

## 2장 자바스크립트 기초 문법

### 2.1 자바스크립트 기초 문법

#### 2.1.1 자바스크립트 선언문

- 선언문은 스크립트 영역을 나타낸다.

```
// 기본형
<script type="text/javascript">
    자바스크립트 실행문;
</script>
```

[실습파일: statement\_test.html, 완성파일: statement\_c.html]

```
01  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
02  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
03  <html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
04  <head>
05  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
06  <title>선언문</title>
07  <script type="text/javascript">
08      /*
09          선언문 은 내부 스크립트(HTML 내부에 작성)로
10          작성할 때 사용합니다.
11      */
12      document.write("환영합니다"); //출력 기능(메서드) 실행
13  </script>
14
15  </head>
16  <body>
17
18  </body>
19  </html>
```

#### 2.1.2 자바스크립트 주석 처리

- 자바스크립트 선언문 안에서 실행문이 아닌 설명글을 적고 싶을 때는 주석 처리를 하면 된다. 이때 한 줄 주석인 경우는 // 한 줄 설명글로 작성하면 되고, 여러 줄 주석일 경우에는 /\* 여러 줄 설명글 \*/로 작성하면 된다.

#### 2.1.3 내부 스크립트 외부로 분리하기

- (X)HTML 내부에 작성된 자바스크립트는 외부로 분리하는 것이 좋다. 그러면 자바스크립트 소스 찾기도 쉬울 뿐만 아니라 누군가 소스를 손상시킬 일도 안 생긴다. 즉, 외부로 분리하는 주된 목적은 관리를 위해서이다.

- 기본형: <script type="text/javascript" src="소스경로"></script>

[js/example.js]

```
01  document.write("환영합니다");
```

[실습파일: import\_test.html, 완성파일: import\_c.html]

```
01 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
02 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
03 <html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
04 <head>
05 <meta http-equiv="content-type" content="text/html; charset=utf-8" />
06 <title>외부 자바스크립트 연동</title>
07 <script type="text/javascript" src="js/example.js"></script>
08 </head>
09 <body>
10
11 </body>
12 </html>
```

## 2.1.4 자바스크립트 코드 입력 시 주의할 점

- 자바스크립트는 대·소문자를 가려서 써야 한다.  
예) New data(); //(X)  
new data(); //(0), 낱자 객체 생성
- 실행문을 마치고 나서는 세미콜론(;)을 작성하는 것이 좋다. 세미콜론을 쓰지 않으면, 다음 예제처럼 한 줄에 두 개의 실행문을 작성할 때 문제가 생긴다.  
예) document.write("hi") document.write("bye") (X)  
document.write("hi"); document.write("bye"); (0)
- 실행문을 작성할 때는, 한 줄에 한 문장만 작성하는 것이 가독성을 위해 좋다.
- 문자형 데이터를 작성할 때는 큰따옴표(" ")와 작은따옴표(' ') 겹침 오류를 주의해야 한다.
- 실행문을 작성할 때 중괄호{ } 또는 소괄호( )의 짝이 맞아야 한다.
- 방문자가 마우스가 없을 때도 키보드만으로도 접근할 수 있도록 작성하는 것이 좋다.

## 2.2 변수

### 2.2.1 변수란?

- 변수(Variables)는 변하는 데이터(값)를 저장할 수 있는 메모리 공간이다.

```
// 기본형
var 변수명, 또는 var 변수명=값;
```

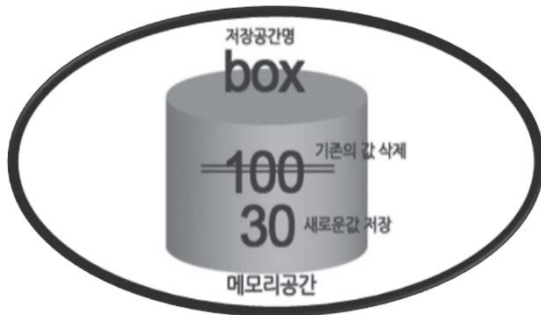
[실습파일: var\_ex1\_test.html, 완성파일: var\_ex1\_c.html]

```
01 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
02 transitional.dtd">
03 <html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
04 <head>
05 <meta http-equiv="content-type" content="text/html; charset=utf-8" />
06 <title>변수 </title>
07 <script type="text/javascript">
08     var box; // 변수를 선언합니다.
09     box=100; // 변수 box에 100을 저장했습니다.
10     box=30; // 변수 box에 저장된 데이터를 지우고, 30을 저장합니다.
```

```

11     document.write(box);
12         // 출력문을 사용하여 변수 box에 저장된 데이터를 화면에 출력합니다.
13 </script>
14 </head>
15 <body>
16
17 </body>
18 </html>

```



### 2.2.2 변수에 저장할 수 있는 데이터형

- 변수에 저장할 수 있는 데이터 형(Data Type)으로는 문자 형(String), 숫자 형(Number), 논리 형(Boolean), Null & Undefined가 있다.

#### (1) 문자형 데이터

- 문자형(String) 데이터는 문자나 숫자를 큰따옴표("") 또는 작은 따옴표('')로 감싸고 있다.

```

// 기본형
var 변수명="사용할 문자나 숫자";

var s="javascript";
var num="100";
var tag="<h1> String </h1>";

```

#### (2) 숫자형 데이터

- 숫자형(Number) 데이터는 숫자만 들어가야 한다. 만일 "100"과 같이 큰따옴표가 감싸고 있다면 숫자가 아닌 문자형 데이터라 볼 수 있다. 이 경우에는 Number("100")을 이용하면 문자형 데이터를 숫자형 데이터로 바꿀 수 있다.

```

// 기본형
var 변수명=숫자, 또는 Number("숫자");

var s=100;
var t=Number("500");

```

#### (3) 논리형 데이터

- 논리형(Boolean) 데이터는 true(참) 또는 false(거짓)가 있다.

```
// 기본형
var 변수명 = true or false 또는 Boolean(데이터);

var s = true;
var t = 10>=100; // 변수 t에 10>=100의 결과 데이터 false를 저장한다.
var k = Boolean("hello"); // 변수 k에 반환된 데이터 true를 저장한다.
var m = Boolean(0); // false를 반환한다.
```

[실습파일: boolean\_test.html, 완성파일: boolean\_c.html]

```
01 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
02 transitional.dtd">
03 <html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
04 <head>
05 <meta http-equiv="content-type" content="text/html; charset=utf-8" />
06 <title>논리형 데이터</title>
07 <script type="text/javascript">
08     var a = true; //변수 a에 데이터 true가 저장됩니다.
09     var b = false; //변수 b에 데이터 false가 저장됩니다.
10     var c = 10 > 5; //변수 c에 10>5의 결과 데이터 true가 저장됩니다.
11     var d = Boolean(null); //변수 d에 반환된 데이터 false가 저장됩니다.
12
13     //각 변수에 저장된 데이터를 출력합니다.
14     document.write(a, "<br />");
15     document.write(b, "<br />");
16     document.write(c, "<br />");
17     document.write(d, "<br />");
18 </script>
19 </head>
20 <body>
21
22 </body>
23 </html>
```

#### (4) Null & Undefined 데이터

```
// 기본형
var s; // 값이 등록되어 있지 않은 경우 undefined
var t=null;
```

#### (5) typeof

typeof는 지정한 데이터 또는 변수에 저장된 데이터형을 알고 싶을 때 사용한다.

```
// 기본형
typeof 변수 또는 데이터;
```

[실습파일: typeof\_test.html, 완성파일: typeof\_c.html]

```
01 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
02 transitional.dtd">
03 <html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
04 <head>
05 <meta http-equiv="content-type" content="text/html; charset=utf-8" />
06 <title>데이터 타입</title>
07 <script type="text/javascript">
08     //<![CDATA[
09     var num = 100; //변수 num에 숫자 형 데이터 100이 저장됩니다.
10
11     //변수 str에 문자 형 데이터 "자바스크립트"가 저장됩니다.
```

```

12         var str = "자바스크립트";
13
14         //변수에 저장된 데이터의 형을 출력합니다.
15         document.write(typeof num, "<br />");
16         document.write(typeof str, "<br />");
17     //]]>
18 </script>
19 </head>
20 <body>
21
22 </body>
23 </html>

```

### [꿀팁] CDATA 선언문은 왜 하는가?

앞의 예제에서 스크립트 구문에 `<![CDATA[ //]]>`를 사용하는 것을 보았다. 이것을 CDATA 선언문이라고 한다. 대부분 자바스크립트는 `<head></head>` 영역에 작성하는데, 자바스크립트에 태그가 포함되어 있으면 유효성 검사기는 `<head>` 영역에 문단 태그를 사용한 것으로 인식한다. 하지만 CDATA 선언문을 선언하면 유효성 검사 시 태그들을 단순히 문자형 데이터로 간주하여 문법 오류가 나는 것을 막을 수 있다.

### (6) 변수를 선언할 때 주의할 사항

- 변수명 첫 글자로는 \$, \_(언더바), 영문자만 올 수 있다.
- 변수명 첫 글자 다음은 영문자, 숫자, \$, \_를 포함할 수 있다.
- 변수명으로는 예약어(document, location, window 등)를 사용할 수 없다.
- 변수명을 지을 때는 되도록 의미를 부여해 작성하는 것이 좋다.
- 변수명을 사용할 때는 대·소문자를 구분해야 한다.

```

var 1num=10; (X)
var $num=10; (0)

var 100num=10; (X)
var num100=10; (0)

var document=10; (X)
var num=10; (0)

var num="hello"; (X)
var num=10; (0)

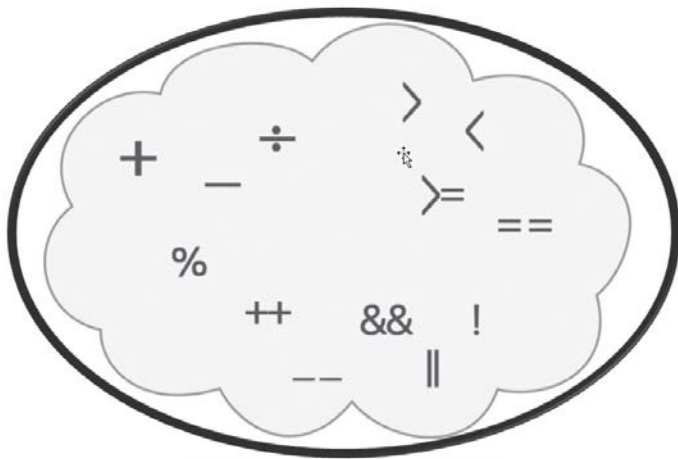
var num=10; document.write(Num) (X)
var num=10; document.write(num) (0)

```

## 2.3 연산자

### 2.3.1 연산자란?

- 연산자는 프로그래밍 계산 작업이라고 보면 된다. 연산자의 종류에는 산술 연산자, 문자 결합 연산자, 대입 연산자, 증감 연산자, 비교 연산자, 논리 연산자, 삼항 조건 연산자가 있다.



### 2.3.2 산술 연산자

- 산술 연산자는 산수 시간에 배운 더하기(+), 빼기(-), 곱하기(\*), 나누기(/), 나머지(%)가 있다.

[실습파일: part2-5-1-test.html, 완성파일: part2-5-1.html]

```
01 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
02 transitional.dtd">
03 <html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
04 <head>
05 <meta http-equiv="content-type" content="text/html; charset=utf-8" />
06 <title>산술 연산자</title>
07 <script type="text/javascript">
08     <![CDATA[
09         var num1 = 15; //연산자를 사용하기 전에 변수 선언을 합니다.
10         var num2 = 2;
11         var result;
12         result = num1 + num2; //더하기
13         document.write(result, "<br />");
14         result = num1 - num2; //빼기
15         document.write(result, "<br />");
16         result = num1 * num2; //곱하기
17         document.write(result, "<br />");
18         result = num1 / num2; //나누기
19         document.write(result, "<br />");
20         result = num1 % num2; //나머지
21         document.write(result, "<br />");
22     </![CDATA[
23 </script>
24 </head>
25 <body>
26
27 </body>
28 </html>
```

### 2.3.3 문자 결합 연산자

- 문자 결합 연산자는 피연산자(연산 대상 데이터) 중 문자 형 데이터를 포함한다. 여러 개 문자를 하나의 문자형 데이터로 결합할 때 주로 사용한다.

[실습파일: string\_plus\_test.html, 완성파일: string\_plus\_c.html]

```

01 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
02 transitional.dtd">
03 <html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
04 <head>
05 <meta http-equiv="content-type" content="text/html; charset=utf-8" />
06 <title> 문자결합 연산자 </title>
07 <script type="text/javascript">
08     var t1="학교중이"; //변수 t1에 문자 형 데이터가 저장됩니다.
09     var t2="땡땡땡"; //변수 t2에 문자 형 데이터가 저장됩니다.
10     var t3=8282; //변수 t3에 숫자 형 데이터가 저장됩니다.
11     var t4="어서 모이자"; //변수 t4에 문자 형 데이터가 저장됩니다.
12     var result;
13
14     /*변수 result에 「 문자+문자+숫자+문자 」이므로
15     하나의 문자로 결합된 데이터가 저장됩니다.*/
16     result=t1+t2+t3+t4; // 학교중이땡땡땡8282어서 모이자
17     document.write(result);
18 </script>
19 </head>
20 <body>
21
22 </body>
23 </html>

```

### 2.3.4 대입 연산자

- 대입 연산자(=)는 연산된 데이터를 최종적으로 변수에 저장할 때 사용한다. 복합 대입 연산자(+=, -=, \*=, /=, %=)란 산술 연산자와 대입 연산자가 복합적으로 적용된 것을 말한다.

[실습파일: dboperator\_test.html, 완성파일: dboperator\_c.html]

```

01 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
02 transitional.dtd">
03 <html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
04 <head>
05 <meta http-equiv="content-type" content="text/html; charset=utf-8" />
06 <title> 복합 대입 연산자 </title>
07 <script type="text/javascript">
08     var num1=10; //변수 num1에 10을 저장합니다.
09     var num2=3; //변수 num2에 3을 저장합니다.
10
11     num1+=num2; //num1=num1+num2; num1의 저장된 값 13;
12     document.write(num1,"<br/>"); // num1에 저장된 값을 출력합니다.
13
14     num1-=num2; //num1=num1-num2; num1의 저장값 10;
15     document.write(num1,"<br/>");
16
17     num1*=num2; //num1=num1*num2; num1의 저장값 30;
18     document.write(num1,"<br/>");
19
20     num1%=num2; //num1=num1*num2; num1의 저장값 0;
21     document.write(num1,"<br/>");
22 </script>
23 </head>
24 <body>
25
26 </body>
27 </html>

```

[실습파일: texttable\_test.html, 완성파일: texttable\_c.html]

```

01 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
02 transitional.dtd">

```

```

03 <html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
04 <head>
05 <meta http-equiv="content-type" content="text/html; charset=utf-8" />
06 <title>복합대입 연산자-테이블 만들기</title>
07 <script type="text/javascript">
08     var t = "<table border='1'>";
09     t += "<tr>";
10     t += "<td>1</td><td>2</td><td>3</td>";
11     t += "</tr>";
12     t += "</table>";
13     document.write(t);
14 </script>
15 </head>
16
17 <body>
18
19 </body>
20 </html>

```

### 2.3.5 증감 연산자

- 증감 연산자는 숫자형 데이터를 1씩 증가시키는 증가 연산자(++)가 있고, 반대로 1씩 감소시키는 감소 연산자(--)가 있다.

[실습파일: growth\_test.html, 완성파일: growth\_c.html]

```

01 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
02 transitional.dtd">
03 <html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
04 <head>
05 <meta http-equiv="content-type" content="text/html; charset=utf-8" />
06 <title>증감 연산자</title>
07 <script type="text/javascript">
08 //
09     var num1=10;           //변수 num1에 10을 저장합니다.
10     var num2=20;           //변수 num2에 20을 저장합니다.
11     var result;            //변수 result를 선언하였습니다.
12
13     num1--;                 //변수 num1에 데이터가 1 감소됩니다.
14     //변수 num1에 변화된 데이터를 출력합니다.
15     document.write(num1,"&lt;br /&gt;");
16
17     num1++;                 //변수 num1에 데이터가 1 증가됩니다.
18     //변수 num1에 변화된 데이터를 출력합니다.
19     document.write(num1,"&lt;br /&gt;");
20
21     //대입 연산자가 먼저 실행되고, 증가 연산자가 실행됩니다.
22     result=num2++;
23     //변수 result에 저장된 값 20이 출력됩니다.
24     document.write(result,"&lt;br /&gt;");
25
26     //증가 연산자가 먼저 발생되고, 대입 연산자가 실행됩니다.
27     result=++num2;
28     //변수 result에 저장된 값 22가 출력됩니다.
29     document.write(result,"&lt;br /&gt;");
30 //]]&gt;
31 &lt;/script&gt;
32 &lt;/head&gt;
33 &lt;body&gt;
34
35 &lt;/body&gt;
36 &lt;/html&gt;
</pre>
</div>
```



### 2.3.6 비교 연산자

- 두 데이터를 크다, 작다, 같다와 같이 비교할 때 사용하는 연산자이다. 연산된 결과의 값은 true(참) 또는 false(거짓)으로 논리 형 데이터를 반환한다.

종류	설명	비고
A > B	A가 B보다 크다.	
A < B	A가 B보다 작다.	
A >= B	A가 B보다 크거나 같다.	
A <= B	A가 B보다 작거나 같다.	
A == B	A와 B는 같다.	숫자를 비교할 경우 데이터형은 숫자형이든 문자형이든 상관하지 않고 표기된 숫자만 일치하면 true를 반환한다.
A != B	A와 B는 다르다.	
A === B	A와 B는 같다.	숫자를 비교할 경우 반드시 표기된 숫자와 데이터형도 일치해야만 true를 반환한다. 10과 "10"을 비교했을 경우 표기된 숫자는 같지만 하나는 숫자형 10이고 다른 하나는 문자형 "10"이므로 false를 반환한다.
A !== B	A와 B는 다르다.	

[실습파일: part2-5-5-test.html, 완성파일: part2-5-5.html]

```

01 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
02 transitional.dtd">
03 <html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
04 <head>
05 <meta http-equiv="content-type" content="text/html; charset=utf-8" />
06 <title> 비교 연산자 </title>
07 <script type="text/javascript">
08     var a=10; //변수 a의 값 10
09     var b=20; //변수 b의 값 20
10     var c=10; //변수 c의 값 10
11     var f="20"; //변수 f의 값 문자 형 "20"
12     var result;
13
14     result=a>b; //거짓
15     document.write(result,"<br/>");
16     result=a<b; //참
17     document.write(result,"<br/>");
18     result=a<=b; //참
19     document.write(result,"<br/>");
20     result=b==f; //데이터 형 상관없이 데이터가 같으므로 참
21     document.write(result,"<br/>");
22     result=a!=b; //데이터가 다르므로 참
23     document.write(result,"<br/>");
24     result=b===f; //데이터 형이 일치하지 않으므로 거짓
25     document.write(result,"<br/>");
26 </script>
27 </head>
28 <body>
29
30 </body>
31 </html>

```

### 2.3.7 논리 연산자

- 논리 연산자는 ||(or), &&(and), !(not)이 있다. 논리 연산자는 피연산자가 논리 형 데이터인 true 또는 false로 결과를 반환한다.

종류	설명
	or 연산자라 부르며, 피연산자 중 값이 하나라도 true가 존재하면 true 결과값을 반환한다.

&&	and 연산자라 부르며, 피연산자 중 값이 하나라도 false가 존재하면 false 결과값을 반환한다.
!	not 연산자라 부르며, 단항 연산자이다. 피연산자의 값이 true이면 반대로 false 결과값을 반환한다.

### 2.3.8 연산자 우선순위

■ 연산자들의 우선순위는 다음과 같다.

1. ()
2. 단항 연산자(--, ++, !)
3. 산술 연산자(+, /, %, +, -)
4. 비교 연산자(>, >=, <, <=, ==, ===, !=)
5. 논리 연산자(&&, ||)
6. 대입(복합 대입) 연산자(=, +=, -=, \*=, /=, %=)

[실습파일: prior\_test.html, 완성파일: prior\_c.html]

```

01 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
02 transitional.dtd">
03 <html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
04 <head>
05 <meta http-equiv="content-type" content="text/html; charset=utf-8" />
06 <title> 논리연산자 </title>
07 <script type="text/javascript">
08     var a=10;
09     var b=20;
10     var m=30;
11     var n=40;
12
13     var result;
14     result= a>b || b>m || m>n; // false || false || false
15     document.write(result,"<br />"); // false
16
17     result= a>b || b>m || m<=n; // false || false || true
18     document.write(result,"<br />"); // true
19
20     result= a<=b && b>=m && m<=n; // true && false && true
21     document.write(result,"<br />"); // false
22
23     result= a<=b && b<=m && m<=n; // true && true && true
24     document.write(result,"<br />"); // true
25
26     result= !(a>b); // !false
27     document.write(result,"<br />"); // true
28 </script>
29 </head>
30 <body>
31
32 </body>
33 </html>

```

### 2.3.9 삼항 조건 연산자

■ 삼항 조건 연산자는 조건식(true 또는 false의 결과 값을 반환)의 데이터 결과에 따라 실행 결과가 달라진다. 삼항 조건 연산자는 연산하기 위해서 피연산자가 세 개 필요한 삼항 연산자이다.

```

// 기본형
조건식 ? 실행문1 : 실행문2;

```

[실습파일: three\_condi\_test.html, 완성파일: three\_condi\_c.html]

```
01 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
02 transitional.dtd">
03 <html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
04 <head>
05 <meta http-equiv="content-type" content="text/html; charset=utf-8" />
06 <title>삼항 조건 연산자</title>
07 <script type="text/javascript">
08     var a = 10;
09     var b = 3;
10
11     // 변수 result에 삼항 조건 연산의 결과 값이 저장됩니다.
12     var result = a > b ? "javascript" : "hello";
13     document.write(result); // "javascript"를 출력합니다.
14 </script>
15 </head>
16 <body>
17
18 </body>
19 </html>
```

## [실습] 적정 체중을 구하는 테스트기 만들기

### (1) 적정 체중 구하기

```
// 적정 체중 계산법
적중 체중 = (본인 신장 - 100) * 0.9

// 예제
신장: 180 (cm)
몸무게: 74 (kg)
적중 체중 = (180 - 100) * 0.9 = 72 (kg)
결과: 적중 체중은 72kg이며, 현재 체중이 2kg 초과한 것을 알 수 있다.
```

- 실습 파일 weight\_step1\_test.html을 열어 문서를 작성한다. 키와 몸무게는 앞으로 바뀔 수 있으므로 다음과 같이 변수로 저장한다.
- 철수의 키와 몸무게를 계산하여 적정 체중인 72가 결과 화면에 나타난 것을 확인한다.

[실습파일: weight\_step1\_test.html, 완성파일: weight\_step1\_c.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title> 과체중 테스트 </title>
<script type="text/javascript">
//
    var userHeight=180; //변수 userHeight에 철수의 신장을 저장함
    var userWeight=74; //변수 userWeight에 철수의 몸무게를 저장함
    //철수의 정상(평균) 체중=(키-100)*0.9
    var normal_w= (userHeight-100)*0.9;
    document.write(normal_w);
//]]&gt;
&lt;/script&gt;
&lt;/head&gt;</pre></div>
```

```
</body>

</body>
</html>
```

## (2) 적중 체중을 구하는 실행문 만들기

```
//prompt() 메서드는 사용자로부터 입력된 값을 문자형 데이터로 반환한다.
prompt("질문", "기본 응답");
```

[실습파일: weight\_input\_test.html, 완성파일: weight\_input\_c.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>적정 몸무게 구하기</title>
<script type="text/javascript">
    //방문자에게 질의응답 창을 띄워 신장을 입력받습니다.
    var height = prompt("당신의 신장은?", "0");

    // 산술 결과 정상체중이 변수 result에 저장 됩니다.
    var result = (height - 100) * 0.9;

    document.write(result); //결과 값을 출력합니다.
</script>
</head>

<body>

</body>
</html>
```

## [과제] 도전 Mission!

(1) 진영이의 하루 지출 내역이 다음과 같다고 할 때, 하루 지출 비용의 합계를 구한 후 적정 지출 비용의 초과 여부를 출력하세요.

- 진영이의 하루 지출 내역은 교통비 3,000원, 식비 6,000원, 음료비 3,000원입니다. 삼항 조건 연산자를 사용하여 하루 적정 지출 비용인 1만원을 초과했을 경우 "000원 초과"라고 출력되고, 아니라면 "돈 관리 잘 했어요!"라고 출력되도록 해 보세요.

[실습파일: 2장\_미션/test\_1\_1\_test.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>종합실습1-1</title>
<script type="text/javascript">
    var price1=3000;
    var price2=6000;
    var price3=3000;
```

```
</script>
</head>

<body>

</body>
</html>
```

(2) 다음 연산자를 이용하여 적정 몸무게 이상/미달 값이 출력되도록 해 보세요.

- 방문자에게 질의응답 창을 이용하여 신장과 몸무게를 입력받아 옵니다. 방문자의 몸무게가 적정 몸무게보다 이상일 경우에는 "적정 몸무게 이상", 미만일 경우에는 "적정 몸무게 미달"로 출력되도록 하세요. 적정 몸무게 공식은 '적정 몸무게 = (신장-100)\*0.9'입니다.

[실습파일: 2장\_미션/test\_1\_3\_test.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title> 종합실습1-3 </title>
<script type="text/javascript">
    var user_height=prompt("당신의 신장은?", "0");
    var user_weight=prompt("당신의 체중은?", "0");
    var normal_weight=

    document.write(result);
</script>
</head>
<body>

</body>
</html>
```