

Fundamentos de Programación

JavaScript
Instrucciones

IES La Nía



Condiciones

- Permiten dividir la ejecución del código dependiendo del valor de la condición, la cual se evalúa a `true` o `false`.
- Las llaves definen bloques de instrucciones

```
if (condición) {  
    // código verdadero  
}
```

```
if (condición) {  
    // código verdadero  
} else {  
    // código falso  
}
```

Operadores en Condiciones

- Operadores:

```
if (a >= 5) {  
    if (respuesta === 'SI')
```

- <, <=, >, >=
- ==, != → comparación (con conversión de tipos)
- ===, !== → comparación estricta (sin conversión)

- Componiendo condiciones

- && → Y
- || → O

Ejercicios

1) (ej201if) Completa el siguiente fragmento

```
var num1 = 5;
var num2 = 8;

if(...) {
    console.log("num1 no es mayor que num2");
}
if(...) {
    console.log("num2 es positivo");
}
if(...) {
    console.log("num1 es negativo o distinto de cero");
}
if(...) {
    console.log("Incrementar en 1 unidad el valor de num1 no
lo hace mayor o igual que num2");
}
```

Condiciones encadenadas

```
if (cond1) {  
    // cond1 verdadero  
} else if (cond2) {  
    // cond2 y no cond1  
} else if (cond3) {  
    // cond3 y no cond1 ni cond2  
} else {  
    // cond1, cond2 y cond3 son falsas  
}
```

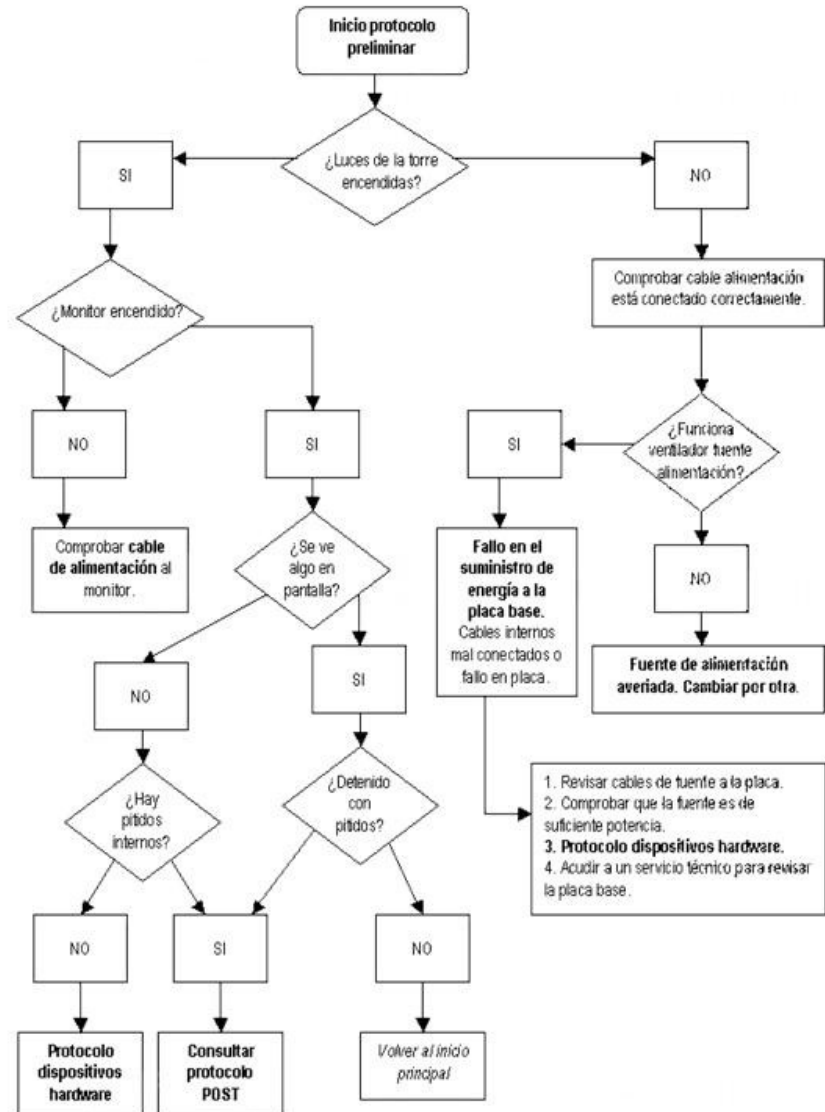
Ejercicios



- 2) (ej202mayor3) Escribe una función que devuelva el mayor de 3 números.
- 3) (ej203mayor4) Escribe una función que devuelva el mayor de 4 números.
- 4) (ej204signo) Escribe una función que indique si un número es “positivo”, “negativo” o “cero”.

Ejercicios

5) (ej205hardware)
Escribe un programa
que utilice el
siguiente árbol de
decisión.
Para ello emplea
confirm(mensaje)



Ejercicios



6) (ej206nombreEdad) Escribir una función que a partir de una edad devuelva:

- **bebé** si tiene menos de 3 años
- **niño** si tiene entre 3 y 12 años
- **adolescente** entre 13 y 17 años
- **adulto** entre 18 y 66
- **jubilado** a partir de 67

Ejercicios



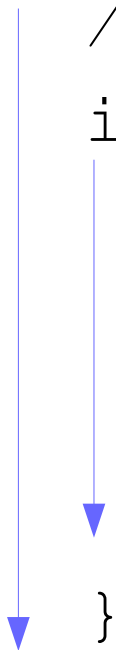
- 7) (`ej207ivaigic`) Escribe un programa que calcule el precio de un producto, sabiendo que en la península el IVA es del 21%, pero en Canarias el IGIC es del 7%. El usuario indicará de algún modo donde se encuentra.
- 8) (`ej208contrasenya`) Escribe un programa que simule el registro de un usuario, de manera que compruebe que la contraseña se introduce correctamente (2 veces la misma)

Condiciones anidadas

```

if (cond1) {
    // cond1 verdadero
    if (cond2) {
        // cond1 y cond2 verdadero
        if (cond3) {
            // cond1, cond2 y cond3 verdadero
        }
    }
}

```



Ejercicios

- 9) (ej209seg1mas) Escribe un programa similar a un reloj de manera que a partir de hora, minuto y segundo devuelva una cadena con la hora dentro de un segundo
- 10) (ej210dia1mas) Escribe un programa similar a un calendario de manera que a partir de día, mes y año devuelva una cadena con la fecha dentro de un día.

switch

- Similar a múltiples `if-else if`

```
switch (condicion) {  
    case valorA:  
        // código A  
        break;  
    case valorB:  
        // código B  
        break;  
    default:  
        // código else  
}
```

- La condición se evalúa una vez. Dependiendo del valor entra en un caso u otro.
- `break` evita que tras un caso se ejecute el siguiente

Ejercicios

11) (ej211letra2num) Escribe una función que a partir de una letra (A-F), devuelva la nota. Nota: hay que usar la instrucción `switch`

- Nota para Lucía: Si no pone NO, es que sí

12) (ej212num2letra) Escribe una función que a partir de la nota, devuelva la letra. Nota: hay que usar la instrucción `switch`.

- (ej212Rnum2letra) mediante `if-else if`

Ejercicios

- 13) (ej 213mes) Escribe una función que a partir de un número, muestre el nombre del mes.
- 14) (ej 214mes18n) Escribe una función que a partir de un número y un idioma (es, ca, ...), muestre el nombre del mes en el idioma indicado.

Condición ternaria

- Permite asignar un valor obtenido a partir de una condición

(condición) ? valorVerdadero : valorFalso;

```
var mayor = (a > b) ? a : b;  
var calificacion = (nota > 5) ? 'apto' :  
    'no apto';
```

Comprobando los resultados

- Hasta ahora, para probar el código, introducimos
`console.log(funcionAProbar());`
- Y verificamos que la salida de la consola es la que esperamos
- Vamos a automatizar las pruebas mediante la librería *QUnit* → ver '035 JS - Librerias.pdf'

Bucles

- Si queremos **repetir** un conjunto de acciones tenemos que usar instrucciones iterativas:
 - `for`
 - `for ... in` → se emplea con objetos o arrays
 - `while`
 - `do / while`

for

- Permite ejecutar instrucciones un número determinado de ocasiones

```
for (inicialización; condiciónInclusión; actualización) {  
    // bloque instrucciones  
}
```

- `inicialización` → valor inicial de la variable sobre la que se itera (1)
- `condiciónInclusión` → *booleano* que decide si continúa el bucle (N)
- `actualización` → modifica la variable de iteración (N)

Ejemplos for

```
var i;      ← i es un contador
```

```
console.log("Del 0 al 9");  
for (i=0; i<10; i++) {  
    console.log(i);  
}
```

← ascendente

```
console.log("Del 9 al 0");  
for (i=9; i>=0; i--) {  
    console.log(i);  
}
```

← descendente

Variables Contador vs Acumulador

- **Contador:** En un bucle, la variable que se incrementa/decrementa.
 - Se inicializa fuera del bucle o en el bloque de inicialización
 - Al finalizar el bucle, ha contado el número de ocurrencias.
 - Nomenclatura: `i`, `j`, `k`, `cont`
- **Acumulador:** en cada iteración del bucle, almacena un cálculo temporal
 - Se inicializa fuera del bucle
 - Si acumula sumas se inicializa a 0
 - Si acumula productos se inicializa a 1
 - Al finalizar el bucle, ha acumulado un resultado en base a todos los elementos.
 - Nomenclatura: `s`, `p`, `aux`

Ejercicios

- 15) (ej215pares050) Escribe una función que muestre los números pares del 0 al 50
- (ej215R) Refactoriza para un rango de números
- 16) (ej216suma110) Escribe una función que calcule la suma de todos los números comprendidos entre el 1 y el 10.
- (ej216R) Refactoriza para un rango de números
- 17) (ej217producto525) Escribe una función que calcule el producto de los números comprendidos entre el 5 y el 25.
- (ej217R) Refactoriza para un rango de números

Ejercicios

- 18) (`ej218producto`) Escribe una función que a partir de dos números, devuelva el producto calculado mediante la acumulación de sumas.
- 19) (`ej219potencia`) Escribe una función que calcule la potencia a partir de la base y el exponente, mediante la acumulación de productos.
- (`ej219R`) Refactoriza para que funcione si el exponente es negativo. Recuerda:

$$a^{-n} = \frac{1}{a^n}$$

Ejercicios

- 20) (`ej220suma5`) Escribe un programa que le pida al usuario 5 números y calcule y muestre el total.
- 21) (`ej221sumaN`) Basándote en el ejercicio anterior, primero obtén cuantos números tiene que leer. Comprueba que los datos leídos son correctos, de manera que si lee un texto o un valor incorrecto lo ignore. Es decir, si el usuario indica que va a leer 5 números, sólo realiza 5 lecturas.
- 22) (`ej222sumaNconMlecturas`) Modifica el programa anterior para que realice las lecturas necesarias para sumar los N números.

Ejercicios

20) (ej223karvonen) La formula *Karvonen* permite obtener el ritmo de pulsaciones cardíacas dependiendo de la intensidad de trabajo para pasar de carga aeróbica a anaeróbica.

```
pulsaciones = (((220 - edad) -  
pulsacionesDescanso) * intensidad) +  
pulsacionesDescanso
```

Crea un programa que muestre una tabla con intensidades del 50 al 100%.

"Edad:30 Pulsaciones Descanso:65 ppm"	
"Intensidad"	"Pulsaciones"
"50%"	127.5 ppm"
"55%"	133.75 ppm"
"60%"	140 ppm"
"65%"	146.25 ppm"
"70%"	152.5 ppm"
"75%"	158.75 ppm"
"80%"	165 ppm"
"85%"	171.25 ppm"
"90%"	177.5 ppm"
"95%"	183.75 ppm"
"100%"	190 ppm"

Bucles anidados

- Podemos incluir un bucle dentro de otro bucle
- Permite trabajar con matrices
- N dimensiones → N bucles

```
var r = "";
for (var i=0;i<3;i++) {
  for (var j=0;j<5;j++) {
    r = r + " *";
  }
  r = r + "\n";
}

console.log(r);
```

← Cuadrado de 3x5

Ejercicios

- 24) (ej224cuadradoN) Escribe un programa dibuje un cuadrado de lado N
- 25) (ej225rectanguloBH) Escribe un programa dibuje un rectángulo de base B y altura H.
- 26) (ej226bordes) Escribe un programa que dibuje un cuadrado de lado N con los bordes con 0 y el relleno con 1.
- ej226R: el usuario elige el carácter de borde y el de relleno

0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

Ejercicios

- 27) (ej227equis) Escribe un programa dibuje una X con 1s mediante un cuadrado impar de lado N
- 28) (ej228hache) Escribe un programa dibuje una H con 1s mediante un cuadrado impar de lado N
- 29) (ej229u) Escribe un programa dibuje una U con 1s mediante un cuadrado impar de lado N
- 30) (ej230ygriega) Escribe un programa que dibuje una Y con 1s mediante un cuadrado impar de lado N
- 31) (ej231eme) Escribe un programa que dibuje una M con 1s mediante un cuadrado impar de lado N

Ejercicios

- 32) (ej232triangulo) Escribe un programa que dibuje un triangulo rectángulo de lado L.
- 33) (ej233tribordes) Escribe un programa que dibuje un triangulo rectángulo de lado L con los bordes con 1s y el relleno con 0s
- 34) (ej234triangulo) Escribe un programa que dibuje un triangulo rectángulo invertido de lado L.



ej229



ej231

Ejercicios

- 35) (ej235tablaMult) Escribe un programa que a partir de un número, muestre su tabla de multiplicar (del 0 al 10).
- 36) (ej236tablaMult) Crea un programa que muestre un cuadrado con las tablas de multiplicar

x	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	10
2	0	2	4	6	8	10	12	14	16	18	20
3	0	3	6	9	12	15	18	21	24	27	30
4	0	4	8	12	16	20	24	28	32	36	40
5	0	5	10	15	20	25	30	35	40	45	50
6	0	6	12	18	24	30	36	42	48	54	60
7	0	7	14	21	28	35	42	49	56	63	70
8	0	8	16	24	32	40	48	56	64	72	80
9	0	9	18	27	36	45	54	63	72	81	90
10	0	10	20	30	40	50	60	70	80	90	100

Problema Olimpiada

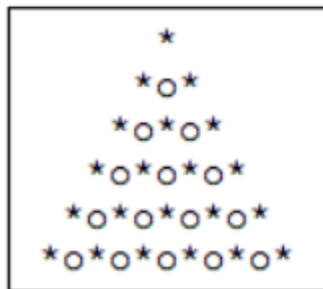
Problema 1: Dibujar figura (20 puntos)

Debes realizar un programa que visualice por pantalla una figura similar a la mostrada en la imagen.

En primer lugar el programa debe preguntar al usuario el número de líneas que debe tener la figura.

A continuación el programa debe mostrar la figura con ese número de líneas.

Ten en cuenta que la solución que propongamos debe contener sentencias de salida por pantalla en las que se impriman exclusivamente uno solo de los siguientes caracteres: blanco ' ', asterisco '*' o la letra 'o'.



Img 1. Ejemplo para 6 filas.

while

- Permite ejecutar instrucciones mientras se cumple una condición
 - Dentro del bucle en algún momento hay que modificar la variable que fija la condición para no provocar un bucle infinito

```
while (condición) {  
    // bloque instrucciones  
}
```

- condición → *booleano* que decide si continúa el bucle (N)

Ejercicios

- 37) (`ej237productoRango`) Rescribe el ejercicio 217 pero haciendo uso de un bucle `while`
- 38) (`ej238leerSI`) Escribe una función que lea desde el *prompt* una cadena, y le pida al usuario que escriba SI para continuar. Si no lo escribe, que le pregunte de manera indefinida.
- 39) (`ej239menu`) Escribe una función que muestre un menú, lea una opción y realice la opción.
- S – SUMAR
 - R – RESTAR
 - X – SALIR

Ejercicios

- 40) (`ej240sumaHasta0`) Escribe una función que sume todos los números hasta introducir el 0.
- 41) (`ej241paresImparesHasta0`) Escribe una función que cuente la cantidad de números pares e impares entrados por teclado. Se acaba cuando el usuario introduce un 0 (No hay que contar el 0).
- 42) (`ej242mayorHasta0`) Escribe una función que averigüe el mayor número entrado por el usuario antes de teclear un 0.
- (`ej242RmenorHasta0`) Igual pero mostrando el menor

do-while

- Permite ejecutar instrucciones mientras se cumple una condición, ejecutándose al menos una vez
 - Dentro del bucle en algún momento hay que modificar la variable que fija la condición para no provocar un bucle infinito

```
do {  
    // bloque instrucciones  
} while (condición)
```

- condición \rightarrow *booleano* que decide si continúa el bucle (N)

Ejercicios

- 43) (`ej243notaMediaHastaNeg`) Escribe una función que devuelva la nota media de un alumno. Para terminar se introducirá un número negativo. Si el alumno no tienen nota alguna, entonces en lugar de escribir la media devolverá “No Presentado”.
- 44) (`ej244primo`) Escribe una función que a partir de un número, devuelva un *booleano* indicando si el número es primo.

Ejercicios

Vamos a refactorizar el ejercicio 106 para que:

- (ej106R1) Compruebe que el usuario sólo introduce números
- (ej106R2) Si el usuario introduce un valor no numérico, mostrar un mensaje apropiado y salir del programa
- (ej106R3) En vez salir, volver a pedir al usuario que introduce la entrada adecuada
- (ej106R4) No permitir que el usuario introduzca un valor negativo
- (ej106R5) Separar los cálculos a una función

Recursividad

- Una función es recursiva cuando la solución de un problema se expresa mediante una llamada a sí misma.
- Sustituye a la iteración
- Compuesto de:
 - 1 caso base (finalización)
 - 1 caso recursivo (llamada a sí mismo)



Potencia Recursiva

- Potencia $\rightarrow a^n = a * a^{n-1}$

```
function potencia(base, exp) {  
    if (exp == 1) {  
        return base;  
    } else {  
        return base * potencia(base, exp-1);  
    }  
}
```

Ejercicios

23) (ej250Factorial) Realiza una función que devuelva el factorial de un número

$$n! = n * (n-1)!$$

$$0! = 1$$

24) (ej251Fibonacci) Realiza una función que devuelva el número fibonacci de un número, siendo

$$f_0 = 1 \text{ y } f_1 = 1$$

$$f_n = f_{n-1} + f_{n-2}$$

