JavaScript Condition

Department of Information system, KICT, IIUM
Dr. Najhan M.Ibrahim

- Conditional statements are used to perform different actions based on different conditions.

- Very often when you write code, you want to perform different actions for different decisions.

- Use if to specify a block of code to be executed, if a specified condition is true
- Use else to specify a block of code to be executed, if the same condition is false
- Use else if to specify a new condition to test, if the first condition is false
- Use switch to specify many alternative blocks of code to be executed

The if Statement

- Use the if statement to specify a block of JavaScript code to be executed if a condition is true.

```
if (condition) {
  //  block of code to be executed if the condition is true
}
```

```
if (hour < 18) {
  greeting = "Good day";
}
```

The else Statement

- Use the else statement to specify a block of code to be executed if the condition is false.

```
if (condition) {
  //  block of code to be executed if the condition is true
} else {
  //  block of code to be executed if the condition is false
}
```

```
if (hour < 18) {
  greeting = "Good day";
}
else {
  greeting = "Good evening";
}
```

The else if Statement

- Use the else if statement to specify a new condition if the first condition is false.

```
if (condition1) {
  //  block of code to be executed if condition1 is true
} else if (condition2) {
  //  block of code to be executed if the condition1 is false and condition2 is true
} else {
  //  block of code to be executed if the condition1 is false and condition2 is false
}
```

```
if (time < 10) {
  greeting = "Good morning";
} else if (time < 20) {
  greeting = "Good day";
}
else {
  greeting = "Good evening";
}
```

JavaScript Switch Statement

- The switch statement is used to perform different actions based on different conditions.

- switch(*expression*) {
  case *x*:
  // code block
  break;
-
  case *y*:
  // code block
  break;
-
  default:
  // code block
  }
- The switch expression is evaluated once.
- The value of the expression is compared with the values of each case.
- If there is a match, the associated block of code is executed.
- If there is no match, the default code block is executed.

Example:

- The getDay() method returns the weekday as a number between 0 and 6. (Sunday=0, Monday=1, Tuesday=2 ..)
- This example uses the weekday number to calculate the weekday name:

```
switch (new Date().getDay()) {
  case 0:
    day = "Sunday";
    break;
  case 1:
    day = "Monday";
    break;
  case 2:
     day = "Tuesday";
    break;
  case 3:
    day = "Wednesday";
    break;
  case 4:
    day = "Thursday";
    break;
  case 5:
    day = "Friday";
    break;
  case 6:
    day = "Saturday";
}
```

- Using when you want to run the same code over and over again, each time with a different value.
- Often use, when working with arrays:

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

  text += cars[i] + "<br>";
}
```

JavaScript supports different kinds of loops:

- for - loops through a block of code a number of times
- while - loops through a block of code while a specified condition is true
- do/while - also loops through a block of code while a specified condition is true

- The for loop has the following syntax:

```
for (statement 1; statement 2; statement 3) {
  // code block to be executed
}
```

- **Statement 1** is executed (one time) before the execution of the code block.
- **Statement 2** defines the condition for executing the code block.
- **Statement 3** is executed (every time) after the code block has been executed.

Example:

```
for (let i = 0; i < 5; i++) {
  text += "The number is " + i + "<br>";
}
```

- From the example above, you can read:
- Statement 1 sets a variable before the loop starts (let i = 0).
- Statement 2 defines the condition for the loop to run (i must be less than 5).
- Statement 3 increases a value (i++) each time the code block in the loop has been executed.

- The while loop, loops through a block of code as long as a specified condition is true.

- Syntax:

```
while (condition) {
  // code block to be executed
}
```

Example:

```
Let i =0;
while (i < 10) {

  text += "The number is " + i;
  i++;
}
```

- The do while loop is a variant of the while loop.
- This loop will execute the code block once, before checking if the condition is true, then it will repeat the loop as long as the condition is true.

- **Syntax**

```
do {
  // code block to be executed
}
while (condition);
```

Example;

```
let text = ""
let i = 0;
Do {

  text += "The number is " + i;
  i++;

}
while (i < 10);
```


Thank You
Asante
Merci
Arigato
Matondo
Spasibo
Salamat
Multumesc
Merci
Thank You