Clase 11

BA MULTIPLICA 2.0
jóvenes x jóvenes

Métodos y Eventos

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

BA Joven

Contenidos de la Clase

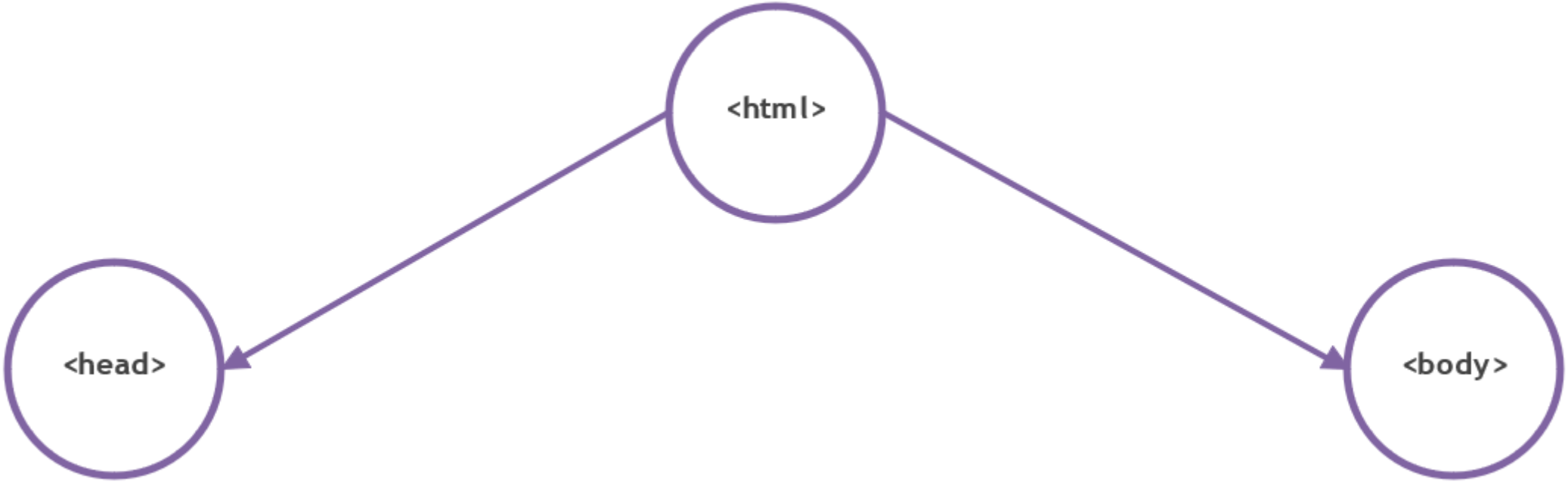
- DOM
- Eventos
- Métodos
- Práctica e Integración

¿Qué es el DOM?

El DOM (Document Object Model, Modelo de Objetos del Documento) 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.

```html
<!DOCTYPE html>
<html>
    <head>
        <title>Document</title>
    </head>
    <body>
        <h1>Mi título principal</h1>
        <a href="https://google.com">Ir a Google</a>
    </body>
</html>
```

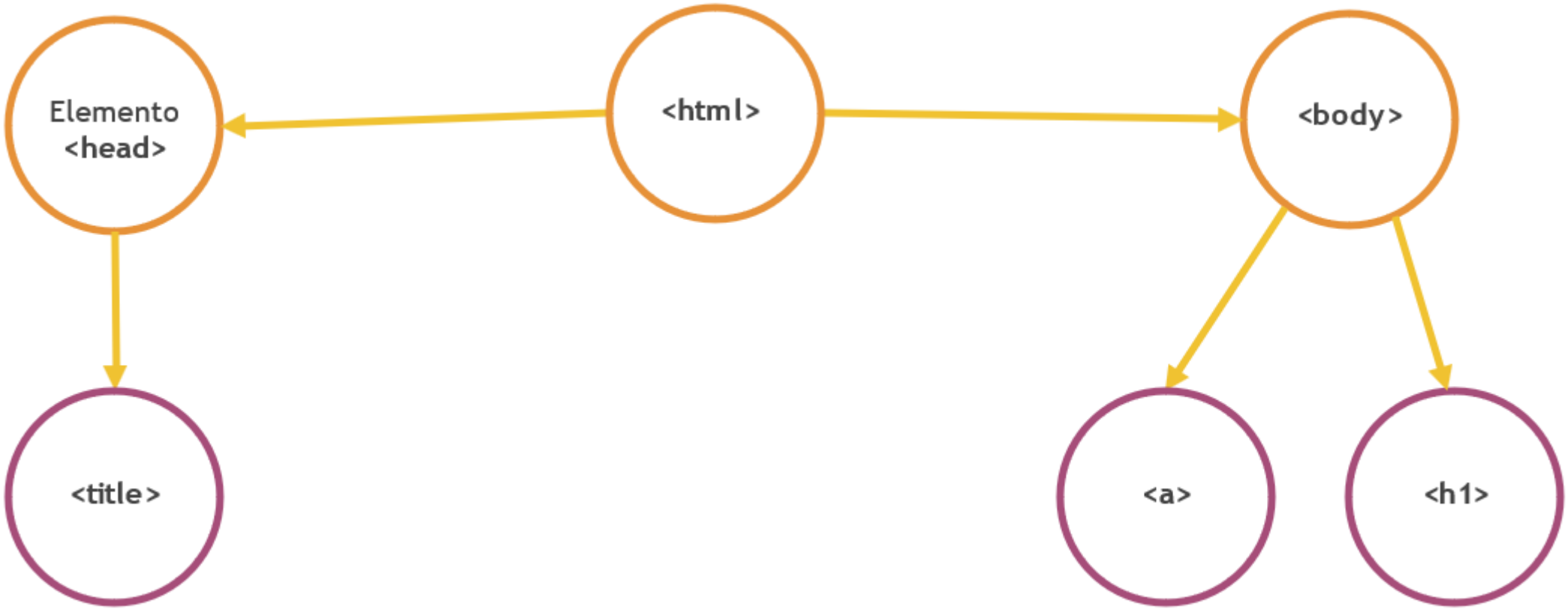
Nuestro DOM



Nuestro HTML

```html
<!DOCTYPE html>
<html>
    <head>
        <title>Document</title>
    </head>
    <body>
        <h1>Mi título principal</h1>
        <a href="https://google.com">Ir a Google</a>
    </body>
</html>
```

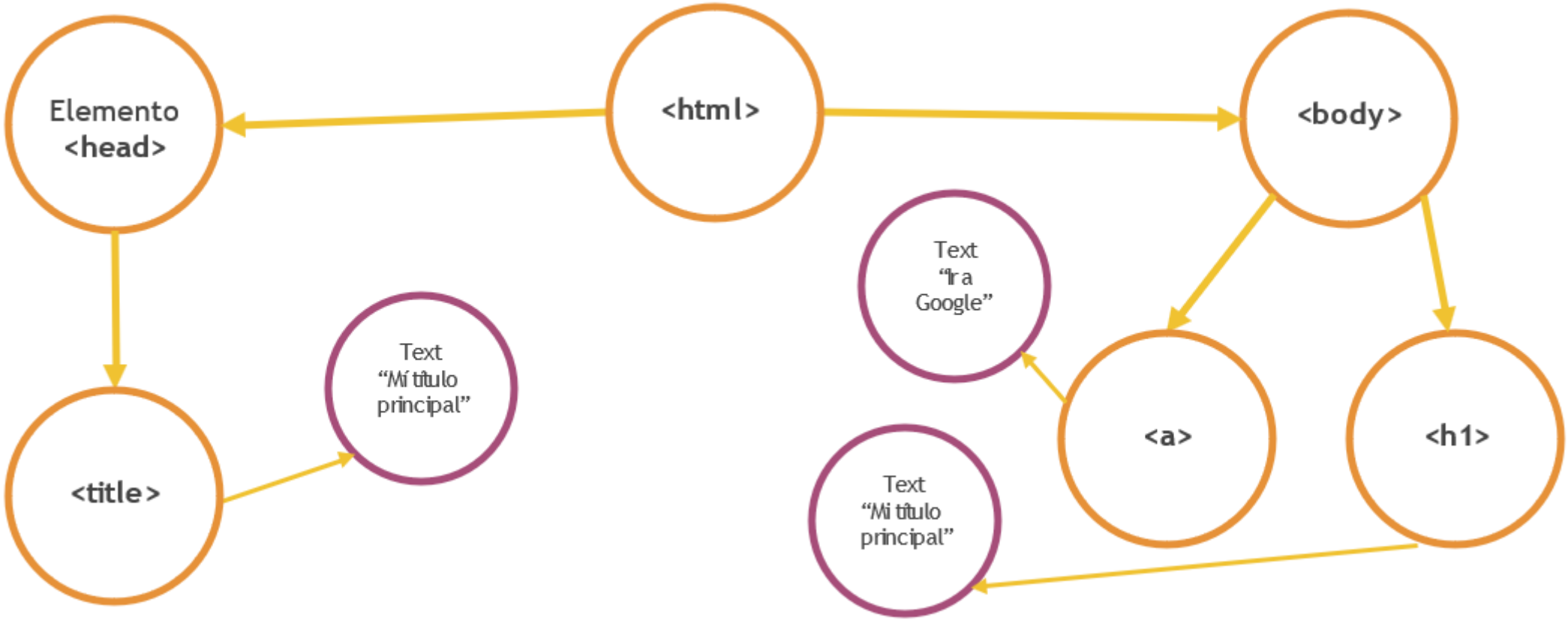
Nuestro DOM



Nuestro HTML

```
<!DOCTYPE html>
<html>
    <head>
        {<title>Document</title> }
    </head>
    <body>
        {<h1>Mi título principal</h1>}
        {<a href="https://google.com">Ir a Google</a>}
    </body>
</html>
```

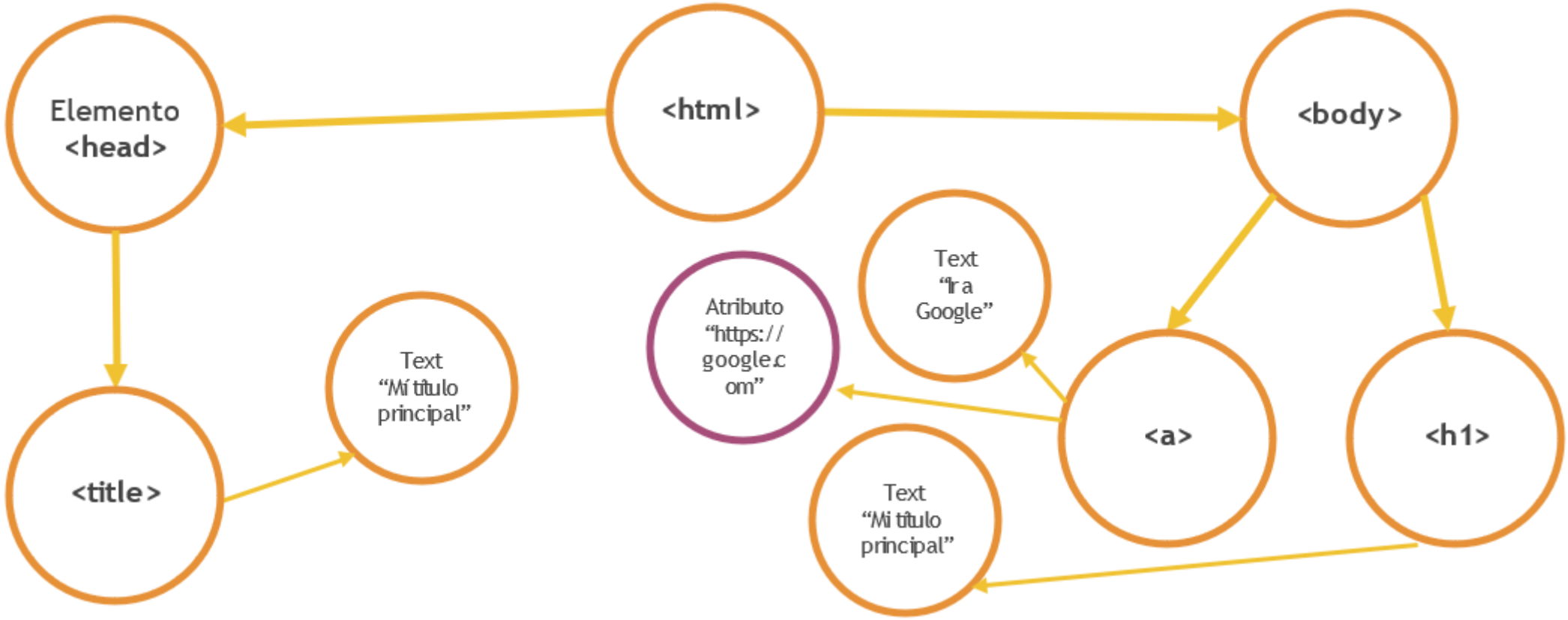
Nuestro DOM


Elemento
<head>
<html>
<body>
Text
"Mi título
principal"
Text
"Ir a
Google"
<title>
<a>
<h1>
Text
"Mi título
principal"

Nuestro HTML

```html
<!DOCTYPE html>
<html>
    <head>
        <title>Document</title>
    </head>
    <body>
        <h1>Mi título principal</h1>
        <a href="https://google.com">Ir a Google</a>
    </body>
</html>
```

Nuestro DOM


Elemento
<head>
<html>
<body>
Text
"Mi título
principal"
Atributo
"https://
google.c
om"
Text
"Ir a
Google"
<title>
<a>
<h1>
Text
"Mi título
principal"

En JS todos los elementos que manipulamos son objetos. Éstos tienen propiedades y métodos.

Veamos un ejemplo...

Métodos y Propiedades

Objeto: Gato

Propiedades: Color, pelo

Métodos: ronronear(), rasguñar()

¿El DOM es un Objeto?

La forma de acceder al DOM es a través de un objeto llamado document, que representa el árbol DOM de la página o pestaña del navegador donde nos encontramos.

Esto quiere decir, que...

¡El DOM tiene métodos y propiedades!

Veamos algunos ejemplos de esto...

Métodos

Los métodos sirven para hacer **acciones**, modificar o encontrar elementos.

document.querySelector()

Nos permite encontrar elementos dentro del documento.

- Para llamar por nombre de etiqueta: **document.querySelector(“h1”)**
- Para llamar por nombre de ID: **document.querySelector(“#titulo”)**
- Para llamar por nombre de clase: **document.querySelector(“.miClase”)**

Propiedades

Las propiedades nos permiten interactuar con el elemento.

- Para acceder al texto dentro de la etiqueta: **element.innerText**
- Para acceder al estilo dentro de la etiqueta : **document.style**

IMPORTANTE

Estas propiedades pertenecen a los elementos dentro del DOM,
y no directamente al objeto document.

Eventos

Cuando un usuario visita una página web e interactúa con ella se producen los eventos y con Javascript podemos definir qué queremos que ocurra cuando se produzcan.

Por ejemplo: podemos definir qué pasa cuando un usuario hace click en un botón

onClick

```html
<button onclick="alert('¡Hola!')">Saludar</button>
```

Si lo que tiene que hacer nuestro código es más complejo podríamos reemplazar el valor del onClick por una función.

```html
<button onclick="saludar()">Saludar</button>
```

Extra - Métodos

```
// getElementById(): se utiliza para obtener un elemento del DOM (Modelo de Objeto de Documento) basado en su ID.
let elemento = document.getElementById("mi-id");

// getElementsByTagName(): se utiliza para obtener una colección de elementos del DOM basados en su etiqueta HTML.
let elementos = document.getElementsByTagName("p");

// element.setAttribute(), element.getAttribute(): Estos métodos nos permiten establecer o recuperar el valor de un atributo HTML de un elemento. Por ejemplo, si queremos establecer el valor del atributo "src" de una imagen con el ID "mi-imagen", podemos hacerlo de la siguiente manera:
let miImagen = document.getElementById("mi-imagen");
miImagen.setAttribute("src","ruta/de/la/imagen.jpg");
```

Extra - Métodos

```
// element.innerHTML: Este método nos permite establecer o recuperar el contenido HTML de un elemento. Por ejemplo, si queremos establecer el contenido de un elemento con el ID "mi-elemento" como un párrafo con el texto "Hola mundo", podemos hacerlo de la siguiente manera:
let miElemento = document.getElementById("mi-elemento");
miElemento.innerHTML = "<p>Hola</p>";

// element.addEventListener(): Este método nos permite añadir un evento a un elemento HTML para que se ejecute una función cuando se produce dicho evento. Por ejemplo, si queremos que se ejecute una función llamada "miFuncion" cuando se hace clic en un botón con el ID "mi-boton", podemos hacerlo de la siguiente manera:
let miBoton = document.getElementById("mi-boton");
miBoton.addEventListener("click", miFuncion);
```

Extra - Métodos

```
// window.scroll(), element.scrollIntoView(): Estos métodos nos permiten controlar el desplazamiento de la página y la posición del viewport mediante JavaScript. Por ejemplo, si queremos hacer que la página se desplace automáticamente hasta un elemento con el ID "mi-elemento", podemos hacerlo de la siguiente manera:
let miElemento = document.getElementById("mi-elemento");
miElemento.scrollIntoView();

// querySelectorAll(): La siguiente variable contiene una colección de todos los elementos que tienen la clase CSS “mi-clase”.
let elementos = document.querySelectorAll("mi-clase");

// createElement(): debemos llamarlo con la etiqueta HTML que deseamos crear. En el siguiente ejemplo, la variable "nuevoElemento" contiene un nuevo elemento "div".
let nuevoElemento = document.createElement("div");
```

Extra – Métodos

```
// element.classList.add(), element.classList.remove(), element.classList.toggle(): Estos métodos nos permiten añadir, eliminar o alternar una clase CSS de un elemento HTML. Por ejemplo, si queremos añadir la clase "activo" a un elemento con el ID "mi-elemento", podemos hacerlo de la siguiente manera:
let miElemento = document.getElementById("mi-elemento");
miElemento.classList.add("activo");

// element.style: Este método nos permite establecer o recuperar las propiedades CSS de un elemento. Por ejemplo, si queremos establecer el color de fondo de un elemento con el ID "mi-elemento" como rojo, podemos hacerlo de la siguiente manera:
let miElemento2 = document.getElementById("mi-elemento2");
miElemento2.style.backgroundColor = "red";

// getElementByClassName(): debemos llamarlo con el nombre de la clase CSS que deseamos seleccionar. La siguiente variable contiene una colección de todos los elementos que tienen la clase.
let elementos = document.getElementsByClassName("mi-clase");

// querySelector(): debemos llamarlo con un selector CSS. La siguiente clase contiene el primer elemento que coincide con el selector CSS "#mi-id .mi-clase".
let elemento = document.querySelector("#mi-id.mi-clase");
```

Extra – Eventos

En estas diapositivas se mostrarán elementos que requieren el concepto de función, el cual se verá en la próxima clase.

```
// click: Este evento se activa cuando se hace clic en un elemento.
let boton = document.getElementById("mi-boton");
boton.addEventListener("click", function(){
    //? Código a ejecutar cuando se hace click en el botón
})

// change: Este evento se activa cuando se cambia el valor de un elemento, como un campo de texto o un menú desplegable. Por ejemplo, podemos utilizarlo para actualizar un resultado cuando se cambia una selección en un menú desplegable.
let menu = document.getElementById("mi-menu");
menu.addEventListener("change",function(){
    //? Código a ejecutar cuando se cambia la selección del menú
})
```

Extra - Eventos

```
// submit: Este evento se activa cuando se envía un formulario. Por ejemplo, podemos utilizarlo para validar los datos de un formulario antes de enviarlo. Para aplicarlo, podemos utilizar el siguiente código:
let formulario = document.getElementById("mi-formulario");
formulario.addEventListener("submit", function(event){
    //? Código a ejecutar cuando se envía el formulario
    event.preventDefault(); //*Evita el envío del formulario si hay errores
})

// keydown: Este evento se activa cuando se presiona una tecla.
let campo = document.getElementById("mi-campo");
campo.addEventListener("keydown", function(event){
    //? Código a ejecutar cuando se presiona una tecla en el campo
})
```

Extra - Eventos

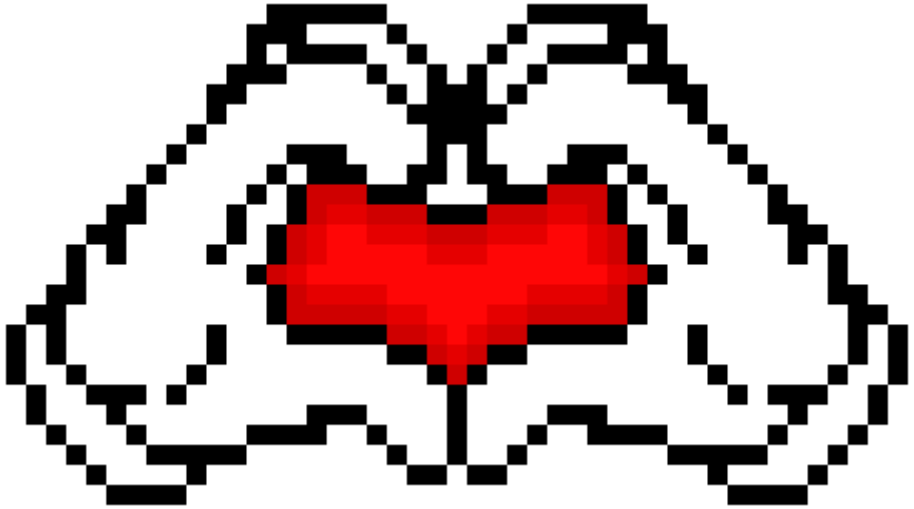
```
// mouseover: Se activa cuando el cursor entra en un elemento. Podemos utilizarlo para mostrar información cuando el usuario mueve el cursor sobre un enlace.
let enlace = document.getElementById("mi-enlace");
enlace.addEventListener("mouseover",function(){
    //? Código a ejecutar cuando el usuario mueve el cursor sobre el enlace
})
//! Para mouseout (el cursor sale del elemento) la sintaxis es igual

// load: Este evento se activa cuando se carga completamente una página o un elemento, como una imagen.
window.addEventListener("load",function(){
    //? Código a ejecutar cuando se carga completamente la página
})

// scroll: Este evento se activa cuando el usuario desplaza el contenido de la página.
window.addEventListener("scroll",function(){
    //? Código a ejecutar cuando el usuario desplaza el contenido de la página
})
```

¿Preguntas?

¡Nos vemos la próxima clase!



BA MULTIPLICA 2.0

jóvenes x jóvenes

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

BA Joven