1장 Javascript 개요

1. Javascript 정의

Javascript는 개체 기반 스크립팅 언어이다.

구조적 문법 표현을 지원한다.

클래스 대신에 프로토타입을 사용한다. 생성자를 만들어 객체를 정의할 수 있다.

변수의 데이터 형식을 명시적으로 선언하지 않는다. 자동으로 형변환을 수행한다.

2. Javascript의 기본

1) 기본 페이지 템플릿 준비

① HTML5 사용

HTML의 버전을 구별하지 않고 동일한 기능을 제공한다.

② DTD 선언문의 변경

<!DOCTYPE html>

③ 파일의 인코딩 방식의 지정

<meta charset="utf-8" />

④ 모바일 웹을 위한 설정

<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0, user-scalable=no"/>

⑤ 브라우저 호환성 설정

브라우저의 호환성 보기 모드(css의 margin 태그에 대한 겹침 현상 등으로 크로스 브라우징 처리의 방해 요소)를 막고 해당 브라우저에서 지원하는 가장 최신 버전의 방식으로

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />

⑥ IE8 이하 버전에서 HTML5의 새로운 태그 인식

https://github.com/afarkas/html5shiv 에서 다운로드하여 압축 해제, html5shiv.js 파일 사용

<!--[if lt IE 9]>

<script src="js/html5shiv.js"></script>

<![endif]-->

⑦ 모바일 단말기에서 모바일 웹의 단축 아이콘 등록

```
<!-- 모바일 웹 페이지 설정 -->
        <link rel="shortcut icon" href="img/icon.png" />        <!-- iSO -->
        <link rel="apple-touch-icon" href="img/icon.png" />   <!-- Android -->
<!-- 모바일 웹 페이지 설정 끝 -->
```

⑧ 기본 템플릿 페이지 소스

default.html

```
<!DOCTYPE html>
<html lang="ko">
        <head>
                <meta charset="utf-8" />
                <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0, user-scalable=no"/>
                <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
                <title>Hello Javascript</title>

                <!-- 모바일 웹 페이지 설정 -->
                <link rel="shortcut icon" href="img/icon.png" />
                <link rel="apple-touch-icon" href="img/icon.png" />
                <!-- 모바일 웹 페이지 설정 끝 -->

                <!--[if lt IE 9]>
                <script type="text/javascript" src="js/html5shiv.js"></script>
                <![endif]-->
        </head>
        <body>
                <h1>Hello Javascript</h1>
        </body>
</html>
```

2) HTML 페이지에 Javascript 포함시키는 방법

① HTML 문서 안에 포함시키는 방법

<head></head> 태그안에 포함

helloJavascript1.html

```
<!DOCTYPE html>
<html lang="ko">
	<head>
		<meta charset="utf-8" />
		<title>Hello Javascript</title>

		<!-- javascript start -->
		<script type="text/javascript">
			alert("안녕하세요. Javascript~!!!");
		</script>
		<!-- javascript end -->
	</head>
	<body>
		<h1>Hello Javascript</h1>
	</body>
</html>
```

② 외부 *.js 파일 참조하기

```
<script type="text/javascript" src="js 파일의 경로"></script>
```

js₩sayHello.js

```
alert("안녕하세요~ Javascript");
```

helloJavascript2.html

```
<!DOCTYPE html>
<html lang="ko">
	<head>
		<meta charset="utf-8" />
		<title>Hello Javascript</title>
		<!-- javascript start -->
```

```
                <script type="text/javascript" src="js/sayHello.js"></script>
                <!-- javascript end -->
        </head>
        <body>
                <h1>Hello Javascript</h1>
        </body>
</html>
```

3) Javascript 코드에 주석문

① 한 줄 주석문

```
<script type="text/javascript">
        // 한 줄 주석문을 설정한다.
</script>
```

② 여러 줄 주석문

```
<script type="text/javascript">
        /*
          이 블록 안에 여러 줄을 주석으로 사용
          ....
        */
</script>
```

4) 결과 값을 출력하기 위한 console의 사용

console 이란 웹 브라우저 화면에 표시되는 것이 아니라 필요한 내용 출력하기 위한 영역이다. 프로그램 중간 단계의 변화 상태를 출력하여 디버그용으로도 사용한다.

① console 사용

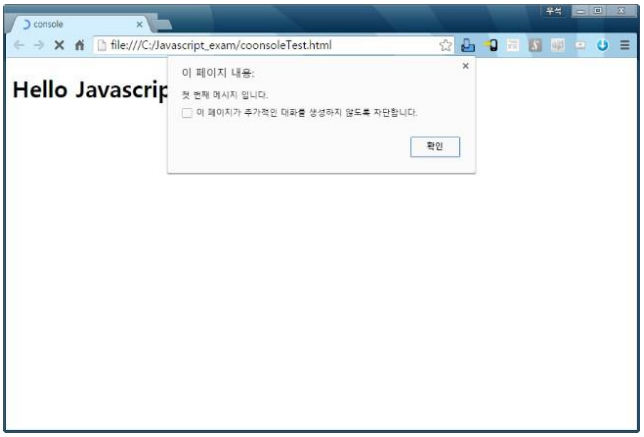
```
 console.log("출력할 내용");
```

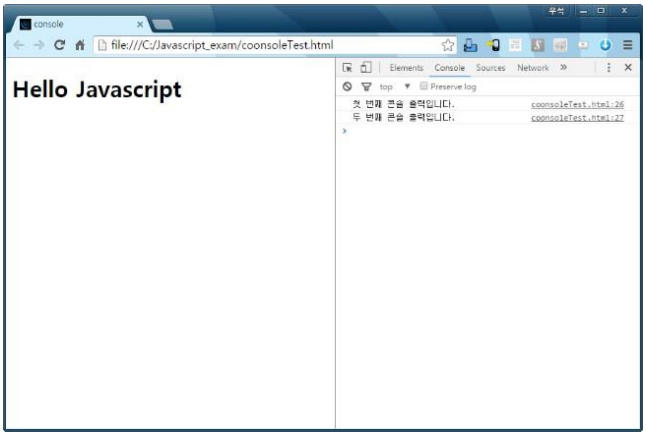
consoleTest.html

```
<!DOCTYPE html>
<html lang="ko">
        <head>
                <meta charset="utf-8" />
                <title>Console Test</title>
                <!-- javascript start -->
```

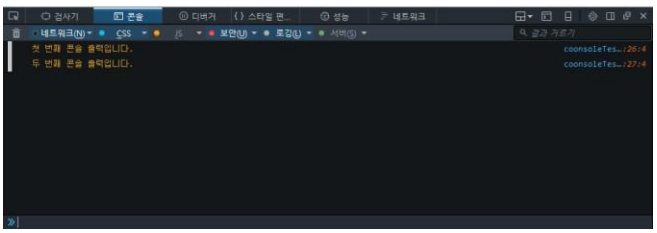
```
                <script type="text/javascript">
                        /* alert 대화상자는 순서대로 메시지 창이 표시되기 때문에,
                            결과에 대한 일괄 확인은 어렵다. */
                        alert("첫 번째 메시지 입니다.");
                        alert("두 번째 메시지 입니다.");

                        // console의 경우에는 실행 결과를 일괄적으로 확인 할 수 있다.
                        console.log("첫 번째 콘솔 출력입니다.");
                        console.log("두 번째 콘솔 출력입니다.");
                </script>
                <!-- javascript end -->
        </head>
        <body>
                <h1>Hello Javascript</h1>
        </body>
</html>
```

② 결과 확인





Hello Javascript



웹 브라우저에서 **F12** 키를 누르면 **개발자 모드** 영역이 나타나고 console의 내용이 출력된다.

2장 Javascript 시작

1. 변수의 이해와 데이터형의 종류

1) 변수의 사용 방법

변수를 사용하기 위해서는 선언과 할당이 필요하다. 변수의 선언은 값이 저장되기 위한 공간을 메모리 상에 마련하는 것이고, 변수의 할당은 실제 메모리에 어떤 값을 저장시키는 것을 의미한다.

① 변수의 선언

변수의 선언은 "**var**" 키워드 뒤에 사용하고자 하는 변수의 이름을 지정하고, 세이콜론(;)으로 한 줄 명령문을 종료한다.

```
var name;
var number;
```

② 변수의 할당

변수의 할당은 선언된 변수에 원하는 값을 대입하는 과정을 의미한다. 숫자나 문자열 등을 사용할 수 있다.

```
name = "홍길동";
number = 78;
```

③ 선언과 할당을 통합

선언과 할당을 할 줄로 표현할 수 있다.

```
var name = "홍길동";
var number = 78;
```

④ 변수 네이밍 규칙

변수의 이름은 영문자, 숫자, _ 만을 사용할 수 있으며, 첫 글자는 숫자로 시작할 수 없다. 대소문자를 엄격히 구별한다. 두 단어 이상일때는 두번째 단어의 첫 문자를 대문자로 한다.

```
myName, my_phone, userInput, userPassword
```

⑤ 변수 선언의 제약

변수에는 다른 값을 할당할 수 있으나 이름이 중복되어 선언할 수 없다.

```
var number = 2345;
number = 987;
var number = 325;   // 에러 발생이 안된다.
```

선언되지 않은 변수는 사용할 수 없다.

```
var number;
number = 3425;
```

```
myNum = 456;  // 에러 발생
```

2) 변수의 종류

변수는 할당되는 값에 따라 그 종류가 결정된다. 이것을 변수의 '데이터 형(Data Type)'이라 한다.

데이터 타입	설 명
Number(정수, 실수)	숫자나 산술 연산에 사용되는 데이터 타입.
String(문자열)	쌍따옴표나 홑따옴표로 감싸진 문자
Boolean(논리)	true, false 값을 가지는 논리형 데이터 타입.
Object(객체)	객체를 저장하기 위한 데이터 타입.
null(참조 값이 없음)	객체가 다른 곳을 참조되지 않았을 때의 값, 즉 비어 있는 객체 상태
undefined(정의되지 않음)	변수가 선언되고 할당되지 않았을 때의 표시 값.

valueTest.html

```
<!DOCTYPE html>
<html lang="ko">
	<head>
		<meta charset="utf-8" />
		<title>Value Test</title>
		<script type="text/javascript">
			/*
				이 블록 안에서는 여러 줄을 주석으로 할 수 있습니다.
				주로 여러 줄의 코드를 주석으로 막아놓고자 하는 경우 사용됩니다.
			 */

			// 변수의 선언
			var num1;
			var msg1;

			// 변수의 할당
			num1 = 12345;
			msg1 = "Hello Javascript";

			// 변수의 선언과 할당
			var num2 = 3.14;
			var msg2 = "안녕하세요. 자바스크립트";

			// 브라우저의 콘솔 영역에 출력하기
```

```
                        console.log(num1);
                        console.log(msg1);
                        console.log(num2);
                        console.log(msg2);

                        // boolean값의 사용
                        var isMan = true;
                        console.log(isMan);

                        // null값 - 의도적으로 null을 대입
                        var value1 = null;
                        console.log(value1);

                        // undefined - 선언만 된 상태
                        var value2;
                        console.log(value2);
                </script>
        </head>
        <body>
                <h1>Value Test</h1>
        </body>
</html>
```

자바스크립트에서 변수 선언을 하는 3가지 방법(var, let, const)이 있다.

var

var 같은 경우에는 중복 선언이 가능하다.

var start='안녕하세요';

var start='시작입니다';

이렇게 start라는 변수가 둘 다 선언이 되고 에러가 발생하지 않는다.

둘 다 위치에 따라 각 다른 값이 출력되지만 코드가 길어진다면 어디에서 어떻게 사용될지 파악하기 힘들다.

let

let 같은 경우에는 중복 선언이 불가능하다.

let start='안녕하세요';

let start='시작입니다';

이렇게 선언하게 되면 중복이 되기 때문에 에러가 발생하여 변수 재선언이 되지 않는다.

그러나 let은 재할당이 가능하다. 무슨 말이냐면 let start='안녕하세요'; 으로 start 변수를 선언하고 나서 start='시작입니다'; 를 통해서 변수를 재할당할 수 있다. 그러면 위치에 따라서 변수를 재할당하여 사용할 수 있다.

const

const 가은 경우에도 중복 선언이 불가능하다.

const start='안녕하세요';

const start='시작입니다';

이렇게 선언하게 되면 중복이 되기 때문에 에러가 발생하여 변수 재선언이 되지 않는다.

그리고 const는 재할당도 불가능하다.

const start='안녕하세요'; 으로 start 변수를 선언하고 나서 start='시작입니다';를 하더라도 에러가 발생한다.

이런 식으로 var, let, const는 각각 차이점이 있다.

보통 기본적으로 안전한 const를 사용한다.

가끔 재할당이 필요한 경우에는 let을 사용하기도 한다.

const를 사용하면 코드가 길어져서 복잡해졌을 때 중복과 재할당이 되지 않기 때문에 안전하게 사용할 수 있다.

2. 연산자

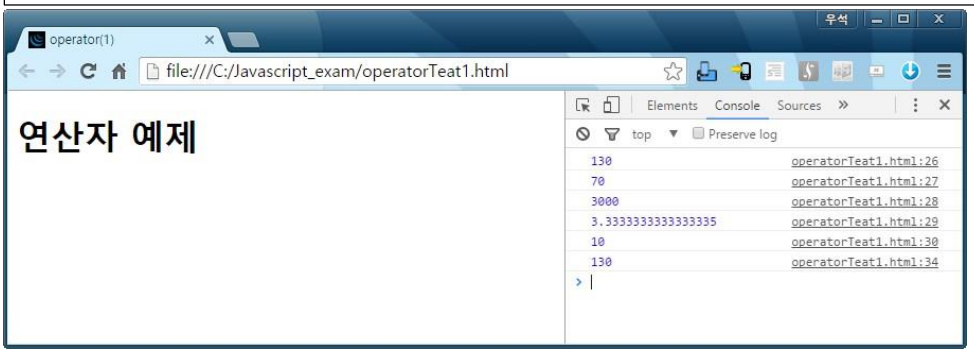
연산자	설 명
대입 연산자	변수에 숫자나 문자열 등의 데이터를 할당한다.
사칙 연산자	+, -, *, /, %(나머지)
단항 연산자	연산 결과를 자기 자신에게 적용한다.
증감 연산자	특정 변수의 값을 1씩 증가, 감소시킨다.
비교 연산자	두 개의 변수 값의 관계를 비교하여 true, false의 결과를 반환한다.
논리 연산자	좌변과 우변의 논리 연산한다. &&, ||

1) 대입, 사칙 연산자

operatorTeat1.html

```
<!DOCTYPE html>
<html lang="ko">
	<head>
		<meta charset="utf-8" />
		<title>Operator Test</title>
		<script type="text/javascript">
			// 대입, 사칙연산
			var num1 = 100 + 30;
			var num2 = 100 - 30;
			var num3 = 100 * 30;
			var num4 = 100 / 30;
			var num5 = 100 % 30;

			console.log(num1);
			console.log(num2);
			console.log(num3);
			console.log(num4);
			console.log(num5);
			// 연산자 우선순위
			var num6 = 100 + 30 * (2 - 1);
			console.log(num6);
		</script>
	</head>
	<body>
		<h1>연산자 예제</h1>
	</body>
```

```
</html>
```



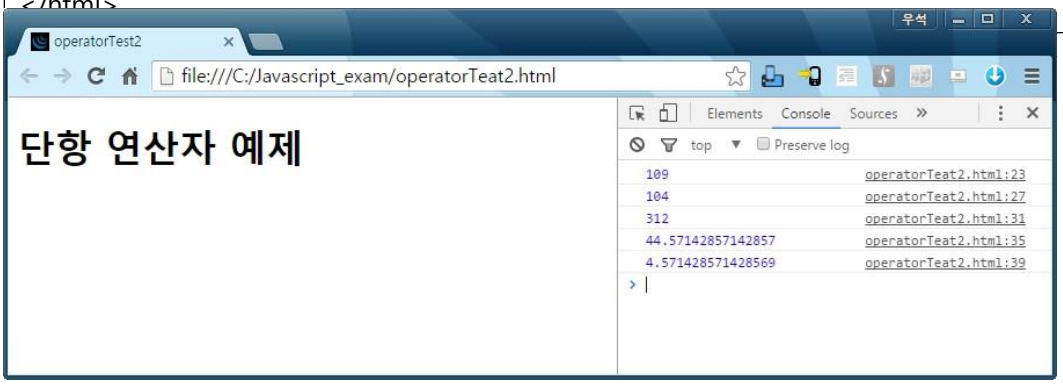
2) 단항 연산자

사칙 연산	단항 연산자	적용
y = y + 3	+=	y += 3
y = y – 3	-=	y -= 3
y = y * 3	*=	y *= 3
x = y / 3	/=	y /= 3
y = y % 3	%=	y %= 3

operatorTeat2.html

```
<!DOCTYPE html>
<html lang="ko">
	<head>
		<meta charset="utf-8" />
		<title>Operator Test</title>
		<script type="text/javascript">
			var num = 100;

			// 덧샘에 대한 단항 연산자
			num += 9;
			console.log(num);

			// 뺄셈에 대한 단항 연산자
			num -= 5;
			console.log(num);

			// 곱셈에 대한 단항 연산자
			num *= 3;
			console.log(num);
```

```
                        // 나눗셈에 대한 단항 연산자
                        num /= 7;
                        console.log(num);

                        // 나머지에 대한 단항 연산자
                        num %= 5;
                        console.log(num);
                </script>
        </head>
        <body>
                <h1>단항 연산자 예제</h1>
        </body>
</html>
```



3) 증감 연산자(++, --)

덧셈과 뺄셈에 대한 표현식으로만 축약해서 사용한다.

operatorTeat3.html

```
<!DOCTYPE html>
<html lang="ko">
        <head>
                <meta charset="utf-8" />
                <title>Operator Test</title>
                <script type="text/javascript">
                        var plus_num = 1;
                        plus_num = plus_num + 1;
                        plus_num += 1;
                        plus_num++;
                        ++plus_num;
```

```
                        // plus_num의 결과는?
                        console.log(plus_num);

                        var minus_num = 4;
                        minus_num = minus_num - 1;
                        minus_num -= 1;
                        minus_num--;
                        --minus_num;

                        // minus_num의 결과는?
                        console.log(minus_num);
                </script>
        </head>
        <body>
                <h1>증감 연산자 예제</h1>
        </body>
</html>
```

① ++x 와 x++ 의 차이점

x++와 같이 증가 연산자를 변수 뒤에 사용하는 경우는 x의 값을 먼저 수식에 적용하고 x에 대한 증가를 나중에 처리한다.

```
var z = 100;
var x = 1;
var y = z + x++;
```

y의 값은 101이고 x의 값은 2가 된다.

x++와 같이 증가 연산자를 변수 앞에 사용하는 경우는 x의 값을 먼저 증가하고 그 결과를 수식에 적용한다.

```
var z = 100;
var x = 1;
var y = z + ++x;
```

수식보다 먼저 x의 값을 먼저 증가시켜 x는 2가 되고 y는 102가 된다.

operatorTeat4.html

```
<!DOCTYPE html>
<html lang="ko">
	<head>
		<meta charset="utf-8" />
		<title>Operator Test</title>
		<script type="text/javascript">
			var z1 = 100;
			var x1 = 1;
			var y1 = z1 + x1++;

			// x1값과 y1값의 결과 확인
			console.log(x1);
			console.log(y1);

			var z2 = 100;
			var x2 = 1;
			var y2 = z2 + ++x2;

			// x2값과 y2값의 결과 확인
			console.log(x2);
			console.log(y2);
		</script>
	</head>
	<body>
		<h1>증감 연산자 비교</h1>
	</body>
</html>
```

4) 문자열 연산

① 문자열과 문자열

문자열 연산은 +만 가능하며 문자열을 더한다는 것는 문자열을 연결한다는 의미이다.

② 문자열과 숫자값

문자열과 숫자값이 더해질 경우, 숫자값이 문자열 형태로 변환된다.

operatorTeat5.html

```
<!DOCTYPE html>
<html lang="ko">
	<head>
		<meta charset="utf-8" />
		<title>Operator Test</title>
		<script type="text/javascript">
			// 문자열끼리의 연산결과 확인
			var msg1 = "안녕하세요.";
			var msg2 = "자바스크립트";
			var msg3 = msg1 + " " + msg2;
			console.log(msg3);

			// 문자열과 숫자값의 연산결과 확인
			var msg = "안녕하세요.";
			var num = 1234;
			var result = msg + num;
			console.log(result);
		</script>
	</head>
	<body>
		<h1>문자열 연산 예제</h1>
	</body>
</html>
```

3장 프로그램의 흐름 제어

1. 조건에 만족하는 경우만 한번 실행

1) 조건문의 종류

조건문	설 명
if문	주어진 조건이 true이면 실행한다.
if ~ else 문	주어진 조건이 true이면 if문이 실행하고 false이면 else문이 실행된다.
if ~ else if ~ else 문	여러 개의 조건문으로 세분화 시켜서 사용되는 다중 if문
switch 문	하나의 값에 대하여 해당되는 경우에만 분기하여 실행된다.

① if 문

```
if ( 조건 ) {
    실행문;
}
```

조건에 사용되는 연산자

==, !=, <, >, <=, >=, &&, ||, true, false, !

if_test1.html

```
<!DOCTYPE html>
<html lang="ko">
	<head>
		<meta charset="utf-8" />
		<title>If Test</title>
		<script type="text/javascript">
			var myage = 19;

			// 같다면 참
			if (myage == 19) {
				document.write("<h1>19살 입니다.</h1>");
			}

			// 다르다면 참
			if (myage != 20) {
				document.write("<h1>20살이 아닙니다.</h1>");
			}
```

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

if_test2.html

```
<!DOCTYPE html>
<html lang="ko">
	<head>
		<meta charset="utf-8" />
		<title>If Test</title>
		<script type="text/javascript">
			var myage = 19;

			if (myage > 19) {
				document.write("<h1>성인입니다.</h1>");
			}

			if (myage <= 19) {
				document.write("<h1>학생입니다.</h1>");
			}
		</script>
	</head>
	<body>
	</body>
</html>
```

if_test3.html

```
<!DOCTYPE html>
<html lang="ko">
	<head>
		<meta charset="utf-8" />
		<title>If Test</title>
		<script type="text/javascript">
			var point = 75;

			// AND(&&) --> 두 조건이 모두 참이어야 전체가 참, 범위를 검사한다.
```

```
                        if (point > 70 && point <= 80)
                                { document.write("<h1>C학점 입니다.</h1>");
                        }

                        // OR(||) --> 두 조건중 하나라도 참이면 전체가 참
                        if (point <= 70 || point > 80) {
                                document.write("<h1>범위를 벗어났습니다.</h1>");
                        }
                </script>
        </head>
        <body>
        </body>
</html>
```

boolean형 변수의 사용

if_test4.html

```
<!DOCTYPE html>
<html lang="ko">
        <head>
                <meta charset="utf-8" />
                <title>If Test</title>
                <script type="text/javascript">
                        var is_korean = true;

                        if (is_korean == true) {
                                document.write("<h1>한국사람입니다.</h1>");
                        }

                        if (is_korean == false) {
                                document.write("<h1>한국사람이 아닙니다.</h1>");
                        }

                        if (is_korean) {     // 값 자체가 참이므로 성립된다.
                                document.write("<h1>한국사람입니다.</h1>");
                        }

                        if (!is_korean) {    // "!"는 값을 부정한다. 참을 부정하므로 거짓이다.
                                document.write("<h1>한국사람이 아닙니다.</h1>");
```

```
                        }

                        var is_japanese = false;

                        if (is_japanese != true) {
                                document.write("<h1>일본사람이 아닙니다.</h1>");
                        }

                        if (is_japanese != false) {
                                document.write("<h1>일본사람 입니다.</h1>");
                        }

                        if (is_japanese) { // 값 자체가 거짓이므로 성립되지 않는다.
                                document.write("<h1>일본사람이 아닙니다.</h1>");
                        }

                        if (!is_japanese) { // 거짓을 부정하였으므로 참이다.
                                document.write("<h1>일본사람 입니다.</h1>");
                        }
                </script>
        </head>
        <body>
        </body>
</html>
```

② if ~ else 문

```
if ( 조건 ) {
    true 일 때 실행문;
} else {
    false 일 때 실행문;
}
```

if_test5.html

```
<!DOCTYPE html>
<html lang="ko">
        <head>
                <meta charset="utf-8" />
                <title>If Test</title>
```

```
                <script type="text/javascript">
                        var myage = 19;

                        if (myage > 19) {
                                document.write("<h1>성인입니다.</h1>");
                        } else {
                                document.write("<h1>성인이 아닙니다.</h1>");
                        }
                </script>
        </head>
        <body>
        </body>
</html>
```

③ 다중 if문

```
if ( 조건1 ) {
    true 일 때 실행문;
} else if
    ( 조건2 ){ true 일
    때 실행문;
} else if
    ( 조건3 ){ true 일
    때 실행문;
} else {
```

if_test6.html

```
<!DOCTYPE html>
<html lang="ko">
        <head>
                <meta charset="utf-8" />
                <title>If Test</title>
                <script type="text/javascript">
                        var point = 82;
                        // 임의로 비워둔 값
                        var grade = null;

                         if (90 < point && point <= 100)
                                 { grade = "A";
```

```
                        } else if (80 < point && point <= 90)
                                { grade = "B";
                        } else if (70 < point && point <= 80)
                                { grade = "C";
                        } else {
                                grade = "F";
                        }

                        var msg = "<h1>" + grade + "학점 입니다.</h1>";
                        document.write(msg);
                </script>
        </head>
        <body>
        </body>
</html>
```

④ switch문

```
switch ( 변수명 )
    { case 값1:
        실행문;
        break;
    case 값2:
        실행문;
        break;
    default:
        그 밖의 값일 경우 실행문;
        break;
}
```

switch_test.html

```
<!DOCTYPE html>
<html lang="ko">
        <head>
                <meta charset="utf-8" />
                <title>Switch Test</title>
                <script type="text/javascript">
                        var grade = "B";
```

```
                        switch (grade) {
                                case "A":
                                        document.write("<h1>91~100점 사이 입니다.</h1>");
                                        break;
                                case "B":
                                        document.write("<h1>81~90점 사이 입니다.</h1>");
                                        break;
                                case "C":
                                        document.write("<h1>71~80점 사이 입니다.</h1>");
                                        break;
                                default:
                                        document.write("<h1>70점 이하 입니다.</h1>");
                                        break;
                        }
                </script>
        </head>
        <body>
        </body>
</html>
```

case절에서 break문을 생략할 경우에는 break문을 만나기 전까지 실행된다.

2. 반복적인 처리를 할 경우의 반복문

① for문

```
for( 초기값; 조건식; 증감식) {
    반복적으로 실행할 문장;
}
```

for_test1.html

```
<!DOCTYPE html>
<html lang="ko">
        <head>
                <meta charset="utf-8" />
                <title>For Test</title>
                <script type="text/javascript">
                        var sum = 0;
```

```
                        // i값이 1부터 100가지 증가하는 동안, i의 값을 sum에 누적한다.
                        for (var i=1; i<=100; i++) {
                                // 결과값 누적
                                sum += i;   // sum = sum + i;
                        }

                        document.write("<h1>1부터 100까지의 합은 " + sum + "입니다.</h1>");
                </script>
        </head>
        <body></body>
</html>
```

for_test2.html

```
<!DOCTYPE html>
<html lang="ko">
        <head>
                <meta charset="utf-8" />
                <title>For Test</title>
                <script type="text/javascript">
                        for (var i=1; i<10; i++) {
                                var j = 7 * i;
                                document.write("<p>7 x " + i + " = " + j + "<p>");
                        }
                </script>
        </head>
        <body></body>
</html>
```

② while문

```
while ( 조건 ) {
    조건이 true일 동안 실행할 문장;
}
```

while_test1.html

```
<!DOCTYPE html>
<html lang="ko">
```

```
	<head>
		<meta charset="utf-8" />
		<title>While Test</title>
		<script type="text/javascript">
			var sum = 0;

			var i = 1;   // 초기값
			while (i<=100) {
				sum += i;
				i++;
			}

			document.write("<h1>1부터 100까지의 합은 " + sum + "입니다.</h1>");
		</script>
	</head>
	<body></body>
</html>
```

while_test2.html

```
<!DOCTYPE html>
<html lang="ko">
	<head>
		<meta charset="utf-8" />
		<title>While Test</title>
		<script type="text/javascript">
			var i = 1;   // 초기값
			while(i<10) {
				var j = 7 * i;
				document.write("<p>7 x " + i + " = " + j + "<p>");
				i++;
			}
		</script>
	</head>
	<body></body>
</html>
```

③ do~while문

```
do {
```

```
    실행문;
} while ( 조건 );
```

실행문을 먼저 실행하고 조건이 true일 동안 반복한다.

while_test3.html

```
<!DOCTYPE html>
<html lang="ko">
	<head>
		<meta charset="utf-8" />
		<title>do while Test</title>
		<script type="text/javascript">
			var i = 1;   // 초기값
			do {
				var j = 7 * i;
				document.write("<p>7 x " + i + " = " + j + "<p>");
				i++;
			} while(i<10);
		</script>
	</head>
	<body></body>
</html>
```

3. 제어문의 중첩 사용

① if문의 중첩

```
if ( 조건 ) {
    if ( 조건 ) {
      true 일 때 실행문;
    }
}
```

```
if ( 조건 ) {
    if ( 조건 ) {
      true 일 때 실행문;
    } else {
      false 일 때 실행문;
    }
```

```
}
```

```
if ( 조건 ) {
    if ( 조건 ) {
      true 일 때 실행문;
    } else {
      false 일 때 실행문;
    }
} else {
   if ( 조건 ) {
      true 일 때 실행문;
    } else {
      false 일 때 실행문;
    }
}
```

multi-if-test.html

```
<!DOCTYPE html>
<html lang="ko">
        <head>
                <meta charset="utf-8" />
                <title>multi if Test</title>
                <script type="text/javascript">
                        var point = 78;

                         if (point > 70 && point <= 80)
                                { if (point > 77) {
                                        document.write("<h1>C+ 입니다.</h1>");
                                 } else if (point < 74) {
                                        document.write("<h1>C- 입니다.</h1>");
                                 } else {
                                        document.write("<h1>C0 입니다.</h1>");
                                 }
                        }
                </script>
        </head>
        <body>
        </body>
```

```
</html>
```

② 반복문 안의 조건문

```
for( 초기값; 조건식; 증감식)
    { if ( 조건 ) {
        true 일 때 실행문;
    } else {
        false 일 때 실행문;
    }
}
```

for-if-test.html

```
<!DOCTYPE html>
<html lang="ko">
    <head>
        <meta charset="utf-8" />
        <title>for if Test</title>
        <script type="text/javascript">
            var sum1 = 0;
            var sum2 = 0;
            for (var i=1; i<=100; i++) {
                // 2로 나눈 나머지가 0이면 짝수
                if (i % 2 == 0) {
                    sum1 += i;
                } else {
                    sum2 += i;
                }
            }
            document.write("<h1>짝수들의 합 = " + sum1 + "</h1>");
            document.write("<h1>홀수들의 합 = " + sum2 + "</h1>");
        </script>
    </head>
    <body>
    </body>
</html>
```

③ 반복문의 중첩

```
for( 초기값; 조건식; 증감식) {
    for( 초기값; 조건식; 증감식) {
        반복적으로 실행할 문장;
    }
}
```

multi-for-test1.html

```
<!DOCTYPE html>
<html lang="ko">
    <head>
        <meta charset="utf-8" />
        <title>multi for Test</title>
        <script type="text/javascript">
            for (var i=2; i<=9; i++) {
                document.write("<h1>" + i + "단</h1><ul>");

                for (var j=1; j<10; j++) {
                    var k = i*j;
                    var result = i + " x " + j + " = " + k;
                    document.write("<li>" + result + "</li>");
                }
                document.write("</ul>");
            }
        </script>
    </head>
    <body>
    </body>
</html>
```

multi-for-test2.html

```
<!DOCTYPE html>
<html lang="ko">
    <head>
        <meta charset="utf-8" />
        <title>multi for Test</title>
        <script type="text/javascript">
```

```
            document.write("<table width='80%' border='1' style='margin: auto; text-align: center'>");

                    for (var i=0; i<6; i++) {
                            document.write("<tr>");

                            for (var j=0; j<7; j++) {
                                    var txt = "(" + i + "," + j + ")";
                                    document.write("<td>" + txt + "</td>");
                            }
                            document.write("</tr>");
                    }
                    document.write("</table>");
            </script>
    </head>
    <body>
    </body>
</html>
```