JavaScript

2. 함수와 흐름 제어문

- Function
- Flow Control Statements

JavaScript

FUNCTION

함수

- 함수는 복잡한 계산을 하거나, 자주 쓰는 루틴을 정형화할 때 쓰인다.
- 한번 선언을 하면 여러 코드에서 사용이 가능하기 때문에 효율적이다.
- 코드 상의 함수는 호출되기 전에는 수행되지 않는다.
- 함수의 호출은 코드 상에서 직접 호출할 수도 있고, 이벤트에 의해 호출될 수도 있다.

함수

```
<script>
    var tVal = 0;
    function addValue (val)
    {
        tVal += val;
        document.write (tVal);
    }
</script>

<input type=button value=10 onClick="addValue(10)">
<script>
    addValue (9);
</script>
```

함수

- 함수는 아래와 같이 선언된다.
  - function은 키워드(예약어)로 반드시 함수 명 앞에 써주어야 한다.

```
function name(parameter1, parameter2, parameter3) {
    code to be executed
}
```

- 함수는 하나 이상의 인자를 포함할 수도 있고, 전혀 포함하지 않을 수도 있다.

```
function myFuncFirst (param1, param2, param3)
function myFuncSecond()
```

- 함수를 호출할 때 인자 값을 주고, 이를 함수는 변수로 받아 함수 내에서 사용한다.

매개변수가 없는 함수

- 매개 변수가 없는 함수를 호출할 때는 값을 넘겨 주지 않는다.

```
<html>
    <head>
    </head>
    <body>
        <script>
            function chgBgColor()
            {
                document.bgColor = "blue";
            }
        </script>
        <input type="button" value="COLOR" name="test" onClick="chgBgColor()">
    </body>
</html>
```

매개변수가 있는 함수

- 매개 변수가 있는 함수를 호출할 때는 인자 값을 넘겨 준다.

```
<html>
    <head>
    </head>
    <body>
        <script>
            function chgBgColor(bgc)
            {
                document.bgColor = bgc;
            }
        </script>
        <input type="button" value="COLOR" onclick="chgBgColor('red')">
    </body>
</html>
```

함수와 변수

- 전역변수
  - 어느 곳에서도 사용 가능함.
- 지역변수
  - 선언된 함수 내부에서만 사용 가능함.
- 함수 내의 전역변수
  - 함수 내에서 var 를 붙이지 않고 변수를 사용했다면, 이 변수는 전역 변수가 된다.
  - 함수가 한번 호출되어야 다른 곳에서 전역 변수를 사용할 수 있다.

함수와 변수

```
<html>
    <head>
    </head>
    <body>
        <script>
            var glbVar = "initial";
            function hello()
            {
                var lclVar = "Local";
                lclGlbVar = "Local to be Global";
                glbVar = "Global";
            }
            hello();
            document.write (lclVar + "<br>");
            document.write (lclGlbVar + "<br>");
            document.write (glbVar + "<br>");

        </script>
    </body>
</html>
```

JavaScript

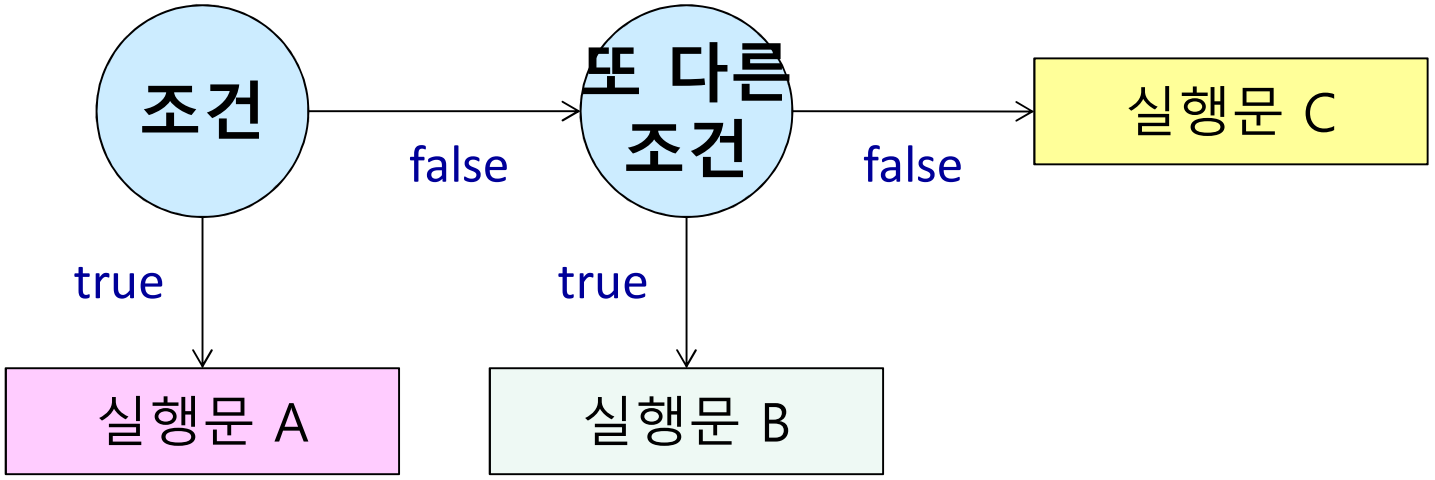
FLOW CONTROL STATMENTS

제어문

- 조건문
  - if문과 switch 문
  - 조건에 따라 특정 부분을 수행하거나 수행하지 않도록 하게 할 수 있다.
- 반복문
  - while문과 for문
  - 특정 조건이 만족하는 동안 또는 만족할 때까지 반복적으로 동작을 수행하도록 한다.

조건문

- if 문
  - 조건이 참인 경우에 명령을 수행한다. 그렇지 않은 경우의 명령도 지정할 수 있다.



조건문

- if 문

```
if (condition1)
{
    process 1
}
else if (condition2)
{
    process 2
}
else
{
    process 3
}
```

```
(condition1)?process 1: (condition2)?process2:process3
```

조건문

- switch 문
  - if 문은 각 조건이 서로 독립적일 수 있으나 switch 문은 하나의 비교 대상에 한정된다.

```
switch (variable)
{
    case value1:
        process1
        break;
    case value2:
        process2
        break;
    default:
        process4
        break;
}
```

반복문

- while 문과 do while 문
  - 상태가 true인 동안 계속 반복 수행
  - while 문은 한번도 수행되지 않을 수 있으며, do while 문은 적어도 한번은 수행된다.

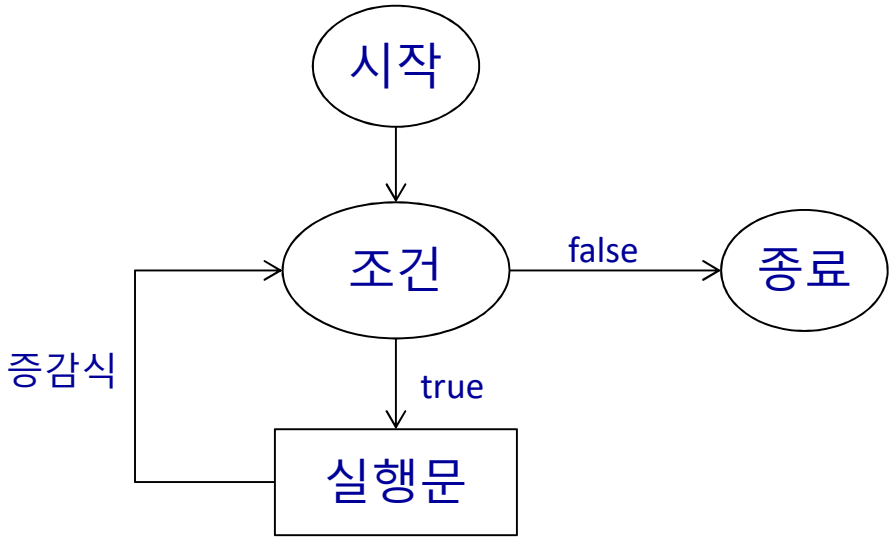
```
while (expression)
{
    process
}

do {
    process
} while (expression)
```

반복문

- for 문
  - 조건에 만족하는 동안 지정한 횟수만큼 명령문을 반복 수행한다.

```
for (초기값; 조건; 증감식)
{
    process
}
```



반복문

- break와 continue
  - for 문이나 while 문 안에서 사용된다.
  - break문을 사용해서 반복문을 빠져 나오게 할 수 있다.
  - continue문을 사용하면 특정조건에서 반복문 내의 명령을 생략할 수 있다.

```
for (초기값; 조건; 증감식)
{
    if (condition)
        continue;

    if (condition)
        break;
    process
}
```

```
while (expression)
{
    if (condition)
        continue;
    if (condition)
        break;
    process
}
```

SIMPLE HTML DOM

HTML Element 찾기

How	Method
아이디를 이용하여 찾기	document.getElementByid(id)
Tag 이름으로 선택	document.getElementsByTagName(tag)
name 속성 값으로 선택	document.getElementsByName(name)
Class 속성 값으로 선택	document.getElementsByClassName(name)

내용 바꾸기

How	Method
문서 내용 바꾸기	document.write()
HTML content 바꾸기	innerHTML을 사용함. document.getElementById(id).innerHTML
HTML 속성 값 바꾸기	document.getElementById(id).속성이름
CSS 스타일 값 바꾸기	document.getElementById(id).style.스타일속성

END