Control Statement

Conditional statements

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

1.if statement-use this statement to execute some code only if a specified condition is true

Syntax:

```
if (condition){
code to be executed if condition is true
}
```

2.if...else statement-use this statement to execute some code if the condition is true and another code if the condition is false

Conditional statements

- Syntax for if else:

```
 if (expression){
Statement(s) to be executed if expression is
true
}
else{  Statement(
s) to be            executed if expression is
false
}
```

Conditional statements

3. if..else if statement

- is the one level advance form of control statement that allows JavaScript to make correct decision out of several conditions

```
if (expression 1){
Statement(1)
}else if (expression 2){
 Statement(2)
}else if (expression 3){
Statement(3)
}else{ Statement(4) }
```

Switch Statement

- use this statement to select one of many blocks of code to be executed

```
switch (expression)
{
  case condition 1: statement(s)
                    break;
  case condition 2: statement(s)
                    break;
   ...
  case condition n: statement(s)
                    break;
  default: statement(s)
}
```

for Loop statement

- The **for** loop is the most compact form of looping and includes the following three important parts:
  - The **loop initialization** where we initialize our counter to a starting value. The initialization statement is executed before the loop begins.
  - The **test statement** which will test if the given condition is true or not. If condition is true then code given inside the loop will be executed otherwise loop will come out.
  - The **iteration** statement where you can increase or decrease your counter.
- You can put all the three parts in a single line separated by a semicolon.

for Loop statement

- ❑ Syntax

```
for (initialization; test condition; iteration
statement){
     Statement(s) to be executed if test
condition is true
}
```

while loop

- The while loop loops through a block of code as long as a specified condition is true.
- Syntax:
  - while (condition){

    //code to be executed

    }

do-while Looping statement

- ❑The **do...while** loop is similar to the **while** loop except that the condition check happens at the end of the loop. This means that the loop will always be executed at least once, even if the condition is *false*.

Syntax:

```
do{
   Statement(s) to be executed;
} while (expression);
```

Jump statement

1. **Break-**The break statement can also be used to jump out of a loop.
   - ❑The **break statement** breaks the loop and continues executing the code after the loop (if any).

2. **Continue** –This statement "jumps over" one iteration in the loop.
   - ❑The **continue statement** breaks one iteration (in the loop), if a specified condition occurs, and continues with the next iteration in the loop.

Thank you.