



HTML 5

Rodrigo Alfaro Pinto

Sobre mí

- Más de 13 años en el mundo TI.
- Ingeniero Civil en Computación, Universidad de Chile.
- MBA, Universitat Politècnica de València (UPV).
- Promover el crecimiento económico de las pymes utilizando una solida infraestructura de tecnología de la información que las ayude a crear ventajas competitivas.
- Viajero, soñador y desarrollador de software amante de C.
- Fundador de Netstream, empresa chilena de consultoría IT con presencia en Perú, Bolivia, Colombia y Venezuela.
- <http://rodrigoalfaropinto.com>



Herramientas a utilizar

- Editor de código de su preferencia: Atom, Sublime Text, Geany, Rapid html, etc.
- Herramienta para el desarrollador: Google Chrome, Firefox.

Agenda día 01

- Como abordar un proyecto de desarrollo web.
- Estructura documento html.
- Desarrollando la estructura.
- Definición y Primeros pasos en CSS.
- Primer layout básico.
- Uso de herramientas del desarrollador.
- Ejercicio, mi primer sitio web en html5 (sitio web de turismo).
- Bootstrap.
- FontAwesome.

Como abordar un proyecto de desarrollo web

- Determinar y definir el proyecto:
 - Objetivos.
 - Publico objetivo.
 - Contenidos requeridos.
 - Propósito del futuro sitio web.

Como abordar un proyecto de desarrollo web

- Determinar el alcance del proyecto:
 - Tiempo de desarrollo.
 - Derechos de autor.
 - Alcance y limites del proyecto.
 - Dejar por escrito el alcance*.

Como abordar un proyecto de desarrollo web

- El diseño y desarrollo del un proyecto web:
 - Realizar un mapa del sitio.
 - Reunir información de la competencia, estilos de diseño para la industria, etc.
 - Construir un mock up funcional que contenga la información de navegación.
 - Llevar a aprobación el wireframe.

Como abordar un proyecto de desarrollo web

- El proceso de desarrollo:
 - Elegir la mejor plataforma (CMS, Custom CMS, Html)
 - Tecnología a utilizar (php, .NET, Ruby on Rails, etc.)
 - Tecnología responsive design(?).
 - Realizar set de pruebas.
 - Paso a producción.



HTML

Estructura HTML

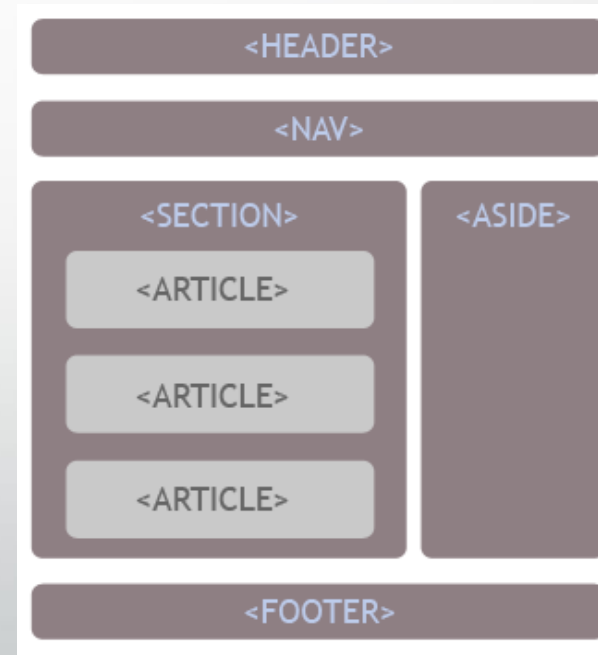
- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<meta charset="UTF-8">`
- `<title>Titulo</title>`
- `</head>`
- `<body>`
- Contenido del documento
- `</body>`
- `</html>`

Etiquetas semánticas

- **Semántica:** Perteneciente o relativo a la significación de las palabras. Esa es la definición de la Real Academia de la Lengua Española. En HTML: Aquellas que dan un significado a las partes del documento.
- Por decirlo de otra forma, estamos ante etiquetas que indican qué es el contenido que contienen, en lugar de cómo se debe formatear al mostrar el documento HTML en un cliente web.

Etiquetas semánticas

- <header>, <footer>, <article> y <section>



Etiquetas semánticas

- Las etiquetas semánticas no tienen un estilo predeterminado que el navegador nos vaya a asignar.
- Es decir, porque HEADER signifique que es una cabecera, el navegador no va en ningún caso a posicionar el elemento en la parte de arriba del documento.
- Lo mismo con FOOTER, que no lo colocará en la parte de abajo, o ASIDE al lateral.

Etiquetas semánticas

- Las usamos para:
- **Posicionamiento.** Los motores de búsqueda son capaces de asimilar mejor de lo que estamos hablando y saber qué partes del contenido de una web son las que realmente tienen más valor. Por tanto, son capaces de saber con mayor exactitud de lo que se está hablando y qué textos son más interesantes para extraer palabras clave sobre las cuales se está tratando.

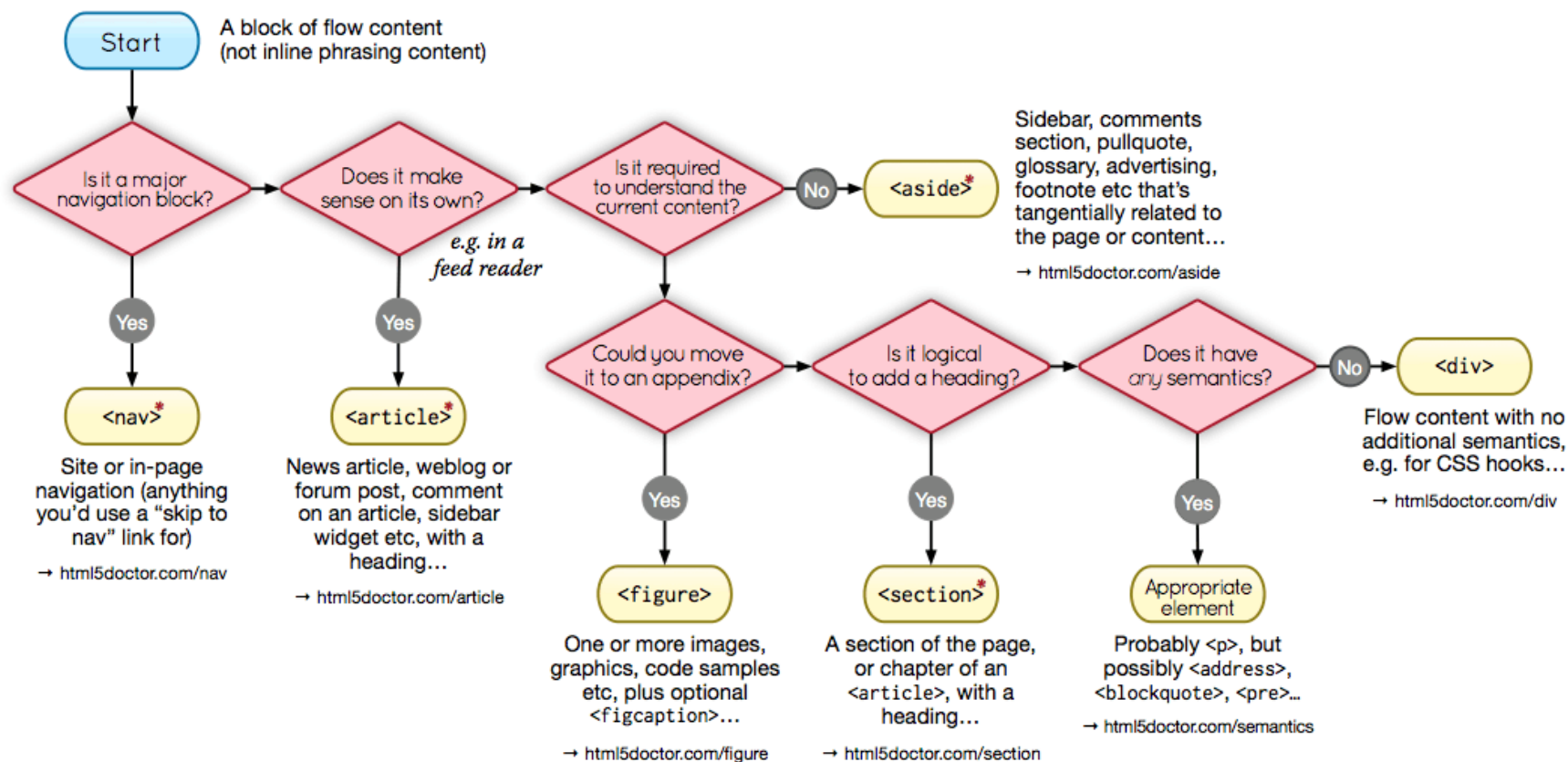


html5 Doctor

HTML5 Element Flowchart

Sectioning content elements and friends

By @riddle & @boblet
www.html5doctor.com



*Sectioning content element

These four elements (and their headings) are used by HTML5's outlining algorithm to make the document's outline
→ html5doctor.com/outline

2011-07-22 v1.5
For more information:
www.html5doctor.com/semantics

Elementos, nuevos input

- color
- date
- datetime
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url
- week

- autocomplete
- autofocus
- form
- formaction
- formenctype
- formmethod
- formnovalidate
- formtarget
- height and width
- list
- min and max
- multiple
- pattern (regexp)
- placeholder
- required
- step

Elementos, nueva sintaxis

Empty	<code><input type="text" value="John" disabled></code>
-------	---

Unquoted	<code><input type="text" value=John></code>
----------	--

Double-quoted	<code><input type="text" value="John Doe"></code>
---------------	--

Single-quoted	<code><input type="text" value='John Doe'></code>
---------------	--

Elementos, Imagen

<canvas>


Draw graphics, on the fly, via scripting (usually JavaScript)

<svg>

Draw scalable vector graphics

Elementos, Multimedia

<u><audio></u>	Defines sound content
<u><embed></u>	Defines a container for an external (non-HTML) application
<u><source></u>	Defines multiple media resources for media elements (<video> and <audio>)
<u><track></u>	Defines text tracks for media elements (<video> and <audio>)
<u><video></u>	Defines video or movie



Ejemplos en pantalla

Web storage o almacenamiento web

- Almacenamiento de datos localmente en el navegador del usuario.
- Antes de HTML5, los datos de la aplicación tenían que almacenarse en cookies, incluidos en cada solicitud del servidor. El almacenamiento web es más seguro y se pueden almacenar grandes cantidades de datos localmente, sin afectar el rendimiento del sitio web.
- A diferencia de las cookies, el límite de almacenamiento es mucho mayor (al menos 5 MB) y la información nunca se transfiere al servidor.
- El almacenamiento web es por origen (por dominio y protocolo). Todas las páginas, desde un origen, pueden almacenar y acceder a los mismos datos.

Web storage o almacenamiento web

- Web storage posee 2 objetos para almacenar datos en el browser:
- **window.localStorage**: Almacena datos sin fecha de caducidad
- **window.sessionStorage**: Almacena los datos de una sesión (los datos se pierden cuando se cierra la pestaña del navegador)

Web storage o almacenamiento web

- `if (typeof(Storage) !== "undefined") {`
- `// Code for localStorage/sessionStorage.`
- `} else {`
- `// Sorry! No Web Storage support..`
- `}`

Web storage o almacenamiento web

- El objeto **localStorage** almacena los datos sin fecha de caducidad.
- Los datos no se eliminarán cuando se cierre el navegador, y estarán disponibles al día siguiente, a la semana o al año.

Web storage o almacenamiento web

- // Store
- `localStorage.setItem("apellido", "Smith");`
- // Recuperación
- `document.getElementById("result").innerHTML = localStorage.getItem("apellido");`
- Destruir
- `localStorage.removeItem("apellido");`

Web storage o almacenamiento web

- El objeto **sessionStorage** es igual al objeto **localStorage**, excepto que almacena los datos para una sola sesión.
- Los datos se eliminan cuando el usuario cierra la pestaña específica del navegador.

Web storage o almacenamiento web

- `if (sessionStorage.clickcount) {`
- `sessionStorage.clickcount = Number(sessionStorage.clickcount) + 1;`
- `} else {`
- `sessionStorage.clickcount = 1;`
- `}`
- `document.getElementById("result").innerHTML = "You have clicked the button " +`
- `sessionStorage.clickcount + " time(s) in this session.";`

Web storage o almacenamiento web

- Ver ejemplo 8.html

Web workers

- Un web worker es un JavaScript que se ejecuta en segundo plano, independientemente de otros scripts, sin afectar el rendimiento de la página. Puede continuar haciendo lo que quiera: hacer clic, seleccionar cosas, etc., mientras el trabajador web se ejecuta en segundo plano.
- Ver ejemplo [og.html](#)

Server-Sent Events - One Way Messaging

- Un evento enviado por el servidor es cuando una página web recibe actualizaciones automáticamente de un servidor.
- Esto también fue posible antes, pero la página web debería preguntar si había actualizaciones disponibles. Con los eventos enviados por el servidor, las actualizaciones vienen automáticamente.
- Ejemplos: actualizaciones de Facebook / Twitter, actualizaciones de precios de acciones, noticias, deportes, etc.

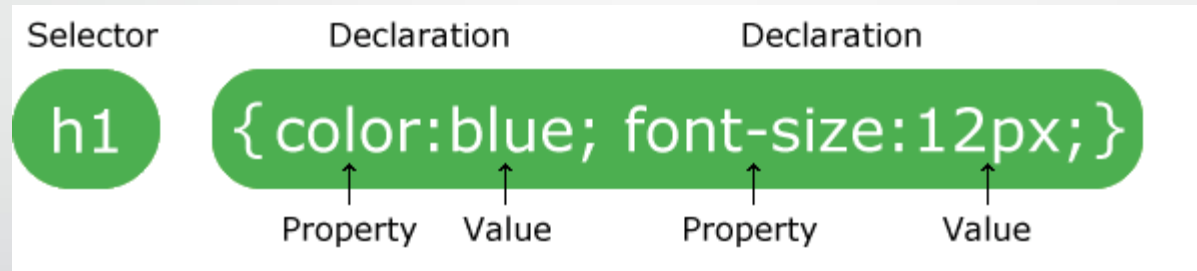


CSS

CSS

- CSS es un lenguaje que describe el estilo de un documento HTML.
- Se utiliza para definir estilos para páginas web, incluido el diseño y las variaciones en la pantalla para diferentes dispositivos.

CSS, Sintaxis



CSS, Archivo externo

- `<link rel="stylesheet" type="text/css" href="estilo.css">`



CSS, Tag

- `<style></style>`

CSS

- Una buena practica es utilizar los colores en formato hexadecimal.
- <https://htmlcolorcodes.com/es/>

CSS

- Ver demostración en pantalla. Ejercicio 11.html

Ejercicio

- Realizar una primera plantilla para una agencia de turismo, que desea aumentar sus ventas.
- <https://pastebin.com/YccS6jPC> //archivo index.html
- <https://pastebin.com/AepBTMuY> // estilo.css



Bootstrap

Librerías para CSS

[Home](#) [Documentation](#) [Examples](#) [Themes](#) [Jobs](#) [Expo](#) [Blog](#) v4.1 [GitHub](#) [Twitter](#) [Facebook](#) [Download](#)


Bootstrap

Build responsive, mobile-first projects on the web with the world's most popular front-end component library.

Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. Quickly prototype your ideas or build your entire app with our Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery.


[Get started](#) [Download](#)

Currently v4.1.0



Ejercicio Final día 01

- Utilizando bootstrap, cree un sitio web de turismo que contenga:
 - Layout de login de usuarios.
 - Pagina que liste resultados y muestre una paginación.
 - Pagina que muestre una grilla de noticias.
 - Pagina index.html que contenga banner autoajustable.

- 
- Muchas gracias por su atención.