**Interactive Development with JavaScript – Module 5**

**if, if/else, and if/else/if**

Selection statements in JavaScript, as in other languages, are used to allow and or prevent execution based on a condition.

* if
  + A statement or block of code executes if the tested condition is true.
  + A statement or block is skipped if the tested condition is false.

Examples:

if (payment == debt){

outputStatement1 = "Paid in full.";

balance = 0;

}

if (payment == debt)

balance = 0;

* if/else
  + A statement or block of code in the if executes if the test condition is true.
  + A statement or block of code in the else executes if the test condition is false.

Examples:

if (hoursWorked <= 40){

// Calculate normal salary

}

else{

// Calculate salary + overtime

}

if (hoursWorked <= 40)

// Calculate normal salary

else

// Calculate salary + overtime

* if/else/if – or sometimes referred to as else/if
  + Used to add another condition to be tested if the first condition is false.

Example 1:

if (condition1) { // condition1 tested

// executed if condition1 true

} else if (condition2) { // tested if condition1

false

// executed if condition2 true

} else {

// executed if condition2 false

}

Example 2:

if (condition1)// condition1 tested

// executed if condition1 true

else if (condition2)// tested if condition1 false

// executed if condition2 true

else

// executed if condition2 false

**switch**

* A switch is similar to a collection of if statements allowing several different blocks of code to be executed based on a value compared with several different options.
* A single switch statement can have several case blocks.
* A switch value is checked for equality (compared) to the case values starting with the first case and continuing until a match is found. If a match is found the case block of code is executed until a break or the end of the switch is encountered.
* If no case matches are found a default block of code will execute. Note that the default block of code is optional. If a default block of code is not present, then no code will execute if no match is found.

Several case values can be grouped for a single block of code.

You may also use a switch to evaluate Strings.

A switch statement may also use an expression such as switch (varA + varB).

When a break is reached, the entire switch is exited. A break is optional.

The curly braces "{ " and "} " are required for the switch statements, however they are optional for each case.

Example 1:

switch(expression) {

case 1:

// Executed Code

break;

case 2:

// Executed Code

break;

…

default:

// code block

}

Example 2:

switch(expression) {

case 1:

// Executed Code

case 2:

// Executed Code

…

default:

// Executed Code

}

Example 3:

switch(expression) {

case 1:

case 2:

case 3:

// Executed Code

break;

case 4:

case 5:

case 6:

// Executed Code

break;

…

default:

// Executed Code

}

Example 4:

var currentDay = new Date();

switch (currentDay.getDay()){

case 1:

document.write("Monday");

break;

case 2:

document.write("Tuesday");

…

default:

document.write("Another Day");

}

**ternary**

* A conditional operator.
* Assigns/returns a value based on a condition.
* The only JavaScript operator that takes three operands.
* The logic of a Ternary is equivalent to that of an if/else statement.
* Syntax:

condition ? resultIfTrue : resultIfFalse

Example 1:

var scoreOne = 70;

var scoreTwo = 85;

var bestScore1 = (scoreOne > scoreTwo) ? scoreOne

: scoreTwo;

document.write(bestScore1 + "<br />");

Example 2:

var scoreThree = 90;

var scoreFour = 65;

document.write(((scoreThree > scoreFour) ?

scoreThree : scoreFour) + "<br />");

**Flowcharts**Diagram

Description automatically generated