

Introducción a la Programación

La programación es el proceso de escribir instrucciones que una computadora puede entender y ejecutar. Estas instrucciones se escriben en lenguajes de programación como JavaScript.

Variables

Una **variable** es un espacio en la memoria de la computadora donde podemos almacenar datos que pueden cambiar a lo largo del programa. En JavaScript, declaramos variables utilizando `let`, `const` o `var`.

Declaración y Asignación de Variables

```
javascript
Copiar código
let edad = 25;           // Declarar una variable llamada edad y asignarle
                          // el valor 25
const nombre = "Ana";    // Declarar una constante llamada nombre y
                          // asignarle el valor "Ana"
var ciudad = "Madrid";   // Declarar una variable llamada ciudad y
                          // asignarle el valor "Madrid"
```

- `let` y `const` son las formas modernas de declarar variables. `const` se utiliza para valores que no cambiarán.
- `var` es una forma más antigua y tiene un comportamiento diferente que generalmente no es preferible en programas modernos.

Procesos Selectivos (Condicionales)

Los procesos selectivos permiten tomar decisiones en base a condiciones. La estructura más común es el `if`.

Sintaxis General

```
javascript
Copiar código
if (condición) {
    // código a ejecutar si la condición es verdadera
} else {
    // código a ejecutar si la condición es falsa
}
```

Ejemplo en JavaScript

```
javascript
Copiar código
let edad = 20;

if (edad >= 18) {
    console.log("Eres mayor de edad.");
} else {
    console.log("Eres menor de edad.");
}
```

Explicación:

1. Se declara una variable `edad` con el valor 20.
2. Si `edad` es mayor o igual a 18, se imprime "Eres mayor de edad."
3. Si `edad` es menor que 18, se imprime "Eres menor de edad."

Bucles (Ciclos)

Los bucles permiten ejecutar un bloque de código múltiples veces. Hay varios tipos de bucles: `for`, `while` y `do-while`.

Bucle `for`

El bucle `for` se usa cuando sabemos cuántas veces queremos repetir una acción.

Sintaxis General

```
javascript
Copiar código
for (inicialización; condición; incremento/decremento) {
    // código a ejecutar
}
```

Ejemplo en JavaScript

```
javascript
Copiar código
for (let i = 0; i < 5; i++) {
    console.log("Esta es la iteración número", i);
}
```

Explicación:

1. **Inicialización:** `let i = 0` inicializa la variable `i` a 0.
2. **Condición:** `i < 5` el bucle se ejecuta mientras `i` sea menor que 5.
3. **Incremento:** `i++` incrementa `i` en 1 después de cada iteración.

Este bucle imprimirá:

```
Copiar código
Esta es la iteración número 0
Esta es la iteración número 1
Esta es la iteración número 2
Esta es la iteración número 3
Esta es la iteración número 4
```

Bucle `while`

El bucle `while` se usa cuando no sabemos cuántas veces se ejecutará el bucle y depende de una condición que se evalúa antes de cada iteración.

Sintaxis General

```
javascript
Copiar código
while (condición) {
    // código a ejecutar
}
```

Ejemplo en JavaScript

```
javascript
Copiar código
let i = 0;
while (i < 5) {
    console.log("Esta es la iteración número", i);
    i++;
}
```

Explicación:

1. **Condición:** $i < 5$ el bucle se ejecuta mientras i sea menor que 5.
2. **Código a ejecutar:** imprime el valor de i y luego incrementa i en 1.

Este bucle imprimirá el mismo resultado que el ejemplo del bucle `for`.

Bucle do-while

El bucle `do-while` es similar al `while`, pero siempre se ejecuta al menos una vez porque la condición se evalúa al final.

Sintaxis General

```
javascript
Copiar código
do {
    // código a ejecutar
} while (condición);
```

Ejemplo en JavaScript

```
javascript
Copiar código
let i = 0;
do {
    console.log("Esta es la iteración número", i);
    i++;
} while (i < 5);
```

Explicación:

1. **Código a ejecutar:** imprime el valor de i y luego incrementa i en 1.
2. **Condición:** $i < 5$ el bucle se sigue ejecutando mientras i sea menor que 5.

Ejemplos Prácticos

Ejemplo 1: Sumar números del 1 al 10

```
javascript
Copiar código
let suma = 0;
for (let i = 1; i <= 10; i++) {
    suma += i;
}
console.log("La suma es", suma);
```

Explicación: Este bucle suma los números del 1 al 10 y almacena el resultado en la variable `suma`.

Ejemplo 2: Contar números pares hasta 10 usando `while`

```
javascript
Copiar código
let i = 0;
let contador = 0;
while (i <= 10) {
    if (i % 2 === 0) {
        contador++;
    }
    i++;
}
console.log("Hay", contador, "números pares hasta 10");
```

Explicación: Este bucle cuenta cuántos números pares hay desde 0 hasta 10.

Resumen

Entender variables, condiciones y bucles es fundamental para aprender a programar. Las variables almacenan datos, las condiciones permiten tomar decisiones y los bucles facilitan la repetición de tareas. Con práctica y ejemplos, tus estudiantes dominarán estos conceptos esenciales en poco tiempo.

Ejercicios Básicos de Procesos Selectivos (Condiciones)

Ejercicio 1: Determinar si un número es par o impar

Problema: Determinar si un número es par o impar.

Solución paso a paso:

1. Declarar una variable para almacenar el número.
2. Usar una condición `if` para verificar si el número es divisible por 2.
3. Si la condición es verdadera, el número es par.
4. Si la condición es falsa, el número es impar.

Código en JavaScript:

```
javascript
Copiar código
let numero = 4;

if (numero % 2 === 0) {
    console.log(numero + " es par");
} else {
    console.log(numero + " es impar");
}
```

Explicación:

- `numero % 2 === 0` verifica si el número es divisible por 2.
- Si la condición es verdadera, el número es par.
- Si no, el número es impar.

Ejercicio 2: Determinar si una persona es mayor de edad

Problema: Determinar si una persona es mayor de edad.

Solución paso a paso:

1. Declarar una variable para almacenar la edad.
2. Usar una condición `if` para verificar si la edad es mayor o igual a 18.
3. Si la condición es verdadera, la persona es mayor de edad.
4. Si la condición es falsa, la persona es menor de edad.

Código en JavaScript:

```
javascript
Copiar código
let edad = 20;

if (edad >= 18) {
    console.log("Eres mayor de edad");
} else {
    console.log("Eres menor de edad");
}
```

Explicación:

- `edad >= 18` verifica si la edad es mayor o igual a 18.
- Si la condición es verdadera, la persona es mayor de edad.
- Si no, la persona es menor de edad.

Ejercicio 3: Determinar si un número es positivo, negativo o cero

Problema: Determinar si un número es positivo, negativo o cero.

Solución paso a paso:

1. Declarar una variable para almacenar el número.
2. Usar una condición `if-else if-else` para verificar si el número es positivo, negativo o cero.

Código en JavaScript:

```
javascript
Copiar código
let numero = -5;

if (numero > 0) {
  console.log(numero + " es positivo");
} else if (numero < 0) {
  console.log(numero + " es negativo");
} else {
  console.log(numero + " es cero");
}
```

Explicación:

- `numero > 0` verifica si el número es positivo.
- `numero < 0` verifica si el número es negativo.
- Si ninguna de las condiciones anteriores es verdadera, el número es cero.

Ejercicio 4: Determinar si un año es bisiesto

Problema: Determinar si un año es bisiesto.

Solución paso a paso:

1. Declarar una variable para almacenar el año.
2. Usar una condición `if-else` para verificar si el año es divisible por 4, no divisible por 100, o divisible por 400.

Código en JavaScript:

```
javascript
Copiar código
let anio = 2020;

if ((anio % 4 === 0 && anio % 100 !== 0) || anio % 400 === 0) {
```

```
    console.log(anio + " es un año bisiesto");
} else {
    console.log(anio + " no es un año bisiesto");
}
```

Explicación:

- `(anio % 4 === 0 && anio % 100 !== 0) || anio % 400 === 0` verifica las reglas de un año bisiesto.

Ejercicio 5: Determinar el mayor de dos números

Problema: Determinar el mayor de dos números.

Solución paso a paso:

1. Declarar dos variables para almacenar los números.
2. Usar una condición `if-else` para verificar cuál número es mayor.

Código en JavaScript:

```
javascript
Copiar código
let numero1 = 10;
let numero2 = 20;

if (numero1 > numero2) {
    console.log(numero1 + " es mayor que " + numero2);
} else if (numero1 < numero2) {
    console.log(numero2 + " es mayor que " + numero1);
} else {
    console.log("Ambos números son iguales");
}
```

Explicación:

- `numero1 > numero2` verifica si el primer número es mayor.
- `numero1 < numero2` verifica si el segundo número es mayor.
- Si ninguna de las condiciones anteriores es verdadera, ambos números son iguales.

Ejercicios con Bucles

Ejercicio 1: Presentar los números pares de n números

Problema: Presentar los números pares hasta n.

Solución paso a paso:

1. Declarar una variable para n.
2. Usar un bucle `for` para iterar desde 1 hasta n.
3. Dentro del bucle, usar una condición `if` para verificar si el número es par.

Código en JavaScript:

```
javascript
Copiar código
let n = 10;

for (let i = 1; i <= n; i++) {
  if (i % 2 === 0) {
    console.log(i + " es par");
  }
}
```

Explicación:

- `for (let i = 1; i <= n; i++)` itera desde 1 hasta n.
- `if (i % 2 === 0)` verifica si i es par.

Ejercicio 2: Presentar los divisores de un número

Problema: Presentar los divisores de un número.

Solución paso a paso:

1. Declarar una variable para el número.
2. Usar un bucle `for` para iterar desde 1 hasta el número.
3. Dentro del bucle, usar una condición `if` para verificar si el número es divisible por el iterador.

Código en JavaScript:

```
javascript
Copiar código
let numero = 20;

for (let i = 1; i <= numero; i++) {
  if (numero % i === 0) {
    console.log(i + " es un divisor de " + numero);
  }
}
```

Explicación:

- `for (let i = 1; i <= numero; i++)` itera desde 1 hasta numero.
- `if (numero % i === 0)` verifica si i es un divisor de numero.

Ejercicio 3: Verificar si un número es perfecto

Problema: Verificar si un número es perfecto. Un número es perfecto si es igual a la suma de sus divisores propios (excluyendo el propio número).

Solución paso a paso:

1. Declarar una variable para el número.

2. Usar un bucle `for` para encontrar los divisores propios y sumarlos.
3. Usar una condición `if` para verificar si la suma de los divisores es igual al número.

Código en JavaScript:

```
javascript
Copiar código
let numero = 28;
let suma = 0;

for (let i = 1; i < numero; i++) {
  if (numero % i === 0) {
    suma += i;
  }
}

if (suma === numero) {
  console.log(numero + " es un número perfecto");
} else {
  console.log(numero + " no es un número perfecto");
}
```

Explicación:

- `for (let i = 1; i < numero; i++)` itera desde 1 hasta `numero - 1`.
- `if (numero % i === 0)` verifica si `i` es un divisor.
- `suma += i` suma los divisores.
- `if (suma === numero)` verifica si la suma de los divisores es igual al número.

Ejercicio 4: Verificar si un número es primo

Problema: Verificar si un número es primo. Un número es primo si solo tiene dos divisores: 1 y el mismo número.

Solución paso a paso:

1. Declarar una variable para el número.
2. Usar un bucle `for` para encontrar los divisores.
3. Usar una condición `if` para verificar si tiene más de dos divisores.

Código en JavaScript:

```
javascript
Copiar código
let numero = 29;
let esPrimo = true;

for (let i = 2; i < numero; i++) {
  if (numero % i === 0) {
    esPrimo = false;
    break;
  }
}
```

```
if (esPrimo) {
    console.log(numero + " es un número primo");
} else {
    console.log(numero + " no es un número primo");
}
```

Explicación:

- `for (let i = 2; i < numero; i++)` itera desde 2 hasta `numero - 1`.
- `if (numero % i === 0)` verifica si `i` es un divisor.
- `esPrimo = false` establece que el número no es primo si se encuentra un divisor.
- `if (esPrimo)` verifica si el número es primo.

Ejercicio 5: Invertir un número dividiendo para diez

Problema: Invertir un número utilizando operaciones de división y módulo.

Solución paso a paso:

1. Declarar una variable para el número.
2. Usar un bucle `while` para extraer y reordenar los dígitos.

Código en JavaScript:

```
javascript
Copiar código
let numero = 12345;
let invertido = 0;

while (numero !== 0) {
    let digito = numero % 10;
    invertido = invertido * 10 + digito;
    numero = Math.floor(numero / 10);
}

console.log("El número invertido es " + invertido);
```

Explicación:

- `while (numero !== 0)` itera mientras `numero` no sea 0.
- `let digito = numero % 10` obtiene el último dígito del número.
- `invertido = invertido * 10 + digito` agrega el dígito al número invertido.
- `numero = Math.floor(numero / 10)` elimina el último dígito del número original.

Ejercicio 6: Multiplicación de dos números por sumas sucesivas

Problema: Multiplicar dos números utilizando sumas sucesivas.

Solución paso a paso:

1. Declarar dos variables para los números.

2. Usar un bucle `for` para sumar el primer número consigo mismo el número de veces del segundo número.

Código en JavaScript:

```
javascript
Copiar código
let numero1 = 4;
let numero2 = 3;
let resultado = 0;

for (let i = 0; i < numero2; i++) {
    resultado += numero1;
}

console.log("El resultado de la multiplicación es " + resultado);
```

Explicación:

- `for (let i = 0; i < numero2; i++)` itera `numero2` veces.
- `resultado += numero1` suma `numero1` a `resultado` en cada iteración.

Ejercicio 7: División de dos números por restas sucesivas

Problema: Dividir dos números utilizando restas sucesivas.

Solución paso a paso:

1. Declarar dos variables para los números.
2. Usar un bucle `while` para restar el divisor del dividendo hasta que el dividendo sea menor que el divisor.

Código en JavaScript:

```
javascript
Copiar código
let dividendo = 20;
let divisor = 4;
let cociente = 0;

while (dividendo >= divisor) {
    dividendo -= divisor;
    cociente++;
}

console.log("El cociente de la división es " + cociente);
```

Explicación:

- `while (dividendo >= divisor)` itera mientras el dividendo sea mayor o igual al divisor.
- `dividendo -= divisor` resta el divisor del dividendo.
- `cociente++` incrementa el cociente en 1 en cada iteración.

Ejercicio 8: Presentar la serie de Fibonacci de n números

Problema: Presentar los primeros n números de la serie de Fibonacci.

Solución paso a paso:

1. Declarar una variable para n.
2. Usar un bucle `for` para generar la serie de Fibonacci.

Código en JavaScript:

```
javascript
Copiar código
let n = 10;
let a = 0;
let b = 1;

console.log(a);
console.log(b);

for (let i = 2; i < n; i++) {
  let c = a + b;
  console.log(c);
  a = b;
  b = c;
}
```

Explicación:

- `console.log(a)` y `console.log(b)` imprimen los dos primeros números de la serie.
- `for (let i = 2; i < n; i++)` itera desde 2 hasta $n - 1$.
- `let c = a + b` calcula el siguiente número de la serie.
- `a = b` y `b = c` actualizan los valores para la siguiente iteración.

Ejercicio 9: Presentar los números múltiplos de 5 de un arreglo

Problema: Presentar los números múltiplos de 5 de un arreglo.

Solución paso a paso:

1. Declarar un arreglo con números.
2. Usar un bucle `for` para iterar a través del arreglo.
3. Usar una condición `if` para verificar si un número es múltiplo de 5.

Código en JavaScript:

```
javascript
Copiar código
let numeros = [10, 15, 23, 30, 45, 50];

for (let i = 0; i < numeros.length; i++) {
  if (numeros[i] % 5 === 0) {
    console.log(numeros[i] + " es múltiplo de 5");
  }
}
```

```
}  
}
```

Explicación:

- `for (let i = 0; i < numeros.length; i++)` itera a través del arreglo.
- `if (numeros[i] % 5 === 0)` verifica si el número es múltiplo de 5.

Ejercicio 10: El promedio de los números de un arreglo

Problema: Calcular el promedio de los números de un arreglo.

Solución paso a paso:

1. Declarar un arreglo con números.
2. Usar un bucle `for` para sumar todos los números.
3. Dividir la suma por la longitud del arreglo.

Código en JavaScript:

```
javascript  
Copiar código  
let numeros = [10, 15, 23, 30, 45, 50];  
let suma = 0;  
  
for (let i = 0; i < numeros.length; i++) {  
    suma += numeros[i];  
}  
  
let promedio = suma / numeros.length;  
console.log("El promedio es " + promedio);
```

Explicación:

- `for (let i = 0; i < numeros.length; i++)` itera a través del arreglo.
- `suma += numeros[i]` suma cada número al total.
- `let promedio = suma / numeros.length` calcula el promedio.

Ejercicio 11: El mayor de los elementos de un arreglo

Problema: Encontrar el mayor de los elementos de un arreglo.

Solución paso a paso:

1. Declarar un arreglo con números.
2. Suponer que el primer elemento es el mayor.
3. Usar un bucle `for` para comparar cada elemento con el supuesto mayor.

Código en JavaScript:

```
javascript  
Copiar código  
let numeros = [10, 15, 23, 30, 45, 50];
```

```
let mayor = numeros[0];

for (let i = 1; i < numeros.length; i++) {
  if (numeros[i] > mayor) {
    mayor = numeros[i];
  }
}

console.log("El mayor número es " + mayor);
```

Explicación:

- `let mayor = numeros[0]` supone que el primer elemento es el mayor.
- `for (let i = 1; i < numeros.length; i++)` itera desde el segundo elemento.
- `if (numeros[i] > mayor)` actualiza `mayor` si se encuentra un número más grande.

Ejercicio 12: Presentar los nombres que no sean "Ana" de un arreglo

Problema: Presentar los nombres que no sean "Ana" de un arreglo.

Solución paso a paso:

1. Declarar un arreglo con nombres.
2. Usar un bucle `for` para iterar a través del arreglo.
3. Usar una condición `if` para verificar si el nombre no es "Ana".

Código en JavaScript:

```
javascript
Copiar código
let nombres = ["Juan", "Ana", "Carlos", "Ana", "Luis"];

for (let i = 0; i < nombres.length; i++) {
  if (nombres[i] !== "Ana") {
    console.log(nombres[i]);
  }
}
```

Explicación:

- `for (let i = 0; i < nombres.length; i++)` itera a través del arreglo.
- `if (nombres[i] !== "Ana")` verifica si el nombre no es "Ana".

Ejercicio 13: Los pares de un arreglo recorridos desde el último elemento al primero

Problema: Presentar los números pares de un arreglo recorriéndolo desde el último elemento al primero.

Solución paso a paso:

1. Declarar un arreglo con números.
2. Usar un bucle `for` para iterar a través del arreglo desde el último elemento al primero.
3. Usar una condición `if` para verificar si el número es par.

Código en JavaScript:

```
javascript
Copiar código
let numeros = [10, 15, 23, 30, 45, 50];

for (let i = numeros.length - 1; i >= 0; i--) {
  if (numeros[i] % 2 === 0) {
    console.log(numeros[i] + " es par");
  }
}
```

Explicación:

- `for (let i = numeros.length - 1; i >= 0; i--)` itera desde el último elemento al primero.
- `if (numeros[i] % 2 === 0)` verifica si el número es par.

Ejercicio 14: Presentar las vocales de una frase

Problema: Presentar las vocales de una frase.

Solución paso a paso:

1. Declarar una variable para almacenar la frase.
2. Usar un bucle `for` para iterar a través de la frase.
3. Usar una condición `if` para verificar si el carácter es una vocal.

Código en JavaScript:

```
javascript
Copiar código
let frase = "Hola, ¿cómo estás?";
let vocales = "aeiouAEIOU";

for (let i = 0; i < frase.length; i++) {
  if (vocales.includes(frase[i])) {
    console.log(frase[i]);
  }
}
```

Explicación:

- `for (let i = 0; i < frase.length; i++)` itera a través de la frase.
- `if (vocales.includes(frase[i]))` verifica si el carácter es una vocal.

Ejercicio 15: Presentar las palabras de una frase, asumiendo un solo espacio entre ellas

Problema: Presentar las palabras de una frase, asumiendo un solo espacio entre ellas.

Solución paso a paso:

1. Declarar una variable para almacenar la frase.
2. Usar el método `split` para dividir la frase en palabras.
3. Usar un bucle `for` para iterar a través de las palabras.

Código en JavaScript:

```
javascript
Copiar código
let frase = "Hola cómo estás";
let palabras = frase.split(" ");

for (let i = 0; i < palabras.length; i++) {
  console.log(palabras[i]);
}
```

Explicación:

- `let palabras = frase.split(" ")` divide la frase en un arreglo de palabras.
- `for (let i = 0; i < palabras.length; i++)` itera a través del arreglo de palabras.

Ejercicio 16: Copiar una frase a otra carácter por carácter excluyendo los dígitos

Problema: Copiar una frase a otra carácter por carácter excluyendo los dígitos.

Solución paso a paso:

1. Declarar una variable para almacenar la frase original.
2. Declarar una variable para la nueva frase vacía.
3. Usar un bucle `for` para iterar a través de la frase original.
4. Usar una condición `if` para verificar si el carácter no es un dígito.

Código en JavaScript:

```
javascript
Copiar código
let fraseOriginal = "Hola 123, cómo estás?";
let nuevaFrase = "";

for (let i = 0; i < fraseOriginal.length; i++) {
  if (!isDigit(fraseOriginal[i])) {
    nuevaFrase += fraseOriginal[i];
  }
}

console.log(nuevaFrase);

function isDigit(char) {
  return char >= '0' && char <= '9';
}
```


Explicación:

- `for (let i = 0; i < fraseOriginal.length; i++)` itera a través de la frase original.
- `if (!isDigit(fraseOriginal[i]))` verifica si el carácter no es un dígito.
- `nuevaFrase += fraseOriginal[i]` agrega el carácter a la nueva frase.

Ejercicio 17: Presentar cuántas consonantes hay en una frase

Problema: Presentar cuántas consonantes hay en una frase.

Solución paso a paso:

1. Declarar una variable para almacenar la frase.
2. Declarar una variable para contar las consonantes.
3. Usar un bucle `for` para iterar a través de la frase.
4. Usar una condición `if` para verificar si el carácter es una consonante.

Código en JavaScript:

```
javascript
Copiar código
let frase = "Hola, ¿cómo estás?";
let consonantes = "bcdfghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ";
let contador = 0;

for (let i = 0; i < frase.length; i++) {
  if (consonantes.includes(frase[i])) {
    contador++;
  }
}

console.log("El número de consonantes es " + contador);
```

Explicación:

- `for (let i = 0; i < frase.length; i++)` itera a través de la frase.
- `if (consonantes.includes(frase[i]))` verifica si el carácter es una consonante.
- `contador++` incrementa el contador de consonantes.