

    </body>
</html>
```

Expression literals evaluates (computes) to a value:

5 + 6

5 * 10

JavaScript Variables

- ❑ In a programming language (and in normal algebra), named variables store data values.
- ❑ JavaScript uses the var keyword to define variables, and an equal sign to assign values to variables (just like algebra):

```
var x, length
x = 5
length = 6
```

Demo 4

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

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

	<script>
	var length;
	length = 6;
	document.getElementById("demo").innerHTML = length;
	</script>

	</body>
</html>
```

JavaScript Operators

JavaScript uses arithmetic operators to compute values (just like algebra):

(5 + 6) * 10

Demo 5

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

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

	<script>
	document.getElementById("demo").innerHTML = (5 + 6) * 10;
	</script>

	</body>
</html>
```

Type of JS Operators

- Type Examples Description
  - Assignment, arithmetic, and bitwise operators = + - * / Described in JS Operators
  - Conditional, comparison, and logical operators == != < > Described in JS Comparisons

JS Statements

In HTML, JavaScript statements are written as sequences of "commands" to the HTML browser.

Statements are separated by semicolons:

```
x = 5 + 6;
y = x * 10;
```

JS Keywords

A JavaScript statement often starts with a keyword. The var keyword tells the browser to create a new variable:

```
var x = 5 + 6;
var y = x * 10;
```

JS Identifiers

- All programming languages must identify variables, functions, and objects, with unique names.
- These unique names are called identifiers.
- Identifier names can contain letters, digits, underscores, and dollar signs, but cannot begin with a number.
- Reserved words (like JavaScript keywords) cannot be used as identifiers.

JavaScript Comments

Not all JavaScript statements are "commands". Anything after double slashes // is ignored by the browser:

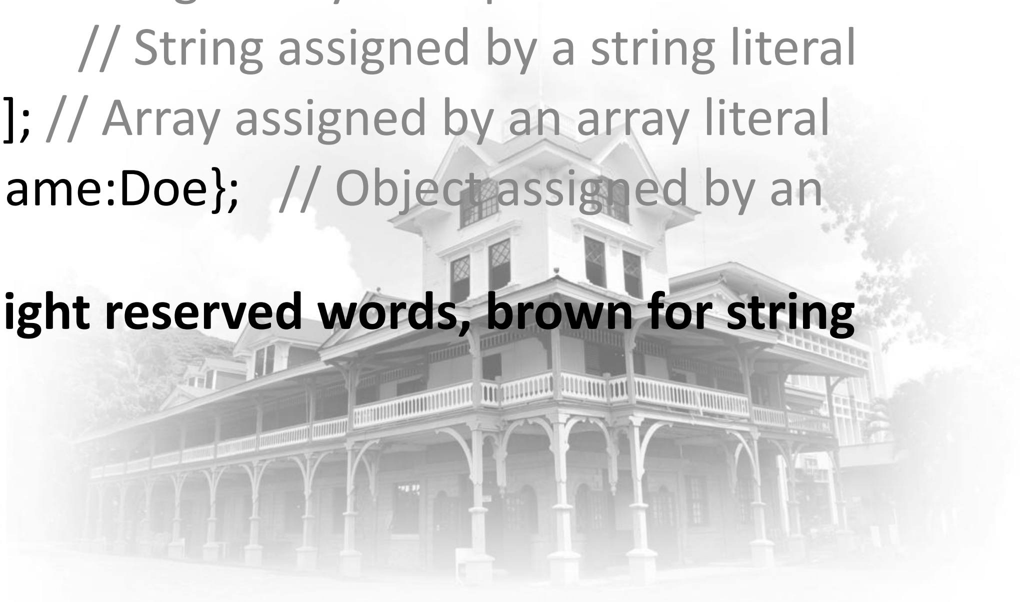
```
// I will not be executed
```

JS Data Types

JavaScript variables can hold many types of data: numbers, text strings, arrays, objects and much more:

```
var length = 16;                    // Number assigned by a number literal
var points = x * 10;                // Number assigned by an expression literal
var lastName = "Johnson";                   // String assigned by a string literal
var cars = ["Saab", "Volvo", "BMW"]; // Array assigned by an array literal
var person = {firstName:John, lastName:Doe};   // Object assigned by an
object literal
```

Note We use blue color to highlight reserved words, brown for string literals, and green for comments.



JS Functions

- ❑ JavaScript statements written inside a function, can be invoked many times (reused):
- ❑ Invoke a function = Call upon a function (ask for the code in the function to be executed).

```
function myFunction(a, b) {
    return a * b;                // returns the product of a
and b
}
```

JS is Case Sensitive

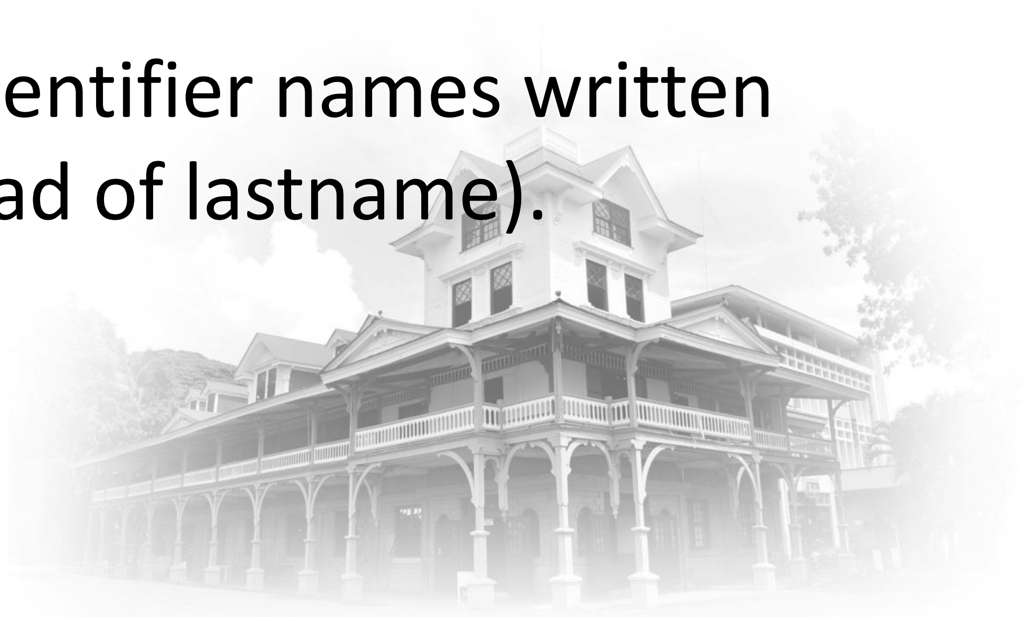
- In JavaScript all identifiers are case sensitive.
- The variables lastName and lastname, are two different variables.
- The functions myFunction and myfunction, are two different functions.
- JavaScript does not interpret Var; as var.

JS Character Set

- JavaScript uses the Unicode character set.
- Unicode covers (almost) all the characters, punctuations, and symbols in the world.
- For a closer look, please study our Complete Unicode Reference.

Did You Know?

- It is common, in JavaScript, to use camelCase names.
- You will often see identifier names written like lastName (instead of lastname).



Lab Activity 3

Write a JS Program that will ask the user to accept the radius of the circle then compute and display the diameter, area, and circumference.

I would like you to create a best UI using HTML and CSS.

*Save in your google drive and I will post a submission link within the week for you to submit.