Module Assignment 2

Lessons

String Properties and methods. Multiple if conditions and nested if statements

String Properties and methods

Previously we used length to determine the number of characters in a string variable.

var firstName = “Alexander”;

var lengthOfFirstName = firstName.length;

console.log(lengthOfFirstName);

// 8

Here length are the property of the firstName variable because firstName has a value of a string and by using the property length we find the number of characters of the string.

String values in javascript has both properties and methods, they are both called by using . after the variable then the name of the property or the methods. But methods are called by using () after the name of the method.

Two string method are toLowerCase() and toUpperCase(), toLowerCase() converts string variable to all lower cases and toUpperCase() converts all string variables to all upper cases.

var firstName = “Alexander”;

var lower = firstName.toLowerCase();

console.log(lower);

//alexander

var firstName = “Alexander”;

var upper = firstName.toUpperCase();

console.log(upper);

//ALEXANDER

In Javascript case for letters MATTER

Javascript consider upper and lower case version of a case to be two completely different things.

You can see here that “H” is not equal to “h”

If (“H” === “h”) {

console.log(“They are equal ”);

} else {

console.log(“They are not equal ”);

}

// “They are not equal ”

If you use the not equal operator, you can see they are not equal

If (“H” !== “h”) {

console.log(“They are not equal ”);

} else {

console.log(“They are equal”);

}

//They are not equal

Also longer strings the same rule apply for

If (“Harry” === “harry”) {

console.log(“They are equal”);

} else {

console.log(“They are not equal”);

}

//They are not equal

The logical AND (&&) operator

If we want to check and see if more than one condition is true we use the and (&&) operator. There is no and operator so we just use &&

This code check to see if the selectedNumber is between 0 and 45

var selectedNumber = 45;

if (selectedNumber <= 45 && selectedNumber >= 0) {

console.log(true);

} else {

console.log(false);

}

You can also use extra brackets around each condition like this

If ((selectedNumber <= 45) && (selectedNumber >= 0))

Nested if statements

We use nested if statements if we want to check for more specific conditions

if ((someStatment === true) && (someOtherStatment === false)) {

if (someCondition === true)

console.log(“One is true and the other is false. ”) {

//Do something

}

If (someOtherStatment === true) {

Console.log(“This statement is true”);

//Do something

}

}

The logical NOT ( ! ) operator

The ! operator is used to check for the opposite Boolean value.

var orderShipped = false;

if (!orderShipped) {

console.log(“Order not shipped”);

//Order not shipped

}

This code is the same as writing this

If (orderShippped === false) {

Console.log(“Order not shipped”);

//Order not shipped

}

Or this code

If (orderShipped !== true) {

Console.log(“Order not shipped”);

//Order not shipped

}

Logical OR ( || ) Operator

We use the && operator to check to see if a condition is true, here all conditions must be true

We use the || operator to see if only one of the conditions passes, when using || operator only one of the conditions needs to pass

var dayOfTheWeek = “Saturday”;

if ((dayOfTheWeek === “Saturday”) || (dayOfTheWeek === “Sunday”)) {

console.log(“Its weekend”);

//Its weekend

}