



## UNIDAD DE TRABAJO 2:

# HOJAS DE ESTILO

DISEÑO DE INTERFACES WEB | 2º DESARROLLO DE APLICACIONES WEB  
CIFP CAMINO DE LA MIRANDA (PALENCIA)



# INDICE

1. HTML y HTML5
2. CSS3

# THE FUTURE IS LANDING

# 1

## HTML Y HTML5

## 1.1 ¿qué es html?

- Herramientas a instalar en VS Code:
  - Spanish language pack
  - Auto rename tag
  - Auto close tag
  - Prettier
  - Live Server

## 1.1 ¿qué es html?

- Es un lenguaje de marcado de hipertexto (Hyper Text Markup Language).
- HTML no es un lenguaje de programación, es un lenguaje de estructura.
- Es la base con la que están creadas TODAS las páginas web del mundo.
- Cada etiqueta le dice al navegador y a los motores de búsqueda cuál es la estructura de los documentos, elementos, organización, etc.

## 1.1 Historia de HTML

- 1989- Inicio de su desarrollo. (Era un boceto de lo que se quería conseguir)
- 1991- Lanzamiento de la web (3 conceptos: http, html, url).
- En 1991 **Sir Tim Berners-Lee** escribió un documento denominado «*HTML Tags*» que describía los 20 elementos que deberían ser utilizados para escribir documentos web.
- En 1994 se creó la W3C (Consortio que define los estándares de la web)

## 1.1 HISTORIA DE HTML5

En noviembre de 1995 la **Internet Engineering Task Force (IETF)** publicó la primera especificación de HTML, denominada **HTML 2.0**



## 1.1 HISTORIA DE HTML5

HTML 3.2 fue liberado en 1997 y el año siguiente se publicó **HTML 4** con tres variaciones: Strict, Transitional y Frameset.

Aquí comenzaron las guerras de navegadores ya que empezó a haber diferencias en la forma en que cada navegador interpretaba HTML.





## 1.1 HISTORIA DE HTML5

Tras HTML 4 el **World Wide Web Consortium (W3C)** decidió suspender el desarrollo de HTML para trabajar en un nuevo lenguaje basado en XML, el **Extensible Hypertext Markup Language (XHTML) 1.0**.

No añadía elementos nuevos respecto a HTML 4 sino que se centraba en unas reglas sintácticas muy estrictas.

## 1.1 HISTORIA DE HTML5

Por esta época (2001) comenzaron a hacerse cada vez más populares las **Hojas de Estilo en Cascada (CSS)**.

En 2002 el W3C liberó el borrador de **XHTML** 1.1.

Si la versión 1.0 era HTML con toques de XML, la versión 1.1 era esencialmente XML.

Esto hacía que si era utilizado para desarrollo web no podría visualizarse en prácticamente ningún navegador.

## 1.1 HISTORIA DE HTML5

En 2004, gente de Apple, Mozilla y Opera comenzaron a trabajar en su propia especificación que fuera compatible hacia atrás.

El W3C rechazó esta propuesta pero este grupo decidió seguir denominándose **Web Hypertext Application Technology Working Group (WHATWG)**.



Web Hypertext Application Technology Working Group

## 1.1 HISTORIA DE HTML5

Este grupo comenzó a trabajar en dos especificaciones:

- Web Forms 2.0
- Web Applications 1.0

Dirigido por **Ian «Hixie» Hickson**, las ideas se buscaron con ayuda de grupos de correo públicos.

Estos dos proyectos se fusionaron en lo que conocemos como **HTML5**

## 1.1 HISTORIA DE HTML5

De forma paralela el W3C seguía trabajando en la especificación de **XHTML 2**.

Pero en 2006 se dieron cuenta que había sido un error abandonar HTML por lo que pasaron a una nueva especificación basándose en los progresos del WHATWG.

Poco después abandonó definitivamente XHTML 2.

## 1.1 HISTORIA DE HTML5

Desde entonces convivimos con dos especificaciones.

El WHATWG quitó en 2011 el 5 del nombre pasando a denominarse HTML y poco después lo nombró como **HTML Living Standard**.

La especificación de W3C sigue llamándose HTML5 y actualmente están trabajando en la versión 5.2

# 1 DIFERENCIAS CON XHTML

## ¿Qué diferencias tenemos que tener en cuenta con HTML5 respecto a XHTML?

- No es necesario cerrar los elementos vacíos.

```
<input type="text">
```

- Algunos elementos (p.e. <p>) no es necesario cerrarlos, pero nosotros los cerraremos para tener un código más limpio.
- El uso indistinto de mayúsculas y minúsculas no afecta a la validación de las páginas. Pero por claridad es aconsejable utilizar siempre minúsculas.

# 1. DIFERENCIAS CON XHTML

- Tampoco es obligatorio poner comillas en los atributos. Hay ocasiones en cambio en que si será necesario, por ejemplo cuando declaramos múltiples clases separadas por espacios.

```
<article class="column wrapper">
```

Por consistencia es aconsejable poner siempre las comillas.

- No es necesario poner el valor en los atributos booleanos.

```
<input type="checkbox" checked>
```

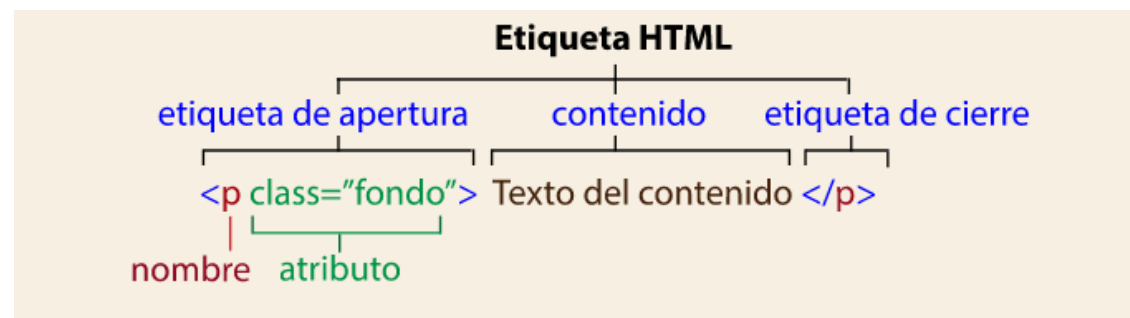


## 1.3 PLANTILLA DE HTML5

### Plantilla de HTML5

HTML se basa en etiquetas.

Las etiquetas (marcas) delimitan cada uno de los elementos que componen un documento HTML. Su sintaxis es la siguiente:



Lo primero que vamos a ver son los elementos fundamentales que debe tener cualquier documento HTML.

# 1.3 PLANTILLA DE HTML5

```
ej01.html
1 <!doctype html>
2 <html lang=es>
3 <head>
4     <meta charset="utf-8">
5     <title>El título de mi página</title>
6     <meta name="description" content="Primera prueba"/>
7     <meta name="author" content="Victor J. González"/>
8
9     <link rel="stylesheet" href="css/styles.css">
10
11     <!--[if lt IE 9]>
12     <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js">
13     </script>
14     <![endif]-->
15 </head>
16 <body>
17     <script src="js/scripts.js"></script>
18 </body>
19 </html>
```

## 1.3 PLANTILLA DE HTML5

### DOCTYPE

El **Document Type Declaration** (Doctype) tiene por finalidad indicarle al navegador de qué tipo de documento se trata.

En HTML5 se indica de la forma:

```
<!doctype html>
```

## 1.3 PLANTILLA DE HTML5

### El elemento <html>

No tiene cambios muy significativos.

El atributo xmlns, requerido en XHTML ya no es necesario.

El atributo lang tampoco es obligatorio.

```
<html lang=es>
```

## 1.3 PLANTILLA DE HTML5

### El elemento <head>

La primera línea dentro de head es la que define la codificación de caracteres.

Esta línea también se ha simplificado.

Para evitar problemas esta línea tiene que estar en los primeros 512 caracteres del documento y antes de cualquier contenido (p.e. <title>)

```
<meta charset="utf-8">
```

## 1.3 PLANTILLA DE HTML5

Otra característica dentro del head es que al enlaza con las hojas de estilo ya no es necesario el atributo `type="text/css"`

```
<link rel="stylesheet" href="css/styles.css">
```

## 1.3 PLANTILLA DE HTML5

### El elemento <body>

Representa el contenido de un documento HTML. Sólo puede haber un elemento <body> en un documento.

```
<html>
  <head>
    <title>Título del documento</title>
  </head>
  <body>
    <p>Este es un párrafo</p>
  </body>
</html>
```

# 1.3 PLANTILLA DE HTML5

## Encabezados

Los elementos de encabezado implementan seis niveles de encabezado del documento, `<h1>` es el más importante, y `<h6>`, el menos importante. Un elemento de encabezado describe brevemente el tema de la sección que presenta.

### **Importante:**

No se deben usar los niveles inferiores para reducir el tamaño de la fuente: se debe usar css

Evitar omitir niveles de encabezado, siempre comenzar usando `<h1>`, después `<h2>` y así sucesivamente.

Con `<section>` sólo se debe usar un `<h1>` por sección-



# 1.3 PLANTILLA DE HTML5

## Encabezados

Ejemplos:

```
<h1>Heading level 1</h1>  
<h2>Heading level 2</h2>  
<h3>Heading level 3</h3>  
<h4>Heading level 4</h4>  
<h5>Heading level 5</h5>  
<h6>Heading level 6</h6>
```

**Heading level 1**

**Heading level 2**

**Heading level 3**

**Heading level 4**

**Heading level 5**

**Heading level 6**

# 1.3 PLANTILLA DE HTML5

## Encabezados

```
<h1>Heading elements</h1>  
<h2>Summary</h2>  
<p>Some text here...</p>
```

```
<h2>Examples</h2>  
<h3>Example 1</h3>  
<p>Some text here...</p>
```

```
<h3>Example 2</h3>  
<p>Some text here...</p>
```

```
<h2>See also</h2>  
<p>Some text here...</p>
```

## Heading elements

### Summary

Some text here...

### Examples

#### Example 1

Some text here...

#### Example 2

Some text here...

### See also

Some text here...

# 1.3 PLANTILLA DE HTML5

## Encabezados

No hacer

```
<h1>Cabecera nivel 1</h1>  
<h3>Cabecera nivel 3</h3>  
<h4>Cabecera nivel 4</h4>
```

Sí hacer

```
<h1>Cabecera nivel 1</h1>  
<h2>Cabecera nivel 2</h2>  
<h3>Cabecera nivel 3</h3>
```

# 1. ETIQUETAS SEMÁNTICAS

Pasemos a las nuevas etiquetas de HTML5 o etiquetas de sección de contenido

**Etiquetas semánticas:** *su objetivo es describir qué es el contenido que contienen, en lugar de cómo se debe formatear.*

**Ventajas:** los motores de búsqueda comprenderán mucho mejor el contenido de tu página

# 1. ETIQUETAS SEMÁNTICAS

## El elemento `<header>`



Un grupo de ayudas introductorias o de navegación.

El primer elemento que nos encontramos en cualquier página web es la cabecera.

Su uso no está limitado a la cabecera de la página, también se puede utilizar en los encabezados de las secciones.

Se podría introducir el menú de navegación, el logotipo de la página, etc.

# 1. ETIQUETAS SEMÁNTICAS

## El elemento `<main>`



Representa la sección del contenido principal del body de un documento o aplicación

Hay bastante contradicción en esta etiqueta entre la especificación WHATWG y la de W3C.

El WHATWG indica que puede haber varios `main` por página mientras que el W3C especifica que debe haber únicamente uno.

Igualmente esta etiqueta no puede estar contenida dentro de `article`, `aside`, `footer`, `header` o `nav`

# 1. ETIQUETAS SEMÁNTICAS

## El elemento `<footer>`



Representa un pie de página para la sección de contenido que es su ascendiente más próximo.

Cuando en la definición se habla de sección puede ser el documento entero pero también un elemento `section`, `article` o `aside`.

# 1. ETIQUETAS SEMÁNTICAS

Esta etiqueta puede contener:

- Información de copyright
- Lista de enlaces relacionados
- Información sobre el autor
- ...

Aunque habitualmente se sitúa en la parte inferior esto no es realmente necesario.

Por ejemplo, la información sobre el autor de un post de un blog puede estar antes del post.



# 1. ETIQUETAS SEMÁNTICAS

## El elemento `<section>`



Representa una sección genérica de un documento o aplicación. Una sección, en este contexto, es un agrupamiento temático de contenido, habitualmente con un encabezado.

Es un contenedor genérico que agrupa contenido que está relacionado. Cuando creamos bloques cuyo contenido es parte de un bloque total usaremos section.

No se debe utilizar para secciones cuyo objetivo sea aplicar estilos o un script. En ese caso hay que utilizar `<div>`

# 1. ETIQUETAS SEMÁNTICAS

Ejemplos donde sería aceptable su uso:

- Cada sección de una interfaz con pestañas
- Partes de una página "Acerca de", por ejemplo en una compañía:
  - Historia de la compañía
  - Objetivos
- Diferentes partes de una página larga de "términos de servicio"
- Varias secciones de un sitio de noticias. Por ejemplo los artículos pueden estar agrupados en secciones de deportes, economía,...

# 1. ETIQUETAS SEMÁNTICAS

Cuando no tengamos claro si utilizar esta etiqueta tenemos que tener en cuenta lo siguiente:

- `section` es genérica, si hay otro elemento semántico más apropiado (`article`, `aside` o `nav`) es mejor usar este.
- `section` tiene significado semántico lo que implica que su contenido está relacionado. Si no eres capaz de describir brevemente todo el contenido dentro del `section` es mejor utilizar el contenedor semánticamente neutral `div`.

# 1. ETIQUETAS SEMÁNTICAS

## El elemento `<article>`



Representa una **composición completa, o auto-contenida** en un documento, página, aplicación o sitio y que es, en principio, **independientemente distribuible** o reusable.

Es un contenedor que representa contenido independiente, es decir, si se lee ese fragmento que está dentro de un artículo, daría igual que no hayamos leído el resto de la página ya que por sí sólo tendría sentido.

El article puede contener un header y un footer, no es obligatorio

# 1.4.1. ETIQUETAS SEMÁNTICAS

Ejemplos donde sería aceptable su uso:

- Un artículo en una revista o periódico
- Una entrada en un blog
- Un comentario de un usuario acerca de un artículo
- Un post en un foro

Tanto `<section>` como `<article>` se pueden anidar de cualquier manera.

# 1. ETIQUETAS SEMÁNTICAS

## Ejemplos de articles incluidos en section:

```
<!--Establece el tipo de documento-->
<!DOCTYPE html>
<!--Establece el lenguaje de la pagina web-->
<html lang="es">
<head>
  <title>Document</title>
</head>
<body>
  <header>
    <h1>NOTICIAS NACIONALES</h1>
  </header>
  <main>
    <section>
      <h2>Noticias del día</h2>
      <article>Noticia1</article>
      <article>Noticia1</article>
      <article>Noticia1</article>
    </section>
  </main>
  <footer>3-10-2022</footer>
</body>
</html>
```

# 1. ETIQUETAS SEMÁNTICAS

Si lo validamos con Validator w3c.org da un error a la hora de utilizar el article porque es una etiqueta que necesitan un header ya que son independientes.

Otro ejemplo:

- Estructura de una etiqueta <article>

```
<body>
  <header>
    <h1>NOTICIAS NACIONALES</h1>
  </header>
  <main>
    <section>
      <h2>Noticias del día</h2>
      <article>
        <header>
          <h3>Noticia1</h3>
        </header>
        <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Quas nam odio, sapiente atque eaque ducimus illum quia quidem repudiandae ut nostrum magni vel tempore sequi, veritatis veniam minus saepe at?</p>
        <footer>Noticia Madrid</footer>
      </article>
      <article>
        <h3>Noticia2</h3>
        <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Quas nam odio, sapiente atque eaque ducimus illum quia quidem repudiandae ut nostrum magni vel tempore sequi, veritatis veniam minus saepe at?</p>
      </article>
      <article>
        <h3>Noticia2</h3>
        <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Quas nam odio, sapiente atque eaque ducimus illum quia quidem repudiandae ut nostrum magni vel tempore sequi, veritatis veniam minus saepe at?</p>
      </article>
    </section>
  </main>
```

# 1. ETIQUETAS SEMÁNTICAS

- Diseña en un documento html una etiqueta <section> dentro de una etiqueta <article> y comprueba con [Validator3wc.org](http://Validator3wc.org) si se permite o no.
- Realiza la actividad PRO101