JavaScript Switch Statement

- description : JavaScript Switch Statement
- author : 오션
- email : shlim@repia.com
- lastupdate : 2021-04-26

the source of this article

JavaScript Switch Statement
`switch` 스테이트먼트는 다른 조건에 기반하여 다른 작업을 하는 데 사용됩니다.

The JavaScript Switch Statement

`switch` 스테이트먼트를 사용하여 실행할 여러 코드 블록 중 하나를 선택합니다.

Syntax

```
switch(expression) {
  case x:
    // code block
    break;
  case y:
    // code block
    break;
  default:
    // code block
}
```

작동 방식은 다음과 같습니다.

- 스위치 표현식은 한 번 평가됩니다.
- 표현식의 값은 각 케이스의 값과 비교됩니다.
- 일치하는 항목이 있으면, 연결된 코드 블록이 실행됩니다.
- 일치하는 항목이 없으면, 기본 코드 블록이 실행됩니다.

Example

`getDay()` 메서드는 0에서 6 사이의 숫자로 요일을 반환합니다.

(일요일 = 0, 월요일 = 1, 화요일 = 2 ..)

다음 예제에서는 요일 번호(weekday number)를 사용하여 요일 이름을 계산합니다.

```
let day;
switch (new Date().getDay()) {
  case 0:
    day = "Sunday";
    break;
  case 1:
    day = "Monday";
    break;
  case 2:
    day = "Tuesday";
    break;
  case 3:
    day = "Wednesday";
    break;
  case 4:
    day = "Thursday";
    break;
  case 5:
    day = "Firday";
    break;
  case 6:
    day = "Saturday";
}
document.getElementById("demo").innerHTML = "Today is " + day;
```

The break Keyword

JavaScript가 `break` 키워드에 도달하면 switch 블록을 벗어납니다.

그러면 스위치 블록 내부의 실행이 중지됩니다.

스위치 블록에서 마지막 케이스를 브레이크 할 필요는 없습니다. 어쨌든 블록이 브레이크 됩니다.(끝납니다).

Note: break 스테이트먼트를 생략하면, 평가가 case와 일치하지 않아도 다음 case가 실행됩니다.

The default Keyword

`default` 키워드는 일치하는 케이스가 없는 경우 실행할 코드를 지정합니다.

Example

getDay() 메서드는 0에서 6 사이의 숫자로 요일을 반환합니다.

오늘이 토요일 (6)도 일요일 (0)이 아닌 경우. default 메시지를 작성합니다.

```
let text;
switch (new Date().getDay()) {
  case 6:
    text = "Today is Saturday";
    break;
  case 0:
    text = "Today is Sunday";
    break;
  default:
    text = "Looking forward to the Weekend";
}
document.getElementById("demo").innerHTML = text;
```

default case의 순서가 스위치 블록의 마지막 case 일 필요는 없습니다.

```
let text;
switch (new Date().getDay()) {
  default:
    text = "Looking forward to the Weekend";
    break;
  case 6:
    text = "Today is Saturday";
    break;
  case 0:
    text = "Today is Sunday";
}
document.getElementById("demo").innerHTML = text;
```

''default''가 스위치 블록의 마지막 case가 아닌 경우, default 케이스를 break로 종료해야 합니다.

Common Code Blocks

때로는 동일한 코드를 사용하기 위해 다른 switch case를 원할 것입니다.

다음 예제에서, case 4와 case 5는 동일한 코드 블록을 공유하고, case 0과 case 6은 다른 코드 블록을

공유합니다.

```
let text;
switch (new Date().getDay()) {
  case 4:
  case 5:
    text = "Soon it is Weekend";
    break;
  case 0:
  case 6:
    text = "It is Weekend";
    break;
  default:
    text = "Looking forward to the Weekend";
}
document.getElementById("demo").innerHTML = text;
```

Switching Details

다수의 cases가 case 값과 일치하면, **첫 번째** 케이스가 선택됩니다.

일치하는 cases가 없으면, 프로그램은 **default** 레이블로 계속됩니다.

default 레이블이 없는 경우, 프로그램은 switch 이후의 스테이트먼트로 계속됩니다.

Strict Comparison

Switch cases는 **strict** 비교(===)를 사용합니다.

값이 일치하려면 동일한 유형이어야 합니다.

strict 비교는 피연산자(operands)가 동일한 유형인 경우에만 참(true)일 수 있습니다.

다음 예제에서는 x와 일치하는 항목이 없습니다.

```
let x = "0";

switch (x) {
  case 0:
    text = "Off";
  case 1:
    text = "On";
    break;
  default:
    text = "No value found";
```

```
    }
    document.getElementById("demo").innerHTML = text;    // No value found
```

오션, Javascript, Switch, Statement

From:
http://125.132.25.164/dokuwiki/ - **문제를 잘 정의하는 것은 문제를 절반 해결한 것이다. - 2023.12**

Permanent link:
http://125.132.25.164/dokuwiki/doku.php?id=wiki:javascript:javascript_note:js_switch&rev=1619430429

Last update: **2022/03/10 19:52**