



# JS | Ejercicios

Llega la hora de poner en marcha los conocimientos adquiridos hasta ahora, para ello os proponemos una serie de ejercicios en los que veréis la utilidad de lo que hemos visto hasta ahora.

Existen 2 maneras de abordar los ejercicios, ejecutándolos con Node (como vimos en el prework), o bien cómo hemos hecho con el CSS, linkando un fichero de javascript a un documento HTML, para poder ver el resultado en nuestro navegador (Chrome).

## Ejercicio 1 → Ponte en marcha

Te dejamos el esqueleto del *ejercicio1.js* comenzado, para que poco a poco vayas completando y cambiando lo que necesites. En este caso se trata de que termines las condiciones que encontrarás dentro de la función llamada `miFuncionDeComparar`.


Debes conseguir que los mensajes que se muestren por la consola sean siempre correctos, sin importar los números que le pases por parámetro a la función

```
(numero1, numero2).
```

```
function miFuncionDeComparar(numero1, numero2) {  
  // Recuerda completar los condicionales 'if'  
  
  if (numero1 === 5) {  
    console.log("numero1 es estrictamente igual a 5");  
  }  
  
  if (/* COMPLETAR */) {  
    console.log("numero1 no es mayor que numero2");  
  }  
  if (/* COMPLETAR */) {  
    console.log("numero2 es positivo");  
  }  
  if (/* COMPLETAR */) {  
    console.log("numero1 es negativo o distinto de cero");  
  }  
}  
  
// Prueba con distintos números, por ejemplo con 5 y 2  
miFuncionDeComparar(5, 2);  
  
// Esto deberías ver por pantalla:  
// "numero1 es estrictamente igual a 5"  
// "numero2 es positivo"
```

## Ejercicio 2 → Cambiando de rueda

Ahora que has completado una función, es hora de que hagas tú una de cero. ¡A por ello!

Debes crear una función que sepa clasificar ruedas de juguete, siempre y cuando le indiques el diámetro de la misma 

Pista: para indicarle el diámetro quizá debas pasárselo como parámetro a la función 😊

En base al diámetro que reciba la función se comportará así:

- 1) Si el diámetro es **inferior o igual** a 10, la función deberá imprimir por consola "es una rueda para un juguete pequeño"
- 2) Si el diámetro es **superior** a 10 y **menor** de 20, la función deberá imprimir por consola "es una rueda para un juguete mediano"
- 3) Si el diámetro es **superior o igual** a 20, la función deberá imprimir por consola "es una rueda para un juguete grande"

Ejecuta varias veces tu función con distintos valores, por ejemplo: 5, 7, 14, 31... y comprueba que los resultados son correctos. Recuerda que puedes realizar el ejercicio en un fichero *ejercicio2.js*.

## Ejercicio 3 → Aplicación práctica (DNI)

¡Vamos a trabajar un poco con arrays!

El cálculo de la letra del Documento Nacional de Identidad (DNI) es un proceso matemático sencillo que se basa en obtener el **resto** de la **división** entera del **número de DNI** y el número **23**. A partir del resto de la división, se obtiene la letra seleccionándola dentro del siguiente array de letras:

```
var letras = ['T', 'R', 'W', 'A', 'G', 'M', 'Y', 'F', 'P', 'D', 'X', 'B', 'N',
             'J', 'Z', 'S', 'Q', 'V', 'H', 'L', 'C', 'K', 'E', 'I'];
```

Por tanto si el resto de la división es **0**, la letra del DNI es la **T** y si el resto es **3** la letra es la **A**. Con estos datos, elaborar un pequeño ficheros JS que:

1. Solicite al usuario su número de DNI y su letra. Debes realizar las preguntas por separado, es decir que tendrá que introducir dos datos, no uno.  
Pista: para solicitar un dato al usuario puedes usar la función prompt

2. En primer lugar (y en una sola instrucción) se debe comprobar si el número es menor que 0 o mayor que 99999999. Si ese es el caso, se muestra un mensaje al usuario indicando que el número proporcionado no es válido y el programa no muestra más mensajes.
3. Si el número es válido (paso), debes comprobar si la letra indicada por el usuario es correcta, para ello sigue el método explicado anteriormente. Deberás indicar al usuario el resultado de la validación, imprimiendo por pantalla "DNI Válido" o "DNI Incorrecto".

Crea un fichero *ejercicio3.js* con las funciones necesarias para calcular y mostrar los mensajes indicados.

## Ejercicio 4 → La pirámide

¿Te gustan las pirámides? Esperamos que sí, porque vas a aprender a hacer una con JavaScript! Para trabajar en tu *ejercicio4.js*, crea una función que sea capaz de **imprimir por consola** todos los números comprendidos entre 1 9, y además encargándose de que se repita tantas veces como su numero indique, es decir que el 7 aparecerá 7 veces.

El resultado será algo parecido a esto:

```
1
22
333
4444
55555
666666
7777777
88888888
999999999
```

Debemos aprovechar la potencia de los bucles para no rellenar nuestro fichero de código repetido.

## Ejercicio 5 → Toma nota

Realiza un *fichero5.js* en el que se declare una variable inicial con el texto:

```
var texto = "marvel mola!";
```

Y ahora recorre la cadena de texto para imprimir por consola "m-a-r-v-e-l -m-o-l-a-!". Aprovechate de los bucles y las funciones **length** - **charAt()**.

## Ejercicio 6 → Jugando a Piedra, Papel o Tijera

Vamos a comprobar que la programación se encuentra en nuestro día a día, en los juegos más simples, desde comparaciones a iteraciones en bucles, por eso os proponemos que realicéis vuestro primer juego de JS - Piedra, Papel o Tijera 🎮

No os vamos a dejar sol@s, por esta vez os vamos a dar el esqueleto del código para que lo completéis. En las zonas con la palabra **CODE** debéis añadir vuestro granito de arena y recuerda

| Piedra gana Tijera / Tijera gana Papel / Papel gana Piedra

```
var jugados = 0;
var ganados = 0;
var perdidos = 0;

while ("Jugar de nuevo") {
  var jugada = prompt("Escribe \"piedra\", \"papel\" o \"tijera\"");

  // GUARDAMOS UN NÚMERO ALEATORIO DE 1 AL 9
  var aleatorio = Math.floor(Math.random() * 9) + 1);
  var elige = "";
  if (aleatorio <= 3) {
    elige = "piedra";
  }
}
```

```

else if (aleatorio <= 6) {
    elige = "papel";
}
else {
    elige = "tijera";
}

// De aqui en adelante, rellena las plantillas de CODE con lo adecuado ;)
if ((jugada == CODE ) && (elige == CODE)) {
    alert(jugada+"-"+elige+": Empate");
}
else if ((jugada == "piedra") && (elige == "papel")) {
    alert(jugada+"-"+elige+": Gana PC");
    perdidos++;
}
else if ((jugada == CODE) && (elige == CODE)) {
    alert(jugada+"-"+elige+": Gana Jugador");
    ganados++;
}
else if ((jugada == "papel") && (elige == CODE)) {
    alert(jugada+"-"+elige+": Empate");
}
else if ((jugada == "papel") && (elige == CODE)) {
    alert(jugada+"-"+elige+": Gana PC");
    perdidos++;
}
else if ((jugada == CODE) && (elige == "piedra")) {
    alert(jugada+"-"+elige+": Gana Jugador");
    ganados++;
}
else if ((jugada == "tijera") && (elige == CODE)) {
    alert(jugada+"-"+elige+": Empate");
}
else if ((jugada == CODE) && (elige == "piedra")) {
    alert(jugada+"-"+elige+": Gana PC");
    perdidos++;
}
else if ((jugada == CODE) && (elige == CODE)) {
    alert(jugada+"-"+elige+": Gana Jugador");
    ganados++;
}
jugados++;

};

document.write("Partidas jugadas: "+jugados+"<br>");
document.write("Partidas ganadas: "+ganados+"<br>");
document.write("Partidas perdidas: "+perdidos+"<br>");
document.write("Partidas empatadas: "+(jugados-ganados-perdidos));

```

Mucha suerte! Y que gane el mejor! 🤪

## Una vez termines con los ejercicios de JS...

Vamos a practicar con el último ejercicio que te proponemos. Un proyecto completo con HTML, CSS y JS en el que combinarás todo lo que sabes 🚀

 [HTML CSS JS | Proyecto](#)