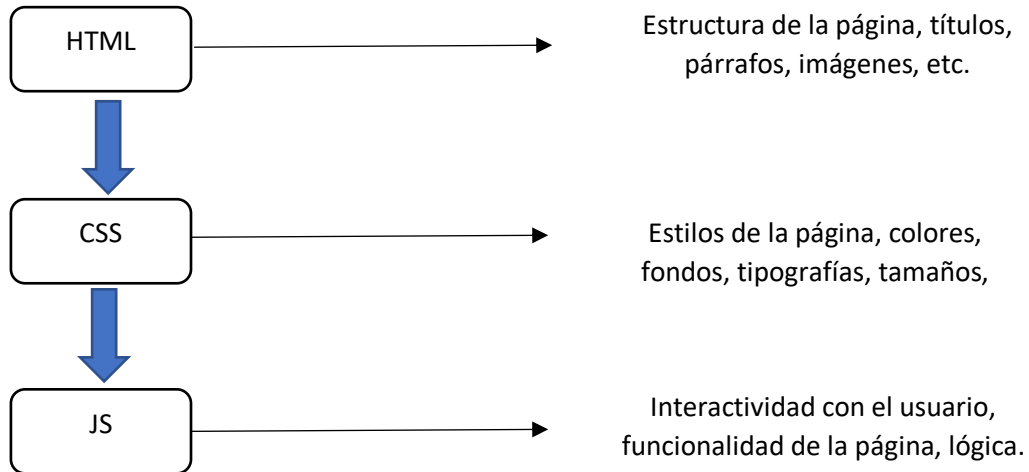
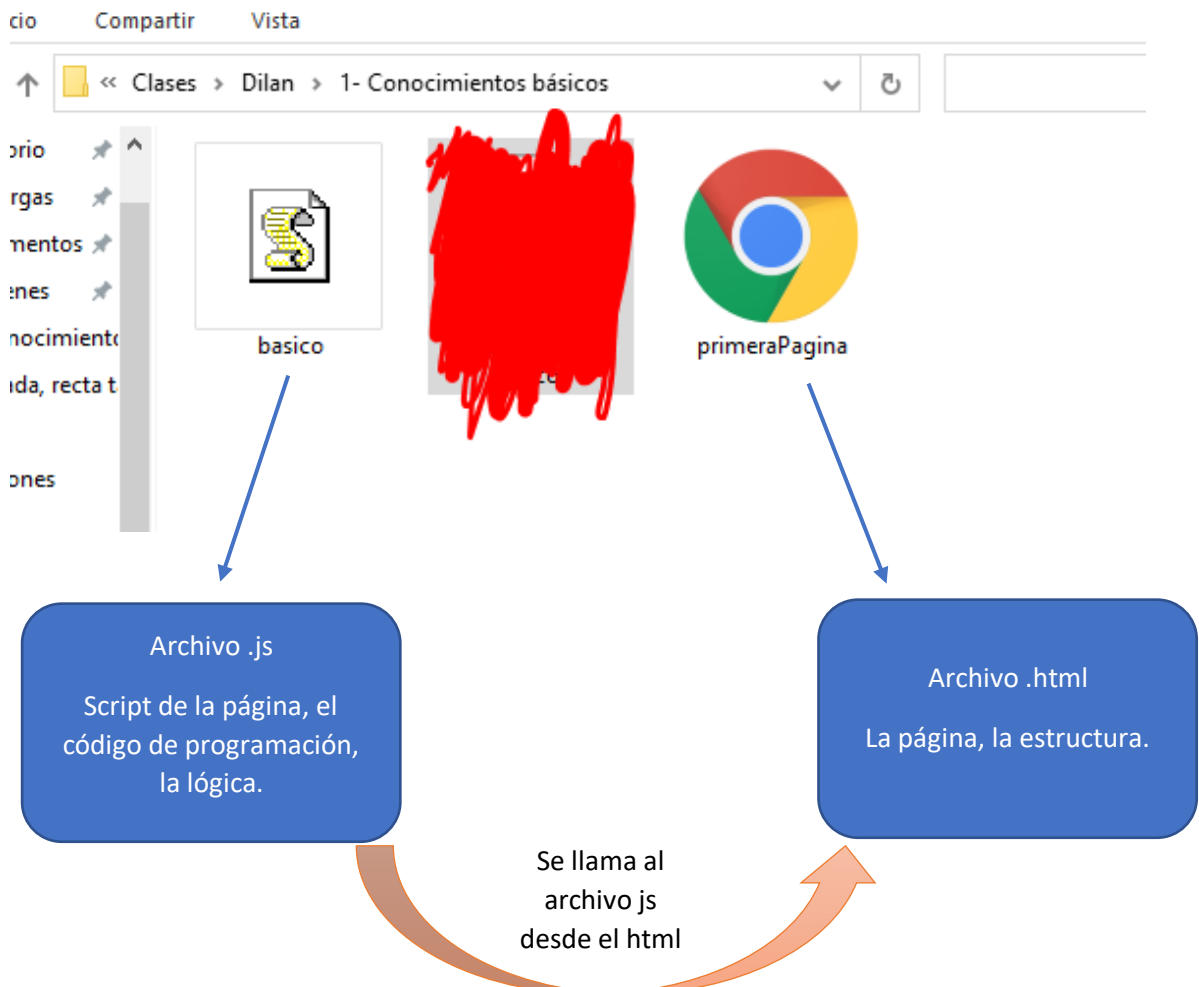


# Conocimientos básicos

## Estructura de página web



Vamos a manejar dos archivos



# HTML

Establece la versión del HTML a la versión 5

Cabecera: Información que no se muestra al usuario

Título que aparece en la pestaña del navegador

Para poder escribir letras con tildes y letra ñ

Inserta el código JavaScript

Título visible en página

párrafo

```
1  <!DOCTYPE html>
2
3  <html>
4    <head>
5      <title>primera página</title>
6      <meta charset="utf-8">
7
8      <script src="basico.js"></script>
9    </head>
10
11    <body>
12      <h1>primera página</h1>
13      <p>esto es un párrafo</p>
14    </body>
15  </html>
16
17
```

HTML

HEAD -> cosas no visibles

BODY -> cosas visibles

# Programación en JavaScript

## Mostrar por pantalla

```
alert( contenido );
```

Función que muestra la ventana emergente

La ventana muestra lo que le pongamos de contenido **dentro de los paréntesis**

Siempre poner un ; al final

Textos SIEMPRE ENTRE COMILLAS

```
alert( "un texto" );
```

Números

```
alert( 10 );
```

El contenido puede ser cualquier cosa

Los dos (separados con un +)

```
alert( "El número es " + 10 );
```

El número es 10

Lo que hace el signo + es juntar los dos términos

```
alert("Esto es una ventana emergente");
```

ocimientos%20básicos/primerPagina.html

Esta página dice  
Esto es una ventana emergente

Aceptar

## Imprimir en consola

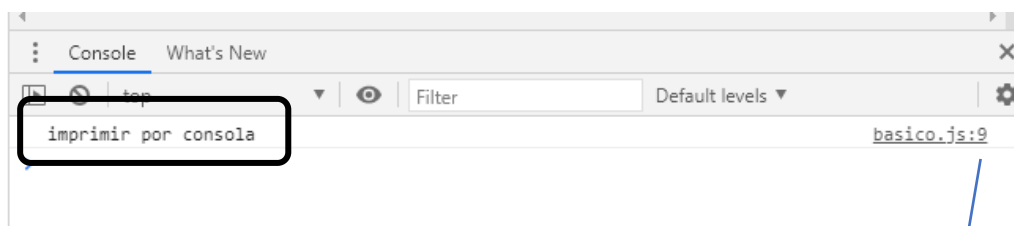
Imprimir significa escribir. Se trata de escribir mensajes por la consola del navegador.

Los mensajes no se muestran al usuario, porque son mensajes que no les importa. Estos mensajes son importantes solo para programadores.

Usar F12 para hacer aparecer la consola.

Funciona exactamente igual que el `alert()`

```
console.log( "imprimir por consola" );
```

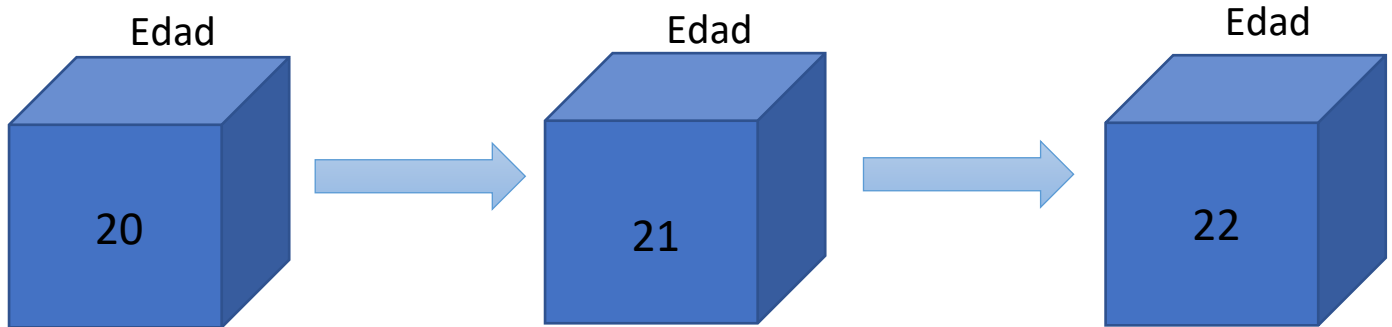


Incluso te dice el archivo js que hacer la impresión y te dice en qué línea del código se manda a escribir (en este caso la línea 9)

## Variables

Son como cajas que guardan datos (nombres, edades, medidas, etc).

Se le asigna un nombre, un valor y un tipo de dato.



El valor de la variable Edad puede cambiar a lo largo del código. Es decir, tenemos la posibilidad de cambiar el valor

## Tipos de datos

El tipo del dato de la variable es asignado automáticamente según el valor que tenga. Es decir, el lenguaje ante de guardar la variable, detecta de qué tipo es y define la variable con ese tipo y luego guarda el valor.

Siempre que escribamos `let` antes del nombre de la variable, el lenguaje interpreta que **se está creando** una nueva variable

Y establece que edad guarda datos de tipo número  
Y guarda el valor

```
let edad = 20;
```

El lenguaje detecta al 20 como un número  
(Si tuviera comillas, la detecta como texto)

```

//números / int
let edad = 20; //define el tipo de dato como número -> int: entero

//string
let nombre = "pablo";

//float -> números con coma decimal
let precio = 5.99;

//booleano -> true y false
let foo1 = true;
let foo2 = false;

alert( "el tipo de la variable nombre es " + typeof(nombre) + " y el valor es: " + nombre);

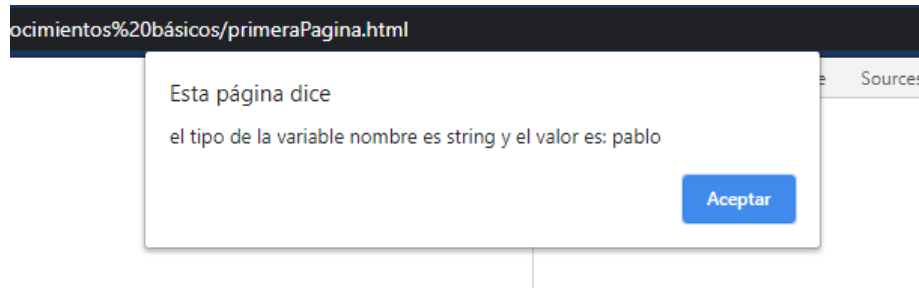
```

Lo que hace el lenguaje

typeof(nombre) deja  
como resultado el tipo  
de la variable que le  
demos dentro de los  
paréntesis

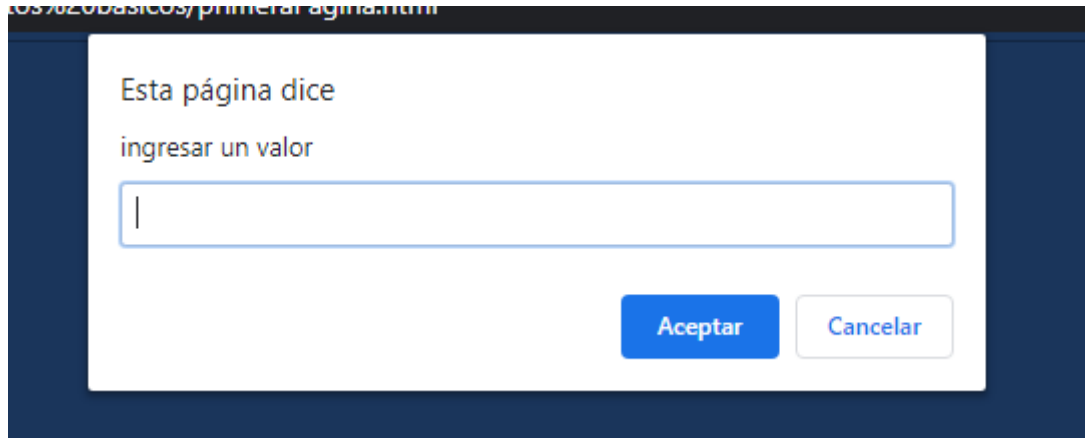
Alert( "el tipo de la variable nombre es " + "string" + " y el valor es: " + "Pablo");

Lo que muestra el código



## Entrada de datos con prompt()

La función `prompt()` pausa la página y muestra una ventana emergente, donde espera a que el usuario ingrese algo por teclado para guardarlo en una variable.



El prompt tiene dos botones: aceptar y cancelar. Si se presiona en aceptar, se envía lo que se haya escrito por teclado. Si se presiona cancelar, Se envía un valor llamado *null*. Este valor significa vacío.

Básicamente, una variable puede tener tres tipos de valores

Números, textos, etc.

undefined: que significa que es una variable creada, pero que no tiene valor ni tipo definido. Solo se creó, pero queremos que tenga un valor

null: que significa que es una variable vacía y que queremos que esté vacía. (intencionalmente vacía)

Non-zero value

0



null



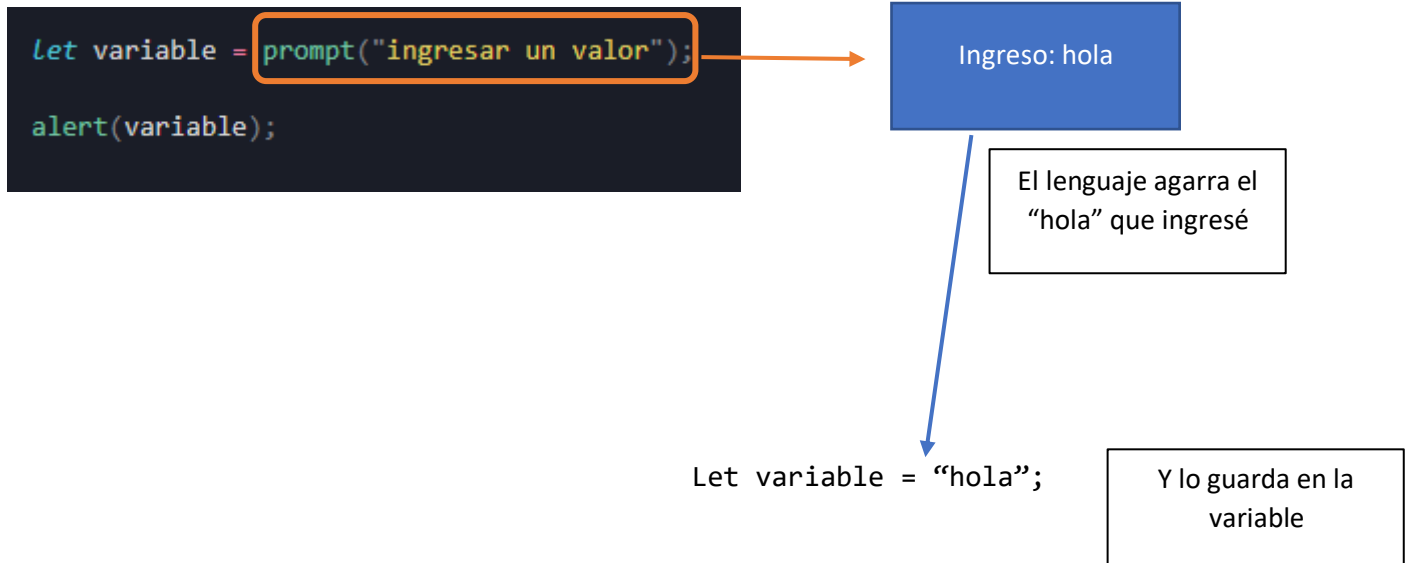
undefined

Imagen que sirve para entender la diferencia entre null (intencionalmente vacío)

Y undefined (no existe el valor)



## Código de prompt



## Conversión de variables

Por defecto, la función `prompt` guarda todos los datos como si fueran un texto. Para guardar números hay que convertirlos a lo que son, números.

Es decir





Para convertir un texto a entero usamos la función `parseInt()` El cual va a convertir en entero la variable que le demos como argumento dentro de los paréntesis.

Si se quiere convertir un texto a float (con coma flotante) se debe de usar la función `parseFloat()`.

Supongamos que quiero ingresar el número 5

```
let numero1 = parseInt( prompt("ingresar número 1") );
```

```
let numero 1 = parseInt(      “5”      );
```

```
let numero 1 = 5;
```

Lo que hace el lenguaje es agarrar lo que se ingresó en el prompt (en este caso 5) y por defecto deja el resultado como un texto (sería “5”). Luego lo convierte a entero y lo guarda en la variable como si fuera un número entero