

switch statement

복수의 if 조건문은 switch문으로 바꿀 수 있다.

switch문을 사용한 비교법은 특정 변수를 다양한 상황에서 비교할 수 있게 해준다. 코드 자체가 비교 상황을 잘 설명한다는 장점도 있다.

❖ Syntax(문법)

switch 문은 하나 이상의 case 문으로 구성된다. 대개 default 문도 있지만, 이는 필수는 아니다.

[예시]

```
switch(x) {  
  case 'value1': // if (x == 'value1')  
    ...  
    [break]  
  
  case 'value2': // if (x == 'value2')  
    ...  
    [break]  
  
  default:  
    ...  
    [break]  
}
```

변수 x의 값과 첫 번째 case문의 값 'value1'를 일치 비교한 후, 두 번째 case문의 값 'value2'와 비교합니다. 이런 과정은 계속 이어진다.

case문에서 변수 x의 값과 일치하는 값을 찾으면 해당 case 문의 아래의 코드가 실행된다. 이때, break 문을 만나거나 switch 문이 끝나면 코드의 실행은 멈춘다.

값과 일치하는 case문이 없다면, default문 아래의 코드가 실행된다(default 문이 있는 경우).

❖ 예시

실제 실행 가능한 switch 문 예시를 살펴보자. 아래 예시에선 강조된 코드가 실행된다.

```
let a = 2 + 2;
```

```

switch (a) {
  case 3:
    alert( '비교하려는 값보다 작습니다.' );
    break;
  case 4:
    alert( '비교하려는 값과 일치합니다.' );
    break;
  case 5:
    alert( '비교하려는 값보다 큼니다.' );
    break;
  default:
    alert( "어떤 값인지 파악이 되지 않습니다." );
}

```

switch문은 a의 값인 4와 첫 번째 case문의 값인 3을 비교한다. 두 값은 같지 않기 때문에 다음 case문으로 넘어간다.

a와 그다음 case문의 값인 4는 일치한다. 따라서 break문을 만날 때까지 case 4 아래의 코드가 실행된다.

case문 안에 break문이 없으면 조건에 부합하는지 여부를 따지지 않고 이어지는 case문을 실행한다.

break문이 없는 경우 어떤 일이 일어나는지 예시를 통해 살펴보자.

```

let a = 2 + 2;

switch (a) {
  case 3:
    alert( '비교하려는 값보다 작습니다.' );
  case 4:
    alert( '비교하려는 값과 일치합니다.' );
  case 5:
    alert( '비교하려는 값보다 큼니다.' );
  default:
    alert( "어떤 값인지 파악이 되지 않습니다." );
}

```

위 예시를 실행하면 아래 3개의 alert문이 실행된다.

```

alert( '비교하려는 값과 일치합니다.' );
alert( '비교하려는 값보다 큼니다.' );
alert( "어떤 값인지 파악이 되지 않습니다." );

```

⊙ switch/case 문의 argument엔 어떤 표현식이든 올 수 있다.

switch 문과 case 문은 모든 형태의 표현식을 인수로 받는다.

[예시]

```
let a = "1";
let b = 0;

switch (+a) {
  case b + 1:
    alert("표현식 +a는 1, 표현식 b+1는 1이므로 이 코드가 실행됩니다.");
    break;

  default:
    alert("이 코드는 실행되지 않습니다.");
}
```

표현식 +a를 평가하면 1이 된다. 이 값은 첫 번째 case문의 표현식 b + 1을 평가한 값(1)과 일치한다. 따라서 첫 번째 case문 아래의 코드가 실행된다.

❖ 여러 개의 "case"문 묶기

코드가 같은 case문은 한데 묶을 수 있다.

case 3과 case 5에서 실행하려는 코드가 같은 경우에 대한 예시를 살펴보자.

```
let a = 3;

switch (a) {
  case 4:
    alert('계산이 맞습니다!');
    break;

  case 3: // (1) 두 case문을 묶음
  case 5:
    alert('계산이 틀립니다!');
    alert("수학 수업을 다시 들어보는걸 권유 드립니다.");
    break;

  default:
    alert('계산 결과가 이상하네요.');
```

case 3과 case 5는 동일한 메시지를 보여준다.

switch/case문에서 break문이 없는 경우엔 조건에 상관없이 다음 case문이 실행되는 부작용이 발생한다. 위 예시에서 case 3이 참인 경우엔 (1)로 표시한 줄 아래의 코드가 실행되는데, 그 아래 줄엔 case 5가 있고 break문도 없기 때문에 12번째 줄의 break문을 만날 때까지 코드는 계속 실행된다.

❖ Type(자료형)의 중요성

switch 문은 일치 비교로 조건을 확인한다. 비교하려는 값과 case 문의 값의 형과 값이 같아야 해당 case 문이 실행된다.

예시를 통해 switch문에서 자료형이 얼마나 중요한지 살펴보도록 한다.

```
let arg = prompt("값을 입력해주세요.");

switch (arg) {
  case '0':
  case '1':
    alert( '0이나 1을 입력하셨습니다.' );
    break;

  case '2':
    alert( '2를 입력하셨습니다.' );
    break;

  case 3:
    alert( '이 코드는 절대 실행되지 않습니다!' );
    break;
  default:
    alert( '알 수 없는 값을 입력하셨습니다.' );
}
```

- ① 0이나 1을 입력한 경우엔 첫 번째 alert문이 실행된다.
- ② 2를 입력한 경우엔 두 번째 alert문이 실행된다.
- ③ 3을 입력하였더라도 세 번째 alert문은 실행되지 않는다. 앞서 배운 바와 같이 prompt 함수는 사용자가 입력 필드에 기재한 값을 문자열로 변환해 반환하기 때문에 숫자 3을 입력하더라도 prompt 함수는 문자열 '3'을 반환한다. 그런데 세 번째 case문에선 사용자가 입력한 값과 숫자형 3을 비교하므로, 형 자체가 다르기 때문에 case 3 아래의 코드는 절대 실행되지 않는다. 대신 default문이 실행된다.

[과제]

🔑 "switch" 문을 "if" 문으로 변환하기

★ 중요도: 5

"switch" 문을 사용해 작성된 아래 코드를 if..else 문을 사용한 코드로 변환해 보자.

```
switch (browser) {  
  case 'Edge':  
    alert( "Edge를 사용하고 계시네요!" );  
    break;  
  
  case 'Chrome':  
  case 'Firefox':  
  case 'Safari':  
  case 'Opera':  
    alert( '저희 서비스가 지원하는 브라우저를 사용하고 계시네요.' );  
    break;  
  
  default:  
    alert( '현재 페이지가 괜찮아 보이길 바랍니다!' );  
}
```

[해답]

switch와 동일한 동작을 하는 코드를 작성하려면 if 문에서 일치 연산자 '==='를 써서 조건을 비교해야 한다.

그런데 문제에선 비교하려는 값(browser) 자체가 string(문자열)이기 때문에 동등 연산자 '='를 사용해도 괜찮다.

```
if(browser == 'Edge') {  
  alert("Edge를 사용하고 계시네요!");  
} else if (browser == 'Chrome'  
  || browser == 'Firefox'  
  || browser == 'Safari'  
  || browser == 'Opera') {  
  alert( '저희 서비스가 지원하는 브라우저를 사용하고 계시네요.' );  
} else {  
  alert( '현재 페이지가 괜찮아 보이길 바랍니다!' );  
}
```

해답에선 가독성을 위해 browser == 'Chrome' || browser == 'Firefox' ...을 여러 줄에 나눠서 작성하였다.

가독성을 고려해 작성하였더라도 switch 문을 이용한 코드가 더 깔끔하고 기술적(descriptive)이라는 것을 느낄 수 있다.

☞ "if"문을 "switch"문으로 변환하기

★ 중요도: 4

아래 코드를 switch 문을 사용한 코드로 바꿔보자. switch 문은 하나만 사용해야 한다.

```
let a = +prompt('a?', '');

if (a == 0) {
  alert( 0 );
}
if (a == 1) {
  alert( 1 );
}

if (a == 2 || a == 3) {
  alert( '2,3' );
}
```

[해답]

첫 번째와 두 번째 if 문은 각각 하나의 case 문으로 대체할 수 있다. 세 번째 if 문은 두 개의 case문으로 쪼개 작성할 수 있다.

```
let a = +prompt('a?', '');

switch (a) {
  case 0:
    alert( 0 );
    break;

  case 1:
    alert( 1 );
    break;

  case 2:
  case 3:
    alert( '2,3' );
    break;
}
```

맨 아래 줄에 있는 break문은 없어도 괜찮다. 해답에선 추후에 case 문이 추가될 경우를 대비하여 break문을 추가하였다.

미래에 조건을 하나 더 추가해야 하는 상황이 생길 수 있다. 이런 식으로 break 문을 넣어주는 습관을 들이면 혹시라도 발생할 수 있는 에러를 방지할 수 있다. case 4를 위 switch 문에 추가했는데 case 3에 break문이 없다면 에러가 발생하기 때문이다.