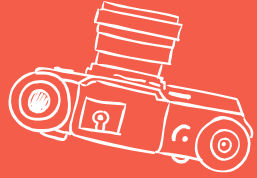


# PROGRAMACIÓN IMPERATIVA EN JAVASCRIPT



# PROGRAMA

1. ENTORNO DE PROGRAMACIÓN
2. PROGRAMAR EN EL EDITOR
3. VARIABLES, CONSTANTES Y SUS TIPOS
4. OPERACIONES BÁSICAS
5. CONDICIONALES
6. CICLOS
7. FUNCIONES



# REPASO DE LA CLASE ANTERIOR

- Algunas funciones JavaScript
  - `console.log`, `JSON.stringify`, `typeof`, `Math.*`
- Condicionales:
  - `if - else`
  - `if`
  - `else if`
- Operadores de comparación
- Operadores lógicos: `AND`, `OR`, `NOT`
- Switch cases

# OPERADOR TERNARIO

( condición ) ? ejecuta este código : ejecuta este código en su lugar

// Ejemplo real

```
let greeting = ( isBirthday ) ? 'Happy birthday Mrs. Smith -- we hope you  
have a great day!' : 'Good morning Mrs. Smith.';
```

# OPERADOR TERNARIO: EJEMPLO

TEST.HTML

```
<label for="theme">Select theme: </label>
<select id="theme">
  <option value="white">White</option>
  <option value="black">Black</option>
</select>
<h1>This is my website</h1>
```

TEST.JS

```
let select = document.querySelector('select');
let html = document.querySelector('html');
document.body.style.padding = '10px';

function update(bgColor, textColor) {
  html.style.backgroundColor = bgColor;
  html.style.color = textColor;
}

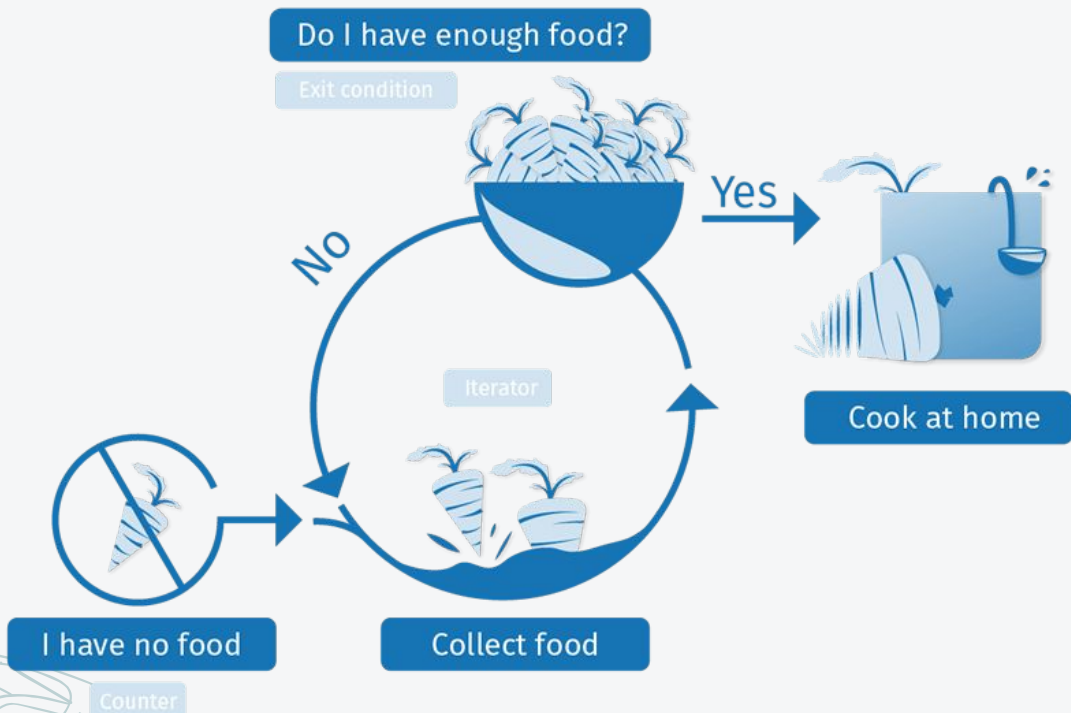
select.onchange = function() {
  ( select.value === 'black' ) ?
  update('black', 'white') : update('white', 'black');
}
```

LIVE CODE

# 6. CICLOS



# EJEMPLO DE LA VIDA REAL



# PSEUDOCÓDIGO DEL AGRICULTOR

```
bucle(comida = 0; comidaNecesaria = 10) {  
  if (comida = comidaNecesaria) {  
    salida bucle;  
    // Tenemos suficiente comida; vamos para casa  
  } else {  
    comida += 2; // Pasar una hora recogiendo 2 más de comida  
    // Comenzar el bucle de nuevo  
  }  
}
```



# EJEMPLO REAL EN CÓDIGO

¿ Por qué son importantes los ciclos?

LIVE CODE

```
for (let i = 0; i < 100; i++) {  
  ctx.beginPath();  
  ctx.fillStyle = 'rgba(255,0,0,0.5)';  
  ctx.arc(random(WIDTH), random(HEIGHT), random(50), 0, 2 * Math.PI);  
  ctx.fill();  
}
```

# BUCLE ESTÁNDAR FOR

```
for (contador = valorInicial; condición de salida; expresión final) {  
    // código a ejecutar  
}
```

# EJEMPLO REAL: FOR

```
let cats = ['Bill', 'Jeff', 'Pete', 'Biggles', 'Jasmin'];  
let info = 'My cats are called: '  
  
for (let i = 0; i < cats.length; i++) {  
  info += cats[i] + ', '  
}  
  
console.log(info);
```

## EJEMPLO REAL: FOR

```
let cats = ['Bill', 'Jeff', 'Pete', 'Biggles', 'Jasmin'];  
let info = 'My cats are called: '  
  
for (let i = 0; i < cats.length; i++) {  
  info += cats[i] + ', '  
}  
  
console.log(info);
```

## EJEMPLO REAL: FOR

```
let cats = ['Bill', 'Jeff', 'Pete', 'Biggles', 'Jasmin'];
let info = 'My cats are called: ';

for (let i = 0; i < cats.length; i++) {
  if (i === cats.length - 1) {
    info = info.slice(0, info.length - 2)
    info += ' and ' + cats[i] + '.';
  } else {
    info += cats[i] + ', ';
  }
}

console.log(info);
```

# SALIR DE UN BUCLE: BREAK

```
for (...) {  
    if (encontre lo que busco) {  
        hago algo;  
        break;  
    } else {  
        sigo buscando  
    }  
}
```

# EJEMPLO REAL: BREAK

```
let contacts = ['Chris:2232322', 'Sarah:3453456', 'Bill:7654322',  
  'Mary:9998769', 'Dianne:9384975'];  
let searchName = "Sarah";  
let text = "";  
  
for (let i = 0; i < contacts.length; i++) {  
  let splitContact = contacts[i].split(':');  
  if (splitContact[0] === searchName) {  
    text = splitContact[0] + '\s number is ' + splitContact[1] + '.';  
    break;  
  } else {  
    text = 'Contact not found.';  
  }  
}  
  
console.log(text);
```

LIVE CODE

# SALTEAR ITERACIONES: CONTINUE

```
for (...) {  
    if (no me interesa este caso) {  
        continue;  
    } else { // me interesa este caso  
        hago algo con los datos  
    }  
}
```



# EJEMPLO REAL: CONTINUE

```
let num = 25;
let text = '';

for (let i = 1; i <= num; i++) {
  let sqRoot = Math.sqrt(i);
  if (Math.floor(sqRoot) !== sqRoot) {
    continue;
  }

  text += i + ' ';
}
```

LIVE CODE

# WHILE

```
initializer  
while (exit-condition) {  
    // code to run  
  
    final-expression  
}
```

# EJEMPLO REAL: WHILE

```
let i = 0;

while (i < cats.length) {
  if (i === cats.length - 1) {
    info += 'and ' + cats[i] + '.';
  } else {
    info += cats[i] + ', ';
  }

  i++;
}
```

# DO ... WHILE

```
initializer  
do {  
    // code to run  
  
    final-expression  
} while (exit-condition)
```

## EJEMPLO REAL: DO ... WHILE

```
let i = 0;

do {
  if (i === cats.length - 1) {
    info += 'and ' + cats[i] + '.';
  } else {
    info += cats[i] + ', ';
  }

  i++;
} while (i < cats.length);
```