

- 
- **HTML básico**
  - **Introducción al DOM**

# Repaso de la estructura de un documento HTML

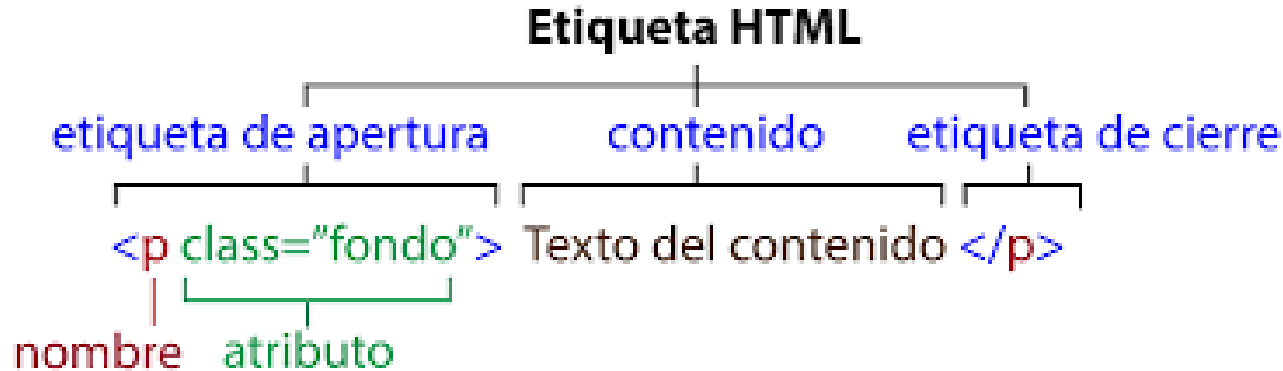
`index.html > ...`

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Mi página web</title>
6  </head>
7  <body>
8
9  </body>
10 </html>
```

# Etiquetas

- Las etiquetas se definen con los símbolos < y >
- Algunas etiquetas deben cerrarse, con </>
- Ejemplo de etiqueta: `<body></body>`
- Ejemplo de etiqueta que no se cierra: `<br>`
- 
-

# Atributos de las etiquetas



- Los atributos nos ayudan a configurar, o identificar los elementos (etiqueta) HTML.
- Son opcionales, no es necesario definir los atributos de una etiqueta.

# Atributo Class

- `<p class="parrafo_principal">Hola mundo</p>`
- Las clases (class) nos permiten seleccionar elementos (desde Javascript y CSS) a un elemento, o a un conjunto de elementos.
- Esto nos sirve para por ejemplo, aplicar el mismo diseño (desde css) a todos los elementos con dicha clase, o desde javascript obtener elementos o sus valores, que contengan dicha clase

# Clases múltiples

- Un elemento HTML puede tener tantas clases como queramos. Debemos colocar el nombre de cada clase separada por un espacio (dentro del atributo class)
- Ejemplo:

```
</head>
<style>
    .principal {
        color: red;
        text-transform: uppercase;
    }

    .titulo { font-weight: bold; }
</style>
<body>
    <p class="principal titulo">Hola xdx dx</p>
</body>
</html>
```

# Atributo id

- El atributo id es otro identificador (al igual que class), pero este debe ser único para cada elemento de nuestro documento html.

```
</head>
<style>
  #principal {
    color: rgb(17, 167, 37);
  }
</style>

<body>
  <p id="principal">Lorem ipsum</p>

  <script>
    let elemento = document.getElementById('principal');
  </script>
</body>
</html>
```

# Atributos específicos por tipo de etiqueta

- Hay algunos atributos que son específicos de cada etiqueta

- Ejemplo:

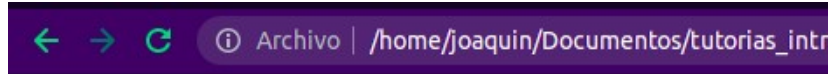
```
<!--  
<style>  
  #titulo {  
    color: #1A38D2;  
    font-size: 25px;  
  }  
</style>  
  
<body>  
  <p id="titulo">Mi pato</p>  
    
</body>  
</html>
```



# Etiqueta img

- Nos permite insertar imágenes en nuestra página.
- En el atributo src (source) debemos indicarle la ruta
- Podemos indicarle una ruta local, o una URL de internet.
- Con el atributo height indicamos la altura de la imagen (en píxeles), y con el atributo width indicamos el ancho.

# Vista del código anterior:



Mi pato



# Etiqueta <a>

- Nos permite crear un enlace a otras páginas de internet, archivos o ubicaciones dentro de la misma página, direcciones de correo, o cualquier otra URL.
- `<a href="www.google.com">Google<a>`

# Heading tags

- Con estas etiquetas podemos definir títulos y subtítulos de nuestro documento html
- `<h1>Titulo principal</h1>`
- `<h2>Sub Titulo</h2>`
- `<h3>otro Sub Titulo</h3>`
- `<h4>otro Sub Titulo</h4>`
- `<h5>otro Sub Titulo</h5>`
- `<h6>otro Sub Titulo</h6>`
- Cada etiqueta se comporta de una manera por defecto, según el navegador

# Otras etiquetas básicas

- `<b>Negrita</b>`   `<strong>Negrita</strong>`
- `<i>Cursiva</i>`   `<em>Cursiva</em>`
- `<u>Subrayado</u>`
- Con `ul` podemos definir una lista:

`<ul>`

`<li>Elemento 1</li>`

`<li>Elemento 2</li>`

`</ul>`

Otra variante, es la etiqueta `ol`, que nos permite crear una lista ordenada.

# Mas info de las etiquetas:

- Para ver todos los atributos, o investigar otras etiquetas, es recomendable leer la documentación de HTML de Mozilla, o la de Devdocs, o investigar en otro sitios.
- Devdocs:  
<https://devdocs.io/html/>
- Mozilla:  
<https://developer.mozilla.org/es/docs/Web/HTML>

# Formularios en HTML

- Los formularios nos ayudan a recopilar datos entrada del usuario. Luego, podemos enviar dicho formulario (por ejemplo) a un servidor.
- Ejemplo de formulario:

Nombre:

Correo:



# Elementos básicos de un formulario

- Inputs: nos permiten ingresar texto
- Labels: es una etiqueta de texto, asociada a un elemento de la interfaz (como por ejemplo un input o un botón)
- Select: define una lista desplegable con opciones
- Textarea: define un campo de texto que puede ser aumentado o reducido de tamaño.
- Button: define un botón al cual se le puede dar click y generar una acción (la acción será programada desde js)



# Ejemplos de elementos de un form:

Esto es un label

Lo de abajo es un textarea:



Para investigar mas acerca de los formularios:  
[https://www.w3schools.com/html/html\\_forms.asp](https://www.w3schools.com/html/html_forms.asp)

# Código HTML del formulario:

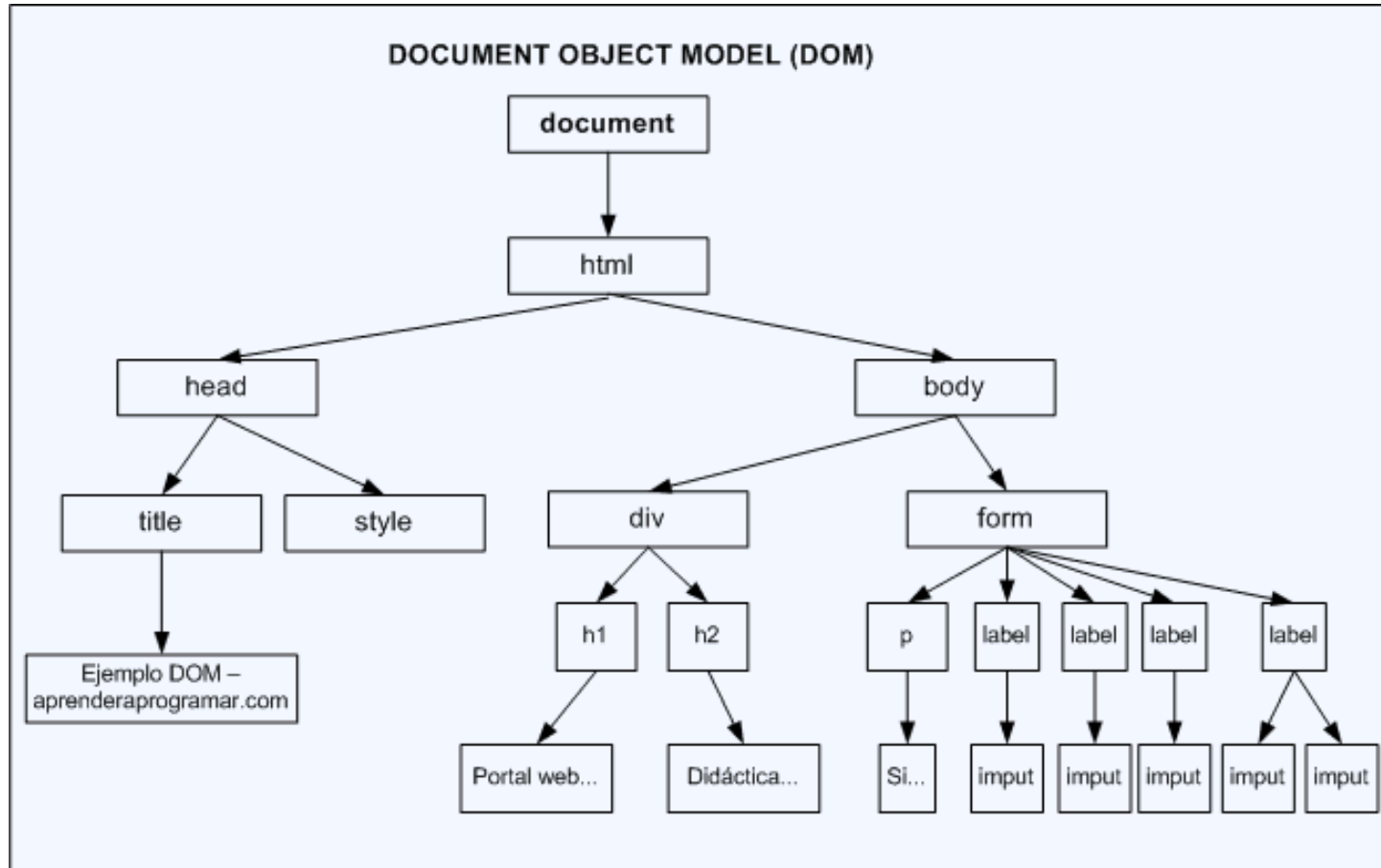
```
<form>
  <label for="entrada">Esto es un label</label>
  <input id="entrada" type="text" value="Esto es un input"> <br>

  <p>Lo de abajo es un textarea:</p>
  <textarea value="esto es un textarea" cols="20" rows="10"></textarea> <br>

  <button>Dame click!</button>

  <select id="opciones">
    <option value="1" selected>Opcion 1</option>
    <option value="2">Opcion 2</option>
    <option value="3">Opcion 3</option>
  </select>
</form>
```

# Document Object Model





# DOM

- El DOM es la estructura de objetos que genera el navegador cuando se carga un documento y se puede alterar mediante Javascript para cambiar dinámicamente los contenidos y aspecto de la página.
- Nos da una representación del documento como un grupo de nodos y objetos estructurados que tienen propiedades y métodos.
- El DOM es un “árbol” (en computación, los arboles son un tipo abstracto de datos (TAD))



# Objeto document

- El objeto document, es una instancia de la interfaz Document, la cual hace referencia al documento HTML, desde javascript.
- En pocas palabras, “document” es el body de nuestra página, en javascript.
- Con este objeto podemos alterar la página (crear, modificar o eliminar elementos), obtener elementos, valores de los elementos, etc.

# Obtener un elemento HTML desde Javascript

- Para seleccionar un elemento, usaremos el objeto document
- Podemos seleccionar un elemento según:
  - Su id
  - Sus clases
  - Su propiedad name
  - Su tag name (nombre de la etiqueta)

# Seleccionar elemento por id

- Usaremos el método `getElementById(id)`
- Debido a que el id es único por elemento, este método nos retorna un elemento solo, o null en caso de no existir o no encontrarlo.
- Se le debe indicar el id como parámetro en forma de string

Nombre:

Apellido:

```
> document.getElementById('apellido')  
< <input class="campos" id="apellido" type="text" value="Perez">  
>
```

# Seleccionar por name

- `document.getElementsByName(name);`
- Este método nos retorna una lista (o array) de todos los elementos que tengan dicho valor en la propiedad name
- Recibe como parámetro un string del valor name.

Parrafo 1	> document.getElementsByName("parrafo")
	<> ▶ NodeList(4) [p, p, p, p]
Parrafo 2	> document.getElementsByName("parrafo")[0]
Parrafo 3	<> <p name="parrafo">Parrafo 1</p>
Parrafo 4	>



# Seleccionar por nombre del tag

- Podemos seleccionar todos los elementos del mismo tag, con `document.getElementsByTagName(tag)`

- 

Parrafo 1	> document.getElementsByTagName("p")
Parrafo 2	< ▶ HTMLCollection(4) [p, p, p, p, parrafo: p]
Parrafo 3	>
Parrafo 4	

# Seleccionar por clase

- Usamos el método `document.getElementsByClassName(class)`

Parrafo 1	> document.getElementsByClassName("parrafo")
Parrafo 2	< ▶ HTMLCollection(4) [p.parrafo, p.parrafo, p.parrafo, p.parrafo]
Parrafo 3	>
Parrafo 4	



# Próxima clase:

- Seguiremos con el manejo del DOM
- Aprenderemos a crear, modificar y eliminar elementos del documento.
- Veremos conceptos básicos de CSS aplicados al manejo de elementos con el objeto document
- Finalizaremos el tema DOM.