

¡Javascript!

guayverd beta hub

Agenda del día



01

Introducción

Repaso if.

Repaso else ... if.

Repaso if ... else if ... else.

02

for

Definición.

switch

Definición.

case

break

default

03

Ejercitación

Opciones con switch



daily

¿Cómo venimos?

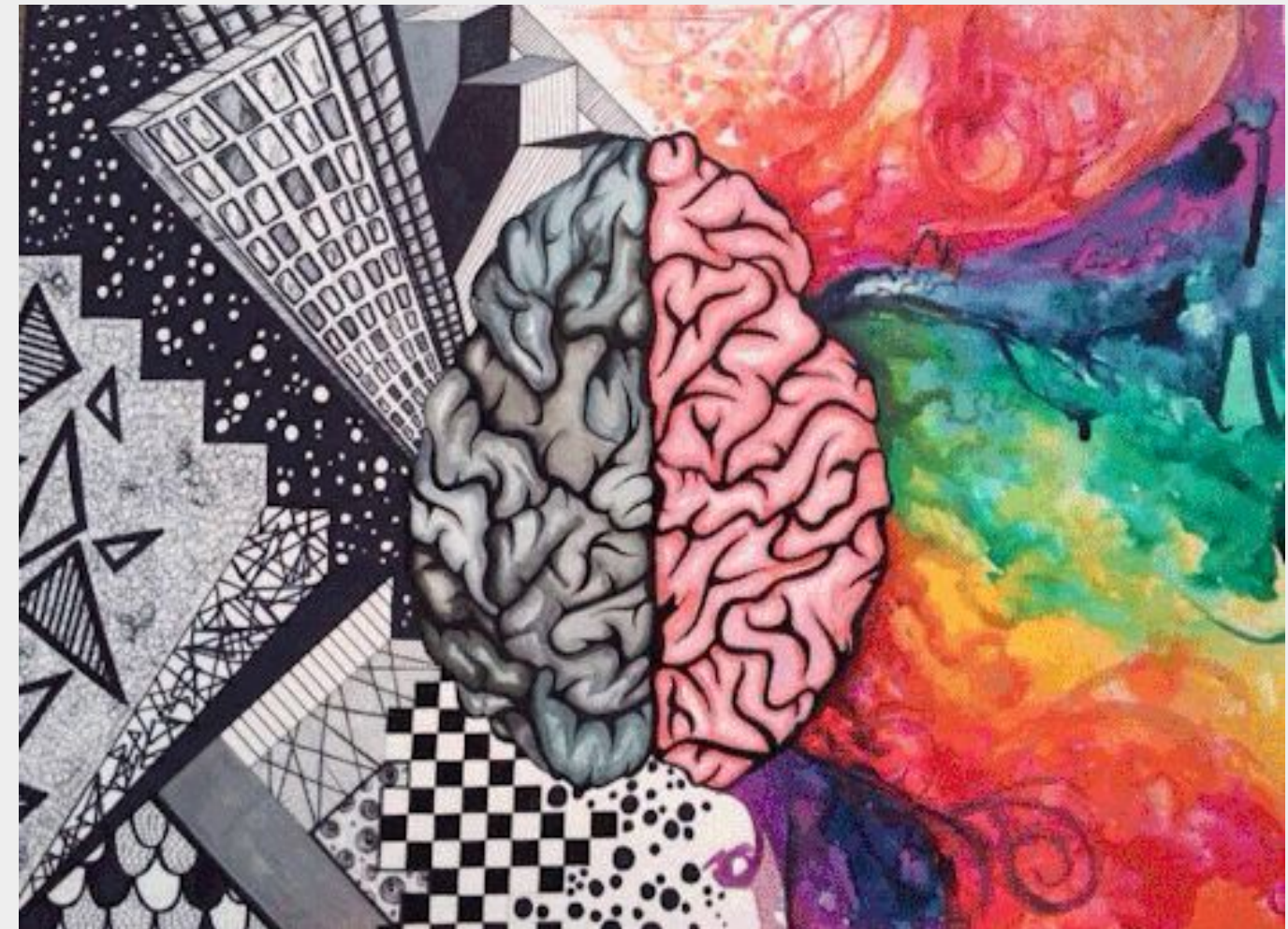
¿Algo nos bloquea?

¿Cómo seguimos?



switch

Estructuras de control



SPRINT 1

switch



switch es una estructura de control que permite recibir una variable y, dependiendo del valor obtenido ejecutar uno entre diferentes bloques de código de manera prolija.

Es similar a tener muchos else if pero más ordenado.

Comparación switch con if, else if, else

switch es más legible.

```
let calificacion = 4;

if (calificacion === 1) {
  console.log("Muy deficiente.");
} else if (calificacion === 2) {
  console.log("Insuficiente.");
} else if (calificacion === 3) {
  console.log("Suficiente.");
} else if (calificacion === 4) {
  console.log("Notable.");
} else if (calificacion === 5) {
  console.log("Sobresaliente.");
} else {
  console.log("Calificación inválida.");
}
```

```
let calificacion = 4;

switch (calificacion) {
  case 1:
    console.log("Muy deficiente.");
    break;
  case 2:
    console.log("Insuficiente.");
    break;
  case 3:
    console.log("Suficiente.");
    break;
  case 4:
    console.log("Notable.");
    break;
  case 5:
    console.log("Sobresaliente.");
    break;
  default:
    console.log("Calificación inválida.");
}
```

case en switch

Si la expresión evaluada en la condición switch(condición) es igual al case, se ejecuta el bloque del case.

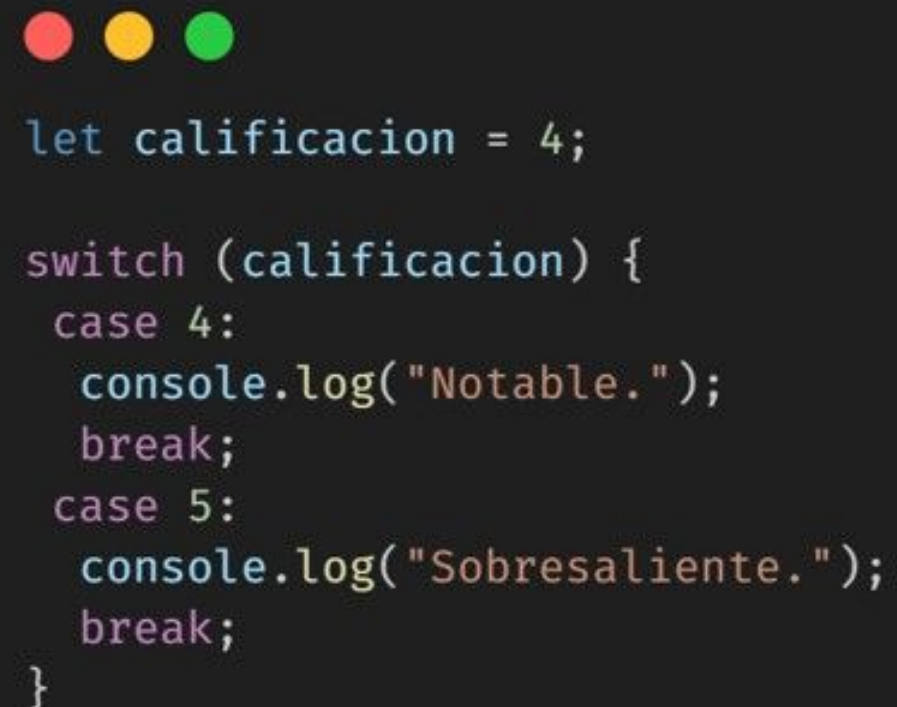
```
let calificacion = 4;

switch (calificacion) {
  case 4:
    console.log("Notable.");
    break;
}

Notable.
```

break en switch

Se utiliza break al final del bloque del case que se ejecutó para salir del switch.



```
let calificacion = 4;

switch (calificacion) {
  case 4:
    console.log("Notable.");
    break;
  case 5:
    console.log("Sobresaliente.");
    break;
}
```


default en switch

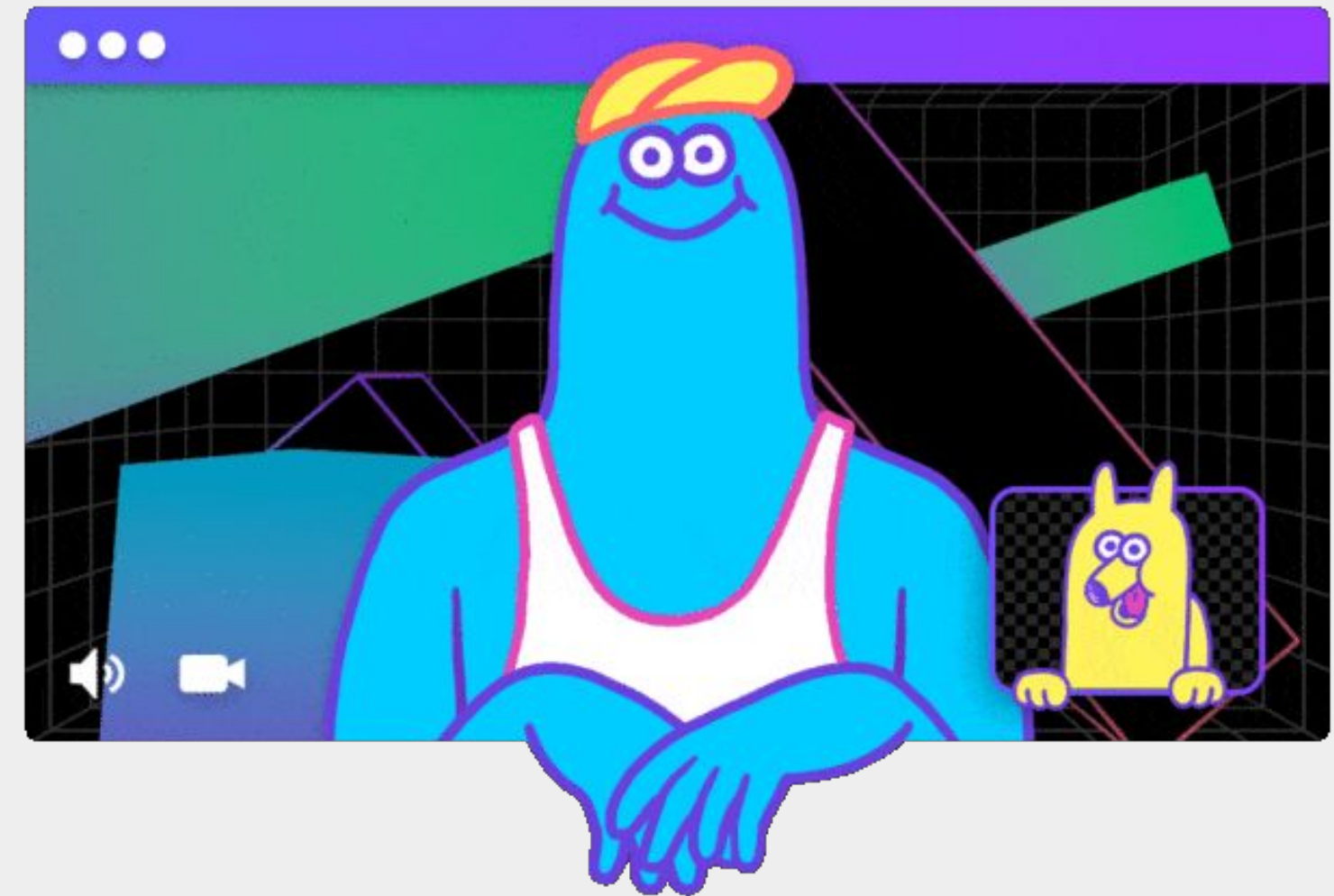
Se utiliza default luego de enumerar todas los case para poder realizar una tarea si la condición no fue capturada por ningún caso.

```
let comiste = "no se";

switch (comiste) {
  case "si":
    console.log("😊");
    break;
  case "no":
    console.log("😞.");
    break;
  default:
    console.log("El case no fue encontrado.");
}
```

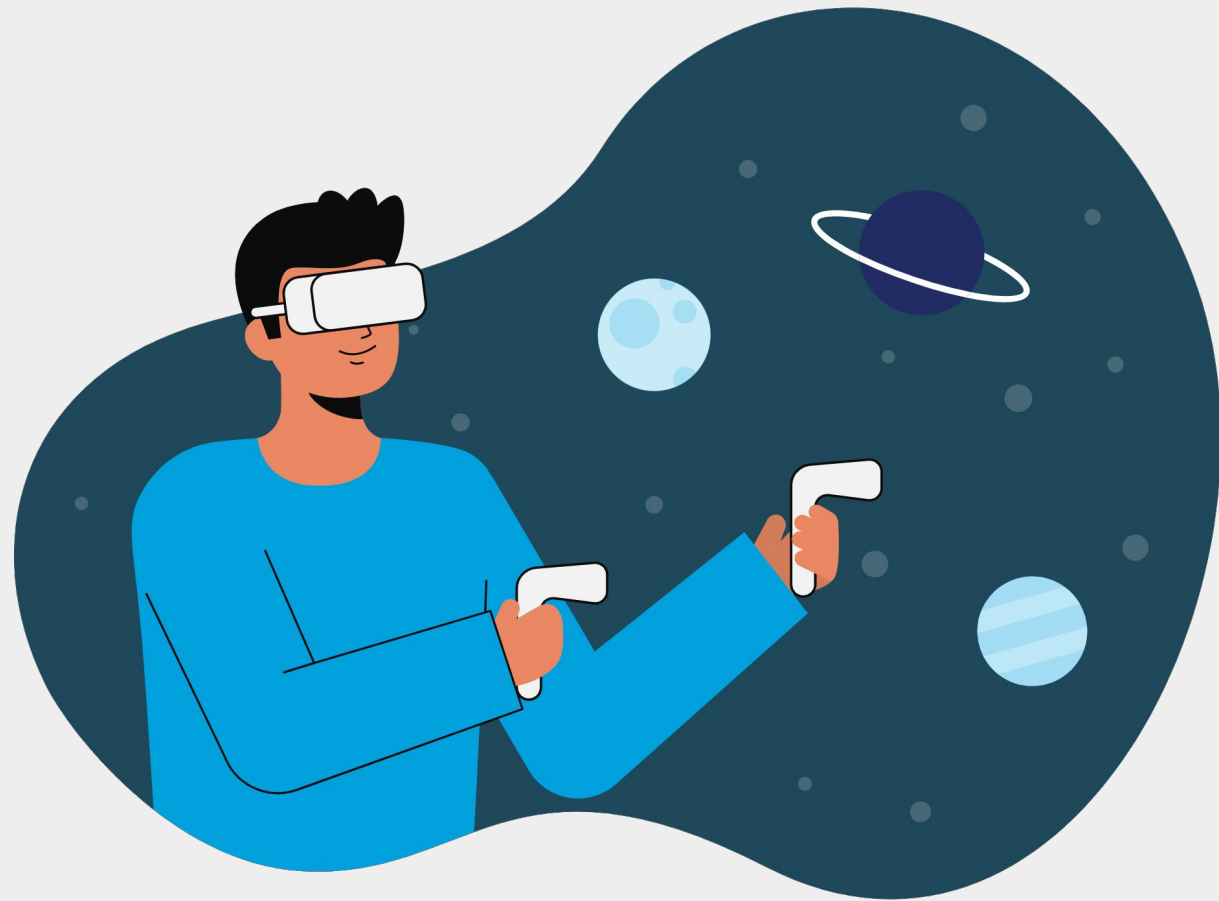
for

Estructuras de control



SPRINT 1

for



for es una de las estructuras de control de Javascript más conocidas.

for permite ejecutar un bloque de código una cantidad determinada de veces.

for es un tipo de bucle, loop o iterador.

Sintaxis de for

for recibe 3 parámetros separados por (;) cuando de invoca.



```
for (let i = 0; i < 3; i++) {  
  // Lo que ejecutamos en cada iteración.  
}
```



```
for (inicializar variable; condición; incremento) {  
  // Lo que ejecutamos en cada iteración.  
}
```

- 1) En el primer parámetro inicializamos la variable i en 0.
- 2) En el segundo parámetro establecemos la condición de ejecución (mientras i sea menor que 3).
- 3) En el tercer parámetro incrementamos el valor de i en 1 al terminar el ciclo.

Ejemplo básico de for

El bucle se ejecutará hasta que no cumpla con la condición, es decir, el console.log imprimirá 3 veces.

```
for (let i = 0; i < 3; i++) {  
  console.log(i)  
}
```

0

1

2

Jugando con for

¿Qué pasará en este caso?



```
for (let i = 0; i !== "a"; i++) {  
  console.log(i)  
}
```

break en for

Si ejecutamos break dentro de un for saldremos del ciclo.

```
for (let i = 0; i < 10; i++) {  
  if (i === 4) {  
    break  
  }  
  console.log(i)  
}
```

0

1

2

3

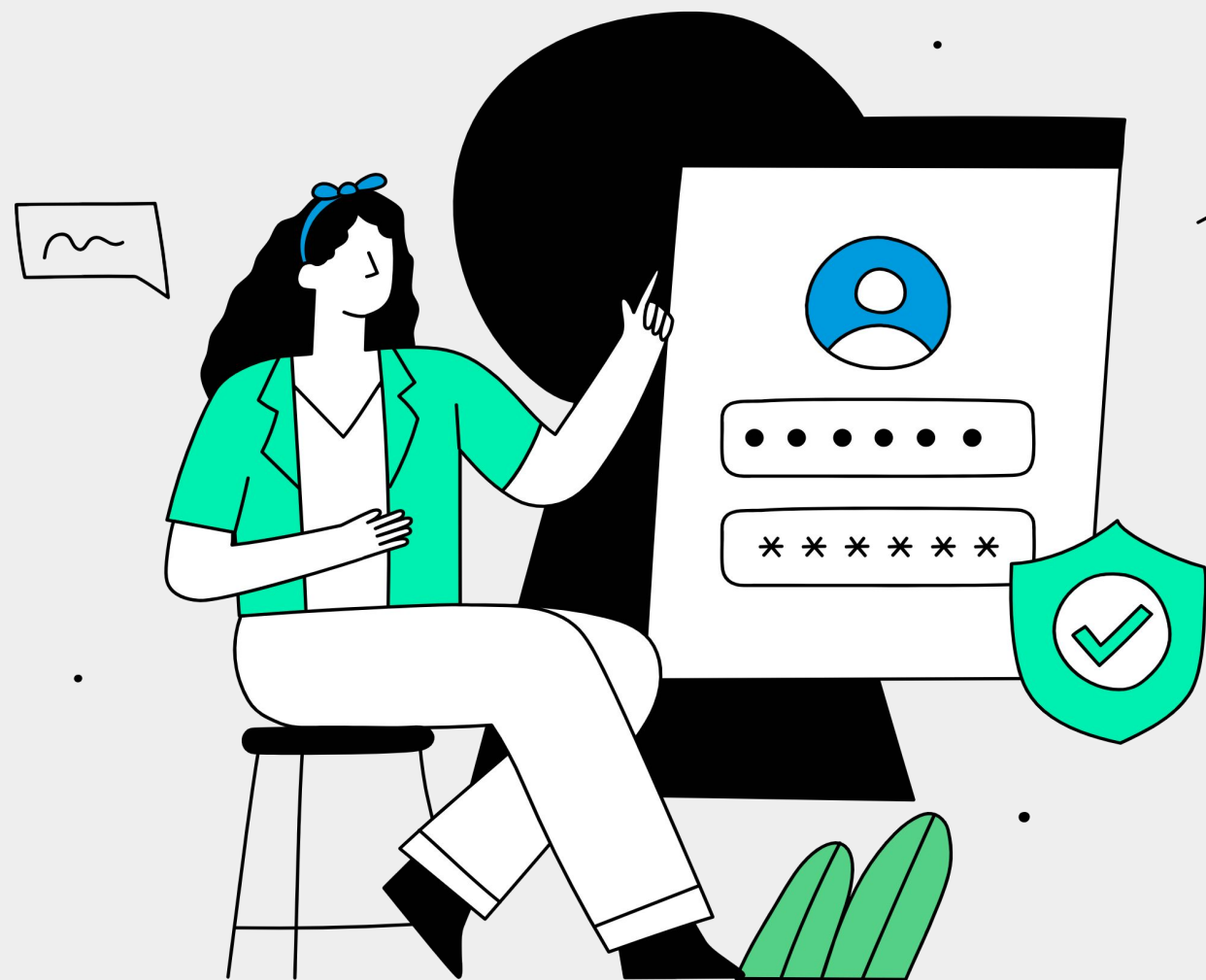
while

Estructuras de control



SPRINT 1

while



while es un bucle, tal como for, pero los datos con los cuáles evalúa la condición, en general, provienen de su exterior.

No se sabe cuándo saldremos del while.

Sintaxis de while



```
let i = 0

while (i < 5) {
  console.log(i)
  i++
}
```



Inicialización

```
while (Condición) {
  console.log(i)
  Incremento
}
```

Mientras se cumpla la condición el while se ejecutará.

Ejemplo básico de while

El bucle se ejecutará hasta que no cumpla con la condición, es decir, el console.log imprimirá 3 veces.



```
let x = 0;

while (x < 3) {
  console.log(x);
  x++;
}
```

0

1

2

Jugando con while

¿Qué pasará en este caso?



```
let x = 0;  
  
while (true) {  
  console.log(x);  
}
```


while con prompt



```
let homer = prompt("¿Cuál es su apellido?");  
  
while (homer !== "Thompson") {  
  alert("Simpson");  
  homer = prompt("¿Cuál es su apellido?");  
}
```



Array push para la entrega

```
const array = [1, 2, 3];  
  
array.push(0);  
  
console.log(array); [ 1, 2, 3, 0 ]
```



Tiranosaurio, 67 millones de años antes de nuestra era.



¡Manos a la obra!

SPRINT 1

Ejercicio Demo

Crear un prompt con varias opciones:
Administrar las opciones con un switch.

Opciones:

1. Pedir 2 números y sumarlos. Deben ser enteros positivos sino mostrar un alert de error.
2. Pedir un número mayor a 10 y mediante un for hacer una cuenta regresiva hasta 0.
3. Pedir un nombre y una edad e interpolarlas.

Los resultados de las opciones se ven en consola.

El programa arranca al hacer F5, con la consola abierta.



retro

¿Cómo nos fué?

¿Qué cosas no quedaron claras y
necesitamos repasar la próxima?

