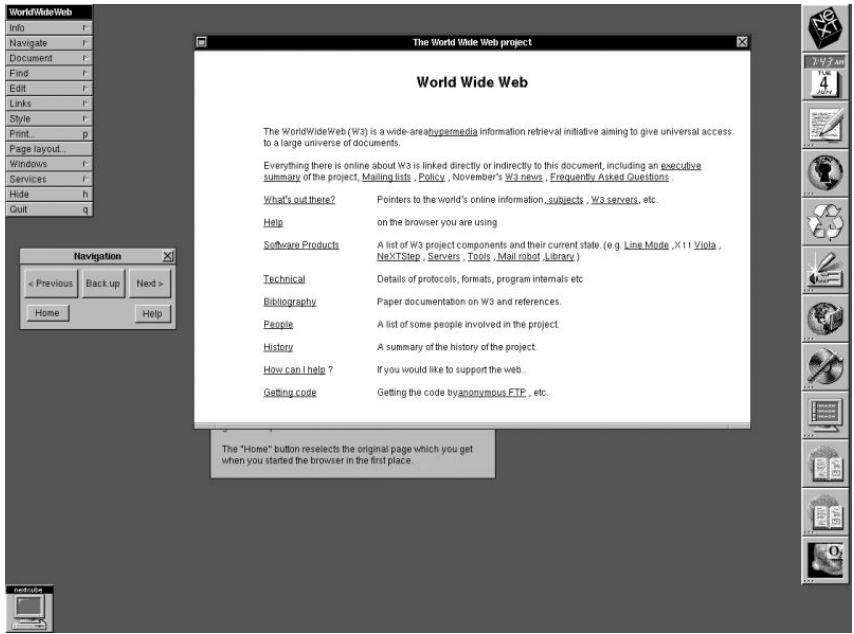
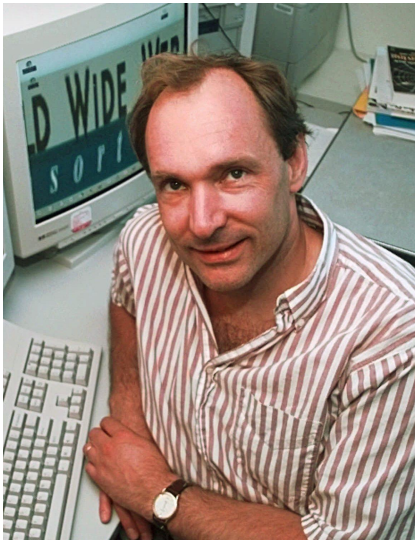


Introducción al Desarrollo con Tecnologías Web



Presentado por Nelson Aguayo
www.consultec-ti.com

Historia de la WEB



¿Cómo funciona la WEB?

Servidor responde vía HTTP con el contenido solicitado



Cliente solicita a través de una URL una página WEB

Día 1





DOM

←

→


↻

🔒

https://www.consultec-ti.com/web/consultec-ti/inicio

English

🇺🇸



Inicio

Sobre Nosotros

Servicios

Experiencia

Blog

Servicio Responsable

Contáctanos

Transformación Digital

es el momento de impulsar su negocio

Contáctanos



Sobre Nosotros

Consultec, es una empresa con operaciones en Centroamérica y el Caribe. Nuestra comprobado se ha logrado combinando la experiencia de nuestro diverso equipo en distintas áreas de negocio e industrias, junto con alianzas y apoyo de fabricantes clave, para lograr las mejores prácticas.

Ver más

¿Tiene alguna pregunta? Estoy encantado de poder ayudarle.

¡Comunícate con nosotros!



1 Solicitamos mediante una URL una página WEB

2 Recibimos el DOM en protocolo HTML

3 Nuestro navegador lo interpreta y le da un formato más amigable al ojo humano

Elements

Console

Sources

15

1

2

⚙️

⋮

✕

...<!DOCTYPE html> == \$0

<html class="ltr yui3-js-enabled webkit js chrome chrome109 chrome109-0 mac secur

e" dir="ltr" lang="es-ES">

▶<head>...</head>

▶<body class=" controls-visible yui3-skin-sam guest-site signed-out public-page

site" data-new-gr-c-s-check-loaded="14.1094.0" data-gr-ext-installed id="senna_

surface1">...</body>

</html>

<!doctype>

Styles

Computed

Layout

Event Listeners

DOM Breakpoints

Properties

>>

Filter

:hov .cls

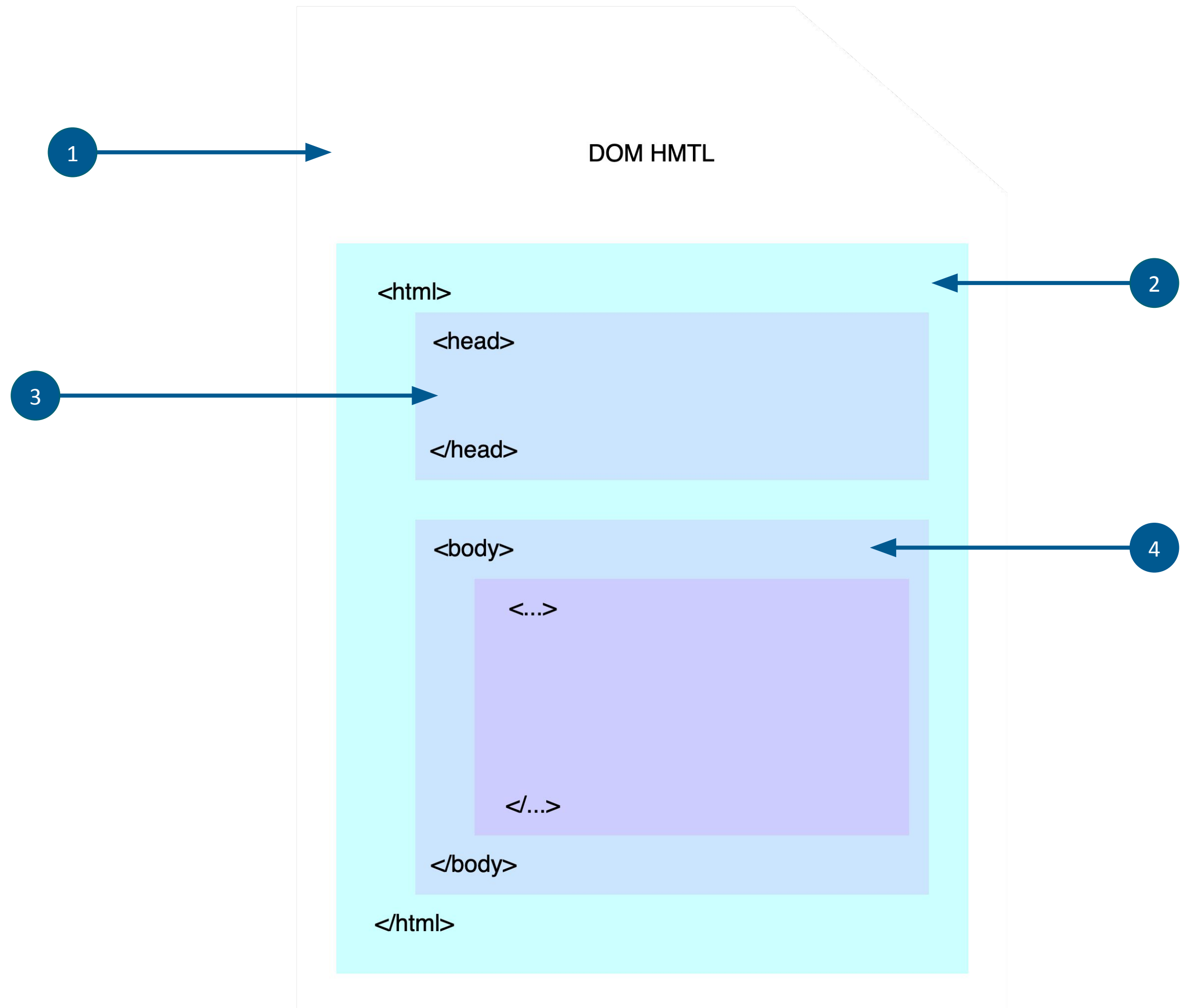
+

🖨️

🔍

No matching selector or style

DOM



Laboratorio guiado

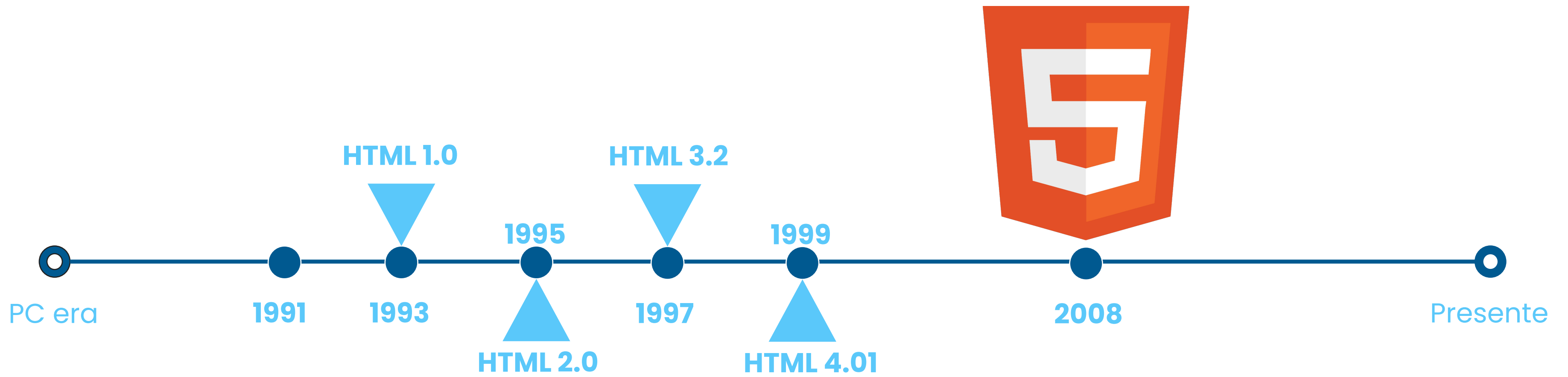
Labs 01

Acompañados del instructor, vamos seleccionar una página WEB de libre elección y vamos a inspeccionar el DOM de su homepage para validar lo aprendido anteriormente.



HTML 5






Etiquetas más comunes **en HTML**

- `<html>...</html>`
- `<head>...</head>`
- `<body>...</body>`

``

- `<p>...</p>`
- `...`
- `<a>...`
- `<div>...</div>`
- `<header>...</header>`
- `<nav>...</nav>`
- `<footer>...</footer>`

Elementos más comunes **en HTML**

 Elementos de encabezado (h1, h2, h3, etc.)

 Párrafos (p)

 Listas (ul, ol)

 Elementos de lista (li)

 Celdas de tabla (td, th)

 Enlaces (a)

 Imágenes (img)

 Botones (button)

 Tablas (table)



Labs 02

Crea una página de inicio para un blog personal. Incluye un encabezado, una imagen de perfil y algunos párrafos de texto que describan el blog.

Formularios **HTML**

Entradas de un formulario
`<form>...</form>`

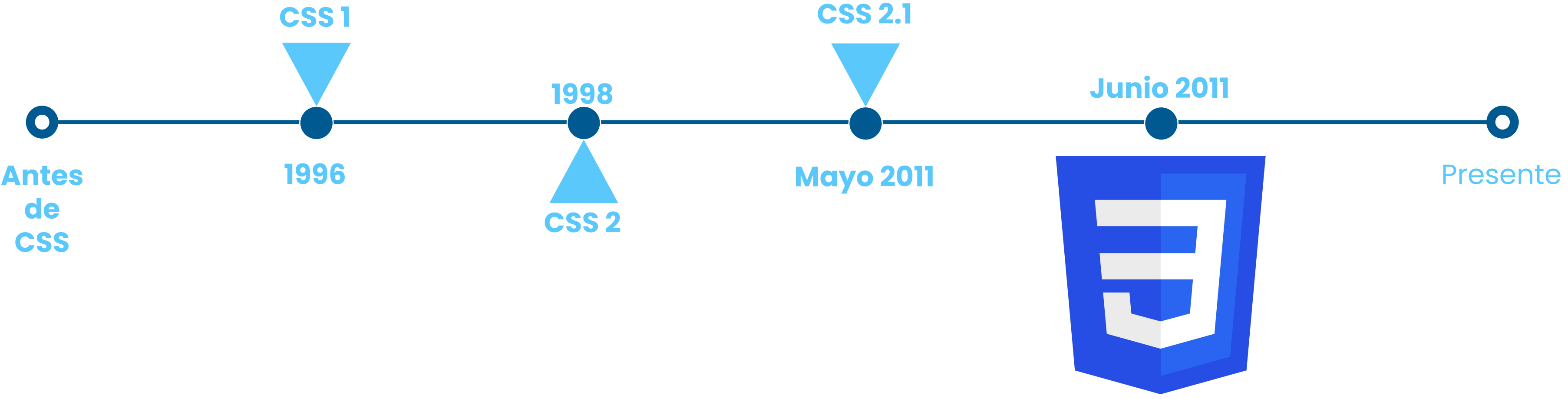
- `<select>...</select>`
- `<textarea>...</textarea>`
- `<button>...</button>`

- `<input type="text">`
- `<input type="password">`
- `<input type="radio">`
- `<input type="checkbox">`
- `<input type="submit">`





CSS



¿Cómo conectar CSS con HTML?

Usando la etiqueta <link> en el <head> del documento HTML

```
<head>  
  <link rel="stylesheet"  
type="text/css" href="estilos.css">  
</head>
```

Usando la etiqueta <style> en el <head> del documento HTML

```
<head>  
  <style>  
/* Reglas de estilo aquí */  
  </style>  
</head>
```

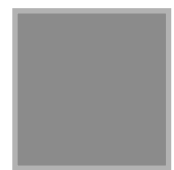
Los principios **fundamentales** de CSS



Cascada



Selección



Herencia



Especificidad



Modelo caja



Diseño responsivo



Flexbox y Grid



Transiciones y animaciones





Modificadores CSS



Clase



ID



Pseudo-clase



Pseudo-elementos



Atributos

```
<div class="content-box">  
  <h2 class="content-box__title">Título</h2>  
  <button class="content-box__button content-box__button--  
primary">Botón</button>  
</div>
```

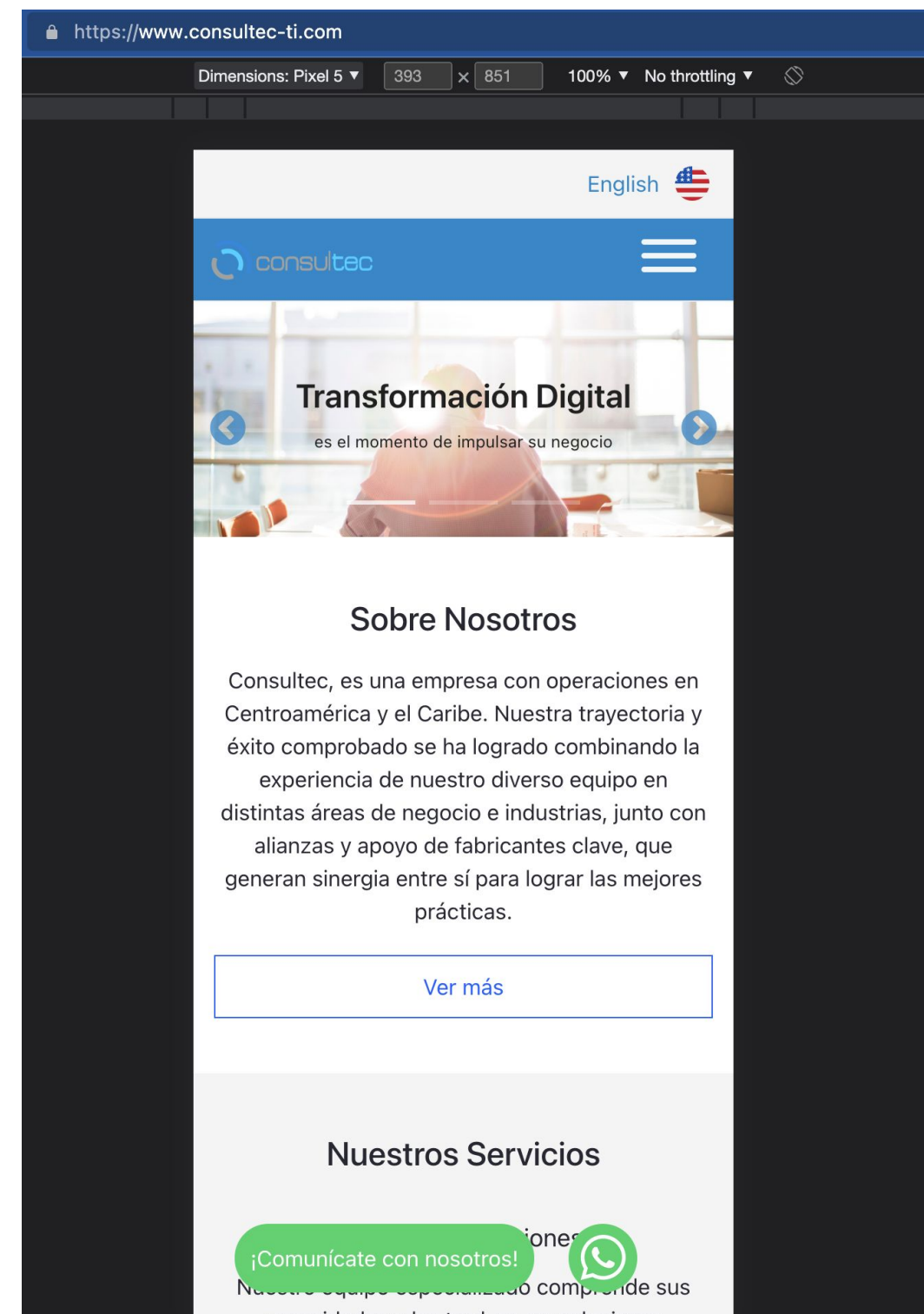


Labs 03

Continuando con la página de inicio del blog personal. Incluye estilos CSS, para hacer la página un poco más colorida.



Páginas Web Responsive



SASS

- Soporte para anidamiento de selectores, lo que permite organizar mejor el código.
- Soporte para variables, lo que permite reutilizar valores y hacer cambios globales más fácilmente.
- Soporte para mixins, que son bloques de código reutilizables que se pueden aplicar a múltiples selectores.

LESS

- Mayor compatibilidad con navegadores antiguos debido a su sintaxis similar a la de CSS.
- Menor curva de aprendizaje debido a su sintaxis similar a la de CSS.
- Mayor facilidad para integrarse con herramientas y sistemas existentes debido a su sintaxis similar a la de CSS.

Día 2



Labs 04

Continuando con la página de inicio del blog personal. Define los modificadores necesarios para que la página sea responsive y pueda ser vista en distintos tamaños de pantalla.

Páginas **WEB** estáticas

https://developer.mozilla.org/en-US/docs/Learn

Theme

Already a subscriber?

Get MDN Plus

mdn web docs

References

Guides

MDN Plus

English (US)

Guides

Related Topics

Complete beginners start here!

▶ Getting started with the web

HTML — Structuring the web

▶ Introduction to HTML

▶ Multimedia and embedding

▶ HTML tables

CSS — Styling the web

▶ CSS first steps

▶ CSS building blocks

▶ Styling text

▶ CSS layout

JavaScript — Dynamic client-side scripting

▶ JavaScript first steps

▶ JavaScript building blocks

▶ Introducing JavaScript objects

Learn web development

Welcome to the MDN learning area. This set of articles aims to guide complete beginners to web development with all that they need to start coding websites.

The aim of this area of MDN is not to take you from "beginner" to "expert" but to take you from "beginner" to "comfortable." From there, you should be able to start making your way, learning from [the rest of MDN](#), and other intermediate to advanced resources that assume a lot of previous knowledge.

If you are a complete beginner, web development can be challenging — we will hold your hand and provide enough detail for you to feel comfortable and learn the topics properly. You should feel at home whether you are a student learning web development (on your own or as part of a class), a teacher looking for class materials, a hobbyist, or someone who just wants to understand more about how web technologies work.

Looking to become a front-end web developer?

We have put together a course that includes all the essential information you need to work towards your goal.

[Get started](#)

Where to start

In this article

Where to start

Topics covered


Getting our code examples

Contact us





See also

Características de una página WEB estática

Ventajas

-  Fácil de crear y mantener
-  Rapidez
-  Seguridad
-  Costo

Desventajas

-  Contenido fijo
-  Interacción limitada
-  Escalabilidad limitada
-  Personalización limitada

JavaScript





- Se utiliza para crear efectos visuales, validar formularios, crear mapas interactivos, crear juegos, etc. Permite crear aplicaciones web más avanzadas y una mejor experiencia del usuario.
- Es un lenguaje de programación interpretado. En lugar de eso, el código se escribe directamente en un archivo HTML o en un archivo externo con extensión .js* y se ejecuta en el navegador del usuario final.
- Es un lenguaje orientado a objetos. Esto permite a los desarrolladores crear y utilizar objetos reutilizables en su código.

Tipos de datos primitivos

En primer lugar es todo aquello que **no** es un objeto



String



Boolean



Number



Undefined



BigInt



Symbol



Funciones

Las funciones son uno de los bloques de construcción fundamentales en JavaScript.

Una función es un conjunto de instrucciones que realiza una tarea, debe tomar alguna entrada y devolver una salida donde hay alguna relación obvia entre la entrada y la salida.



JavaScript **Asíncrono**

Las tareas pueden ejecutarse simultáneamente o en cualquier orden, lo que permite que el programa siga ejecutándose mientras se completan las tareas.

JavaScript provee dos mecanismo que permiten programar de forma asíncrona, llamados Eventos y Promesas.

Funciones Flecha

Inicialmente no es más que una alternativa para crear una función tradicional, sin embargo:

- No tiene sus propios enlaces a **this** o **super** y no se debe usar como métodos.
- No tiene argumentos o palabras clave.
- No apta para los métodos que generalmente se basan en establecer un ámbito o alcance
- No se puede utilizar como constructor.
- No se puede utilizar **yield** dentro de su cuerpo.



Manejo de objetos y scopes

Un objeto es una colección de propiedades, y una propiedad es una asociación entre un nombre (o clave) y un valor. El valor de una propiedad puede ser una función, en cuyo caso la propiedad es conocida como un método.

Scope de las variables

- let
- var
- const



GRACIAS

¡Nos vemos pronto!