

prompt()

Uno de ellos es el `prompt()`, este muestra un cuadro de diálogo con mensaje opcional, que solicita al usuario que introduzca un texto. Además tiene dos opciones de “Aceptar” o “Cancelar”.

```
prompt("Por favor introduzca su nombre");
```

En la consola del navegador podremos ver el resultado que nos devuelve este método, **lo que sea que introduzca el usuario retornado en una cadena de texto.**

🚨 ¿Esto qué quiere decir? Que por más que el usuario introduzca solo números, lo que obtendremos serán esos números pero como si fueran texto.

Por el momento ese dato se pierde, pero si lo almacenamos en un variable podremos utilizarlo para el resto de nuestro programa. Hagamos esa prueba.

```
let nombreUsuario = prompt("Por favor, introduzca su nombre");  
console.log(nombreUsuario);
```

Al utilizar un espacio en memoria para guardar este dato, podremos utilizarlo como información.

En el caso de que el usuario seleccione “Cancelar”, el resultado que nos retorna es **null**. En cambio si el usuario presiona “Aceptar” sin completar nada, nos devolverá un **texto vacío**. Probemos también este caso y veamos el resultado.

confirm()

El método **confirm()** muestra un cuadro de diálogo con un mensaje opcional y dos botones, “Aceptar” y “Cancelar”. En este caso, lo que nos permite es ingresar alguna pregunta o indicación al usuario para que este responda por sí o no únicamente.

El valor que nos va a retornar es un booleano indicando **true si pulsamos Aceptar y false si elegimos Cancelar**.

```
confirm("¿Desea un trato personalizado?")
```

Recordemos que por sí solo el método no almacena ningún dato, pero si almacenamos el valor retornado en una variable, lo podremos utilizar como información.

👨‍💻 Hagamos la prueba de combinar ambos métodos en algo más funcional.

```
let tratoPersonalizado = confirm("¿Desea un trato personalizado?")
```

```
if(tratoPersonalizado){  
  let nombre = prompt("Por favor, introduzca su nombre");  
  alert("Le damos la bienvenida a nuestro sitio "+ nombre +". ¡Muchas  
gracias por visitarnos, estamos a su disposición! 😊");  
} else {  
  alert("Gracias por conectarse.")  
}
```

Ahora si podemos ir viendo cómo tiene más sentido implementar los métodos, pero de manera conjunta para lograr un resultado más complejo. O por lo menos guardar esos datos y utilizarlos cuando lo creamos conveniente.