

La sentencia "switch"

Una sentencia `switch` puede reemplazar múltiples condiciones `if`.

Provee una mejor manera de comparar un valor con múltiples variantes.

La sintaxis

`switch` tiene uno o mas bloques `case` y un opcional `default`.

Se ve de esta forma:

```
1 switch(x) {  
2   case 'valor1': // if (x === 'valor1')  
3     ...  
4     [break]  
5  
6 case 'valor2': // if (x === 'valor2')  
7   ...  
8   [break]  
9  
10  default:  
11    ...  
12    [break]  
13 }
```

- El valor de `x` es comparado contra el valor del primer `case` (en este caso, `valor1`), luego contra el segundo (`valor2`) y así sucesivamente, todo esto bajo una igualdad estricta.
- Si la igualdad es encontrada, `switch` empieza a ejecutar el código iniciando por el primer `case` correspondiente, hasta el `break` más cercano (o hasta el final del `switch`).
- Si no se cumple ningún caso entonces el código `default` es ejecutado (si existe).

Ejemplo

Un ejemplo de `switch` (se resalta el código ejecutado):

```
1 let a = 2 + 2;  
2  
3 switch (a) {
```

鱈 鰱

```

4   case 3:
5       alert( 'Muy pequeño' );
6       break;
7 case 4:
8       alert( '¡Exacto!' );
9       break;
10 case 5:
11 alert( 'Muy grande' );
12     break;
13     default:
14         alert( "Desconozco estos valores" );
15 }

```

Aquí el `switch` inicia comparando `a` con la primera variante `case` que es `3`. La comparación falla.

Luego `4`. La comparación es exitosa, por tanto la ejecución empieza desde `case 4` hasta el `break` más cercano.

Si no existe `break` entonces la ejecución continúa con el próximo `case` sin ninguna revisión.

Un ejemplo sin `break`:

```

1 let a = 2 + 2;
2
3 switch (a) {
4 case 3:
5     alert( 'Muy pequeño' );
6 case 4:
7     alert( '¡Exacto!' );
8 case 5:
9     alert( 'Muy grande' );
10    default:
11        alert( "Desconozco estos valores" );
12 }

```

鱈 鰾

En el ejemplo anterior veremos ejecuciones de tres `alert` secuenciales:

```

1 alert( '¡Exacto!' );
2 alert( 'Muy grande' );
3 alert( "Desconozco estos valores" );

```



Cualquier expresión puede ser un argumento switch/case

Ambos `switch` y `case` permiten expresiones arbitrarias.

Por ejemplo:

```
1 let a = "1";
2 let b = 0;
3
4 switch (+a) {
5   case b + 1:
6     alert("esto se ejecuta, porque +a es 1, exactamente igual b+1");
7     break;
8
9   default:
10    alert("esto no se ejecuta");
11 }
```

Aquí `+a` da `1`, esto es comparado con `b + 1` en `case`, y el código correspondiente es ejecutado.

Agrupamiento de "case"

Varias variantes de `case` los cuales comparten el mismo código pueden ser agrupadas.

Por ejemplo, si queremos que se ejecute el mismo código para `case 3` y `case 5`:

```
1 let a = 2 + 2;
2
3 switch (a) {
4   case 4:
5     alert('¡Correcto!');
6     break;
7
8   case 3: // (*) agrupando dos cases
9   case 5:
10    alert('¡Incorrecto!');
11    alert("¿Por qué no tomas una clase de matemáticas?");
12    break;
13
14   default:
15    alert('El resultado es extraño. Realmente.');
```

Ahora ambos `3` y `5` muestran el mismo mensaje.

La habilidad para "agrupar" cases es un efecto secundario de como trabaja `switch/case` sin `break`. Aquí la ejecución de `case 3` inicia desde la línea `(*)` y continúa a través de `case 5`, porque no existe `break`.

El tipo importa

Vamos a enfatizar que la comparación de igualdad es siempre estricta. Los valores deben ser del mismo tipo para coincidir.

Por ejemplo, consideremos el código:

```
1 let arg = prompt("Ingrese un valor");
2 switch (arg) {
3   case '0':
4   case '1':
5     alert( 'Uno o cero' );
6     break;
7
8 case '2':
9     alert( 'Dos' );
10    break;
11
12 case 3:
13     alert( '¡Nunca ejecuta!' );
14     break;
15 default:
16     alert( 'Un valor desconocido' );
17 }
```

1. Para `0`, `1`, se ejecuta el primer `alert`.
2. Para `2` se ejecuta el segundo `alert`.
3. Pero para `3`, el resultado del `prompt` es un string `"3"`, el cual no es estrictamente igual `===` al número `3`. Por tanto ¡Tenemos un código muerto en `case 3`! La variante `default` se ejecutará.

Tareas

Reescribe el "switch" en un "if" [↗](#)

importancia: 5

Escribe el código utilizando `if..else` que corresponda al siguiente `switch`:

```
1 switch (navegador) {
2 case 'Edge':
3 alert( "¡Tienes Edge!" );
4     break;
5
6 case 'Chrome':
7 case 'Firefox':
8 case 'Safari':
9 case 'Opera':
10    alert( 'Esta bien, soportamos estos navegadores también' );
```

```
11     break;
12
13     default:
14         alert( '¡Esperamos que esta página se vea bien!' );
15 }
```

solución

Reescribe "if" en "switch" [🔗](#)

importancia: 4

Reescribe el código debajo utilizando solo un argumento `switch`:

```
1 let a = +prompt('a?', '');
2
3 if (a == 0) {
4     alert( 0 );
5 }
6 if (a == 1) {
7     alert( 1 );
8 }
9
10 if (a == 2 || a == 3) {
11     alert( '2,3' );
12 }
```

鱈 𩚑

