

## 03-03 선택문

### switch문

선택문인 switch문은 변수에 저장된 값과 switch문에 있는 경우(case)의 값을 검사하여 변수와 경우의 값에서 일치하는 값이 있을 때 그에 해당하는 코드를 실행합니다.

if문과 용도는 비슷하나 if문은 만족하는 데이터가 여러 개일 경우에 주로 사용하고, switch문은 여러 경우의 값 중 일치하는 데이터를 찾아 그에 해당하는 코드를 실행시킬 때 사용합니다.



기본형 | switch(변수) { case 값1 코드; break; default: 코드5; }

변수에 저장된 값은 switch문을 만나면 case의 값을 하나씩 검사합니다.

일치하는 데이터가 있으면 그에 해당하는 코드를 실행하고 break문을 만나 switch문을 종료합니다.

만일 경우(case)의 값 중에 일치하는 데이터가 없으면 마지막 default에 있는 코드를 실행하고 switch문을 종료합니다.

다음은 질의응답 창을 통해 포털 사이트 이름을 입력받고, switch문을 이용하여 입력한 이름에 해당하는 사이트로 이동하도록 작성한 예제입니다.

```
<script>
// 문자열 데이터는 줄을 바꾸면 오류가 발생합니다.
// 하지만 \와 함께 코드를 작성하면 문자열의 줄을 바꿔도 오류가 발생하지 않습니다.

var site=prompt("네이버, 다음, 네이트, 구글 중\ 즐겨 사용하는 포털 검색 사이트는?");
var url;

switch(site){
    case "구글" : url="www.google.com";
                break;
    case "다음" : url="www.daum.net";
                break;
```

```

        case "네이버" : url="www.naver.com"; // site의 값이 네이버일 경우에 실행
        break;
        case "네이트" : url="www.nate.com";
        break;
        default : alert("보기 중에 없는 사이트입니다.");
    }

    if(url) location.href="http://"+url; // url에 저장된 주소로 이동합니다.

</script>

```

## 산술 연산자

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

산술 연산자로 연산을 하기 위해서는 연산 대상 데이터가 반드시 2개 있어야 합니다.

다음은 산술 연산자의 종류와 기본형을 정리한 표입니다.

| 연산자 | 설명     | 사용 예      | 사용 예에 대한 설명                    |
|-----|--------|-----------|--------------------------------|
| =   | 대입 연산자 | a = 3     | 정수 3을 a에 대입한다.                 |
| +   | 더하기    | a = 5 + 3 | 정수 5와 3을 더한 값을 a에 대입한다.        |
| -   | 빼기     | a = 5 - 3 | 정수 5와 3을 뺀 값을 a에 대입한다.         |
| *   | 곱하기    | a = 5 * 3 | 정수 5와 3을 곱한 값을 a에 대입한다.        |
| /   | 나누기    | a = 5 / 3 | 정수 5를 3으로 나눈 값을 a에 대입한다.       |
| %   | 나머지 값  | a = 5 % 3 | 정수 5를 3으로 나눈 뒤 나머지 값을 a에 대입한다. |

산술 연산자의 종류와 기본형

다음 예제를 통해 더하기, 빼기, 곱하기, 나누기, 나머지 연산자를 어떻게 사용하는지 확인해 보세요.

```

<script>
    var num1=15;
    var num2=2;
    var result;
    result=num1+num2;
    document.write(result,"<br>");
    result=num1-num2;

```

```

document.write(result, "<br>");
result=num1*num2;
document.write(result, "<br>");
result=num1/num2;
document.write(result, "<br>");
result=num1%num2;
document.write(result, "<br>");
</script>

```

## 문자 결합 연산자

문자 결합 연산자는 피연산자(연산 대상 데이터)가 문자형 데이터입니다.

여러 개의 문자를 하나의 문자형 데이터로 결합할 때 사용합니다.

다음과 같이 더하기에 피연산자로 문자형 데이터가 한 개라도 포함되어 있으면 다른 피연산자의 데이터는 자동으로 문자형 데이터로 형 변환되고 문자 결합이 이루어져 하나의 문자형 데이터를 반환합니다.



기본형 |

문자형 데이터 + 문자형 데이터 = 하나의 문자형 데이터

ex) "do it" + "javascript" = "do it javascript"

문자형 데이터 + 숫자형 데이터 = 하나의 문자형 데이터

ex) "100"+200="100200";

다음은 t1, t2, t3, t4라는 변수에 문자와 숫자 데이터를 각각 넣고 하나의 문자 데이터로 결합하는 예제입니다.

```

<script>
var t1="학교종이";
var t2=" 땡땡땡 ";
var t3=8282;
var t4=" 어서 모이자";
var result;

```

```
result=t1+t2+t3+t4;
document.write(result); //"학교종이 땡땡땡 8282 어서 모이자"
</script>
```

## 대입연산자

대입 연산자(=)는 연산된 데이터를 변수에 저장할 때 사용합니다.

복합 대입 연산자(+=, -=, \*=, /=, %=)는 산술 연산자와 대입 연산자가 복합적으로 적용된 것을 말합니다.

다음 예제를 보며 대입 연산자에 대해 알아보겠습니다.

```
<script>
    var num1=10;
    var num2=3;

    num1+=num2; //10+3
    document.write(num1, "<br>");

    num1-=num2; //13-3
    document.write(num1, "<br>");

    num1*=num2; //10*3
    document.write(num1, "<br>");

    num1%=num2;
    document.write(num1, "<br>"); // 30%3
</script>
```

다음은 여러 개의 문자형 데이터로 저장된 HTML 태그를 복합 대입 연산자를 이용하여 하나의 문자로 결합한 다음 document.write(str); 을 이용하여 화면에 출력하는 예제입니다.

```
<script>
    var str="<table border='1'>";
    str="<tr>";
    str="<td>1</td><td>2</td><td>3</td>";
    str+="</tr>";
    str+="</table>";
```

```
document.write(str);  
</script>
```

## 증감 연산자

증감 연산자에는 숫자형 데이터를 1씩 증가시키는 증가 연산자(++)와 반대로 1씩 감소시키는 감소 연산자(--)가 있습니다.

증감 연산자는 앞에서 배운 연산자와는 달리 피연산자가 한 개만 필요한 다항 연산자입니다. 증감 연산자는 변수의 어느 위치에 오는가에 따라 결괏값이 달라집니다.



### 기본형 |

- 1) 변수의 값을 1만큼 감소시킵니다. → 변수--; 또는 --변수;
- 2) 변수의 값을 1만큼 증가시킵니다. → 변수++; 또는 ++변수;

다음은 증감 연산자를 사용해 변수에 저장된 숫자를 증가하거나 감소하여 문서에 출력하는 예제입니다.

```
<script>  
    var num1=10;  
    var num2=20;  
    var result;  
  
    num1--; //9  
    document.write(num1,"<br>");  
  
    num1++; //10  
    document.write(num1,"<br>");  
  
    result=num2++; // result: 20, num2: 21  
    document.write(result,"<br>");  
  
    result=++num2; //result:22, num2:22  
    document.write(result,"<br>");  
</script>
```

## 비교 연산자

두 데이터를 '크다, 작다, 같다'와 같이 비교할 때 사용하는 연산자입니다.

연산된 결과값은 true(참) 또는 false(거짓)로 논리형 데이터를 반환합니다. 다음은 비교 연산자의 종류입니다.

| 연산자 | 개요   | 사용 예         | 결과     |
|-----|--|--------------|--------|
| ==  | 좌변과 우변의 값이 같을 경우 true                        | 3 == 3       | true   |
| !=  | != 좌변과 우변의 값이 같지 않을 경우, true                 | 3 != 3       | false  |
| <   | 좌변이 우변보다 작은 경우, true                         | 3 < 7        | true   |
| <=  | 좌변이 우변보다 작거나 같을 경우, true                     | 3 <= 3       | true   |
| >   | 좌변이 우변보다 클 경우, true                          | 3 > 7        | false  |
| >=  | 좌변이 우변보다 크거나 같을 경우, true                     | 3 >= 3       | true   |
| === | 좌변과 우변의 값이 같고 데이터 형도 같은 경우, true             | 3 === 3      | true   |
| !== | 좌변과 우변 값이 같지 않거나 데이터 형이 다른 경우, true          | 3 !== 3      | false  |
| ?:  | 조건 식? 식1:식2, true 인 경우 식1을 false 인 경우 식2를 수행 | <x==1) ? 1:0 | 1 또는 0 |

비교 연산자 ==와 ≠는 피연산자의 자료형(type)이 비교 연산자의 실행 결과에 영향을 미치지 않습니다.

다음 예제의 경우 변수 k에는 숫자형 데이터 10이 저장되었고, 변수 m에는 문자형 데이터 "10"이 저장되었습니다.

그리고 두 변수 모두 숫자형 데이터 10과 같은지 비교하였고 결과는 둘 다 true를 반환합니다.

```
var k=10, m="10";
k===10 //ture
m===10 // true
```

하지만 비교 연산자 중 ===와 ≠는 피연산자 자료형에 따라 결과값이 달라집니다.

다음 예제의 경우 변수 k에는 숫자형 데이터 10이 저장되었고, k===10은 숫자형 데이터로 둘 다 일치하기 때문에 true를 반환합니다.

하지만 변수 m에는 문자형 데이터 "10"이 저장되었습니다.

문자형 데이터 "10"과 숫자형 데이터 10은 자료형이 다르므로 m===10은 false를 반환합니다.

```
var k=10, m="10";  
k===10 //true  
m===10 //false
```

다음 예제로 비교 연산자에 대해 더 살펴보겠습니다.

a,b,c,d라는 변수에 값을 각각 지정하고 지정하고 위에서 배운 비교 연산자를 이용해 결과값을 반환합니다.

```
<script>  
  var a=10;  
  var b=20;  
  var c=10;  
  var f="20";  
  var result;  
  
  result=a>b; //false  
  document.write(result,"<br>");  
  result=a<b; //true  
  document.write(result,"<br>");  
  result=a<=b; //true  
  document.write(result,"<br>");  
  result=b==f; //true  
  document.write(result,"<br>");  
  result=a!=b; //true  
  document.write(result,"<br>");  
  result=b===f; //false  
  document.write(result,"<br>");  
</script>
```

---

## 논리 연산자

논리 연산자에는 || (or), &&(and), !(not)이 있으며, 논리 연산자는 피연산자가 논리형 데이터인 true 또는 false로 결괏값을 반환합니다.

|| (or) 연산자는 피연산중 하나만 true이면 true라는 결괏값을 반환합니다.

하지만 &&(and) 연산자는 피연산자 중 하나만 false이면 false라는 결괏값을 반환합니다.

!(not)은 논리 부정 연산자로, 피연산자가 true이면 false라는 반대의 결괏값을 반환합니다.

논리 연산자의 종류를 표로 정리하면 다음과 같습니다.

| 연산자         | 구문                                  | 설명  |
|-------------|-------------------------------------|---|
| 논리 AND (&&) | <code>expr1 &amp;&amp; expr2</code> | <code>expr1</code> 을 true로 변환할 수 있는 경우 <code>expr2</code> 을 반환하고, 그렇지 않으면 <code>expr1</code> 을 반환합니다. |
| 논리 OR (  )  | <code>expr1    expr2</code>         | <code>expr1</code> 을 true로 변환할 수 있으면 <code>expr1</code> 을 반환하고, 그렇지 않으면 <code>expr2</code> 를 반환합니다.   |
| 논리 NOT (!)  | <code>!expr</code>                  | 단일 피연산자를 true로 변환할 수 있으면 false를 반환합니다. 그렇지 않으면 true를 반환합니다.   |

### 논리 연산자의 종류

## 연산자 우선순위

일반적인 산수를 연산할 때처럼 연산자에도 우선순위가 있습니다.

예를 들어 '2+(1\*3)=5'와 같은 식입니다. 이 식의 결과가 '9'라고 대답한다면 우선순위를 고려하지 않은 것이겠죠?

지금까지 배운 연산자들의 우선순위는 다음과 같습니다.



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

가령, 이러한 우선순위를 고려하여 ++A\*B≤C 라는 코드를 실행한다면 몇 변수 A에 데이터를 증가시키고 B를 곱한다음 마지막으로 곱한 값을 C와 비교한 후 최종 결괏값을 반환합니다.



다.

다음은 비교 연산자와 논리 연산자를 적용한 예제입니다.

코드 한 줄에 비교 연산자와 논리 연산자가 함께 포함되어 있다면 연산자 우선순위에 따라 비교 연산이 먼저 실행되고, 그 다음 논리 연산자가 실행됩니다.

```
<script>
    var a=10;
    var b=20;
    var m=30;
    var n=40;

    var result;
    result=a>b || b>m || m>n; // false || false || false
    document.write(result,"<br>"); //false

    result=a>b || b>=m || m<=n; //false || false || true
    document.write(result,"<br>"); //true

    result=a<=b && b>=m && m<=n; // true && false && true
    document.write(result,"<br>"); // false

    result=a<=b && b<=m && m<=n; //true && true && ture
    document.write(result,"<br>"); //true

    result=!(a>b); //!false를 계산
    document.write(result,"<br>"); //true
</script>
```

## 실행 조건 연산자

삼항 조건 연산자는 조건식(false 또는 true의 결괏값을 반환)의 결과에 따라 실행 결과가 달라지는 삼항 연산자로, 연산을 위해 피연산자 3개가 필요합니다.

삼항 조건 연산자는 연산한 결과 조건식의 만족 여부에 따라 실행하는 코드를 다르게 실행하고자 할 때 사용합니다.

예를 들어 방문자로부터 입력받은 나잇값이 20세 이상이면 '성인'이라 출력하고, 미만이면 '미성년자'라고 출력하도록 할 때 사용할 수 있습니다.



## 기본형 | 조건식 ? 자바스크립트 코드1 : 자바스크립트 코드2

```
<script>
  var a=10;
  var b=3;

  var result=a>b ? "javascript":"hello";
  document.write(result);
</script>
```

### | 총정리 실습! | 걱정 체중을 구하는 테스트기 만들기

다음 걱정 체중 구하기 테스트는 앞에서 배운 문법들을 사용하여 만든 예제입니다.

물론 당장 완벽하게 하나하나 만들기는 어렵겠지만 자바스크립트 전체의 기본 문법을 배우고 나면 여러분 스스로 만들 수 있을 것입니다.

우선 시작하기 전에 걱정 체중 구하기 테스트 예제의 작동 원리에 대해 간단히 살펴보겠습니다.

예를 들어 철수의 신장이 180cm이고 체중이 74kg이라고 했을 때 체중 상태를 알아보겠습니다.

다음은 걱정 체중 계산법입니다.



걱정 체중 = (본인 신장-100) x 0.9

철수의 걱정 체중을 계산하면 다음과 같은 계산 결과가 나옵니다.



신장 : 180(cm)

체중 : 74(kg)

적정 체중 :  $(180-100) \times 0.9 = 72(\text{kg})$

결과 : 적정 체중은 72kg이며, 현재 적정 체중에서 2kg 초과한 것을 알 수 있습니다.

## 적정 체중 구하기

앞에서 배운 연산자를 사용하여 철수의 적정 체중이 웹 페이지에 나타나도록 만들어 보겠습니다.

위 공식에서 입력한 신장과 체중은 사용자의 신장과 체중에 따라 값이 변하므로 변수에 저장해야 합니다.

## 적정 체중 구하는 코드 작성하기

앞에서 살펴본 적정 체중 구하기 예제에서 산술 연산자를 사용했던 것을 기억하시나요?

이제 적정 체중값을 단순히 페이지에 보여주기만 하는 것이 아닌, 페이지를 방문한 방문자가 질의응답 창에 자신의 신장값을 입력하면 적정 체중을 구해 주는 코드를 작성하겠습니다.

질의응답 창은 방문자에게 질문을 던져 응답을 받아올 수 있는 창입니다.

이 창을 사용할 때는 `prompt()` 메서드를 사용합니다.

`prompt()` 메서드의 기본 사용법은 아래와 같습니다.



기본형 | `prompt("질문", "기본 응답");`

```

<script>
    var userHeight=180; // 신장
    var userWeight=74; // 체중
    var name=prompt("당신의 이름은?", ""); // 이름 입력
    var height=prompt("당신의 신장은?", "0"); // 신장 입력
    var weight=prompt("당신의 몸무게는?", "0"); // 체중 입력

    //평균 체중 = (신장-100)*0.9
    var normal_w=(userHeight-100)*0.9; // 평균 체중

```

```
var result=weight>=normal_w-5&&weight<=normal_w+5; // 오차 범위 +-5
result=result? "적정 체중입니다." : "적정 체중이 아닙니다.";
document.write(name+"님은 "+result);
</script>
```