CONQUER

BLOCKS

HTML

HTML

<HTML>

Clase 01

CONQUER BLOCKS

HTML-C01

<Índice>

Clase 01: Introducción a HTML

¿Por qué aprender HTML hoy en día?

¿Qué es el Frontend?

¿Qué es el Backend?

¿Qué es FullStack?

Estático vs Dinámico

CONQUER BLOCKS

¿Por qué aprender HTML hoy en día?

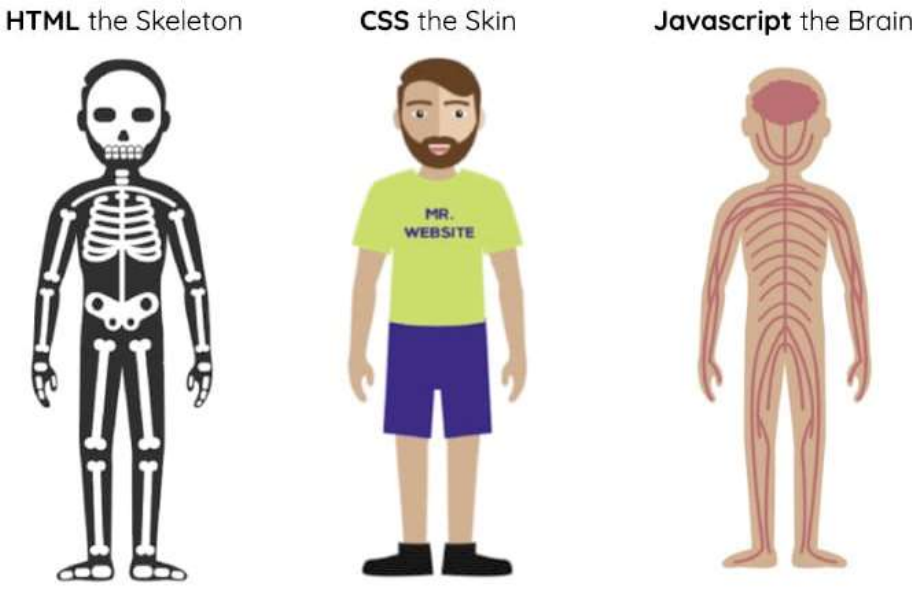
HTML-C01

¿Por qué aprender HTML hoy en día?

HTML CSS JavaScript

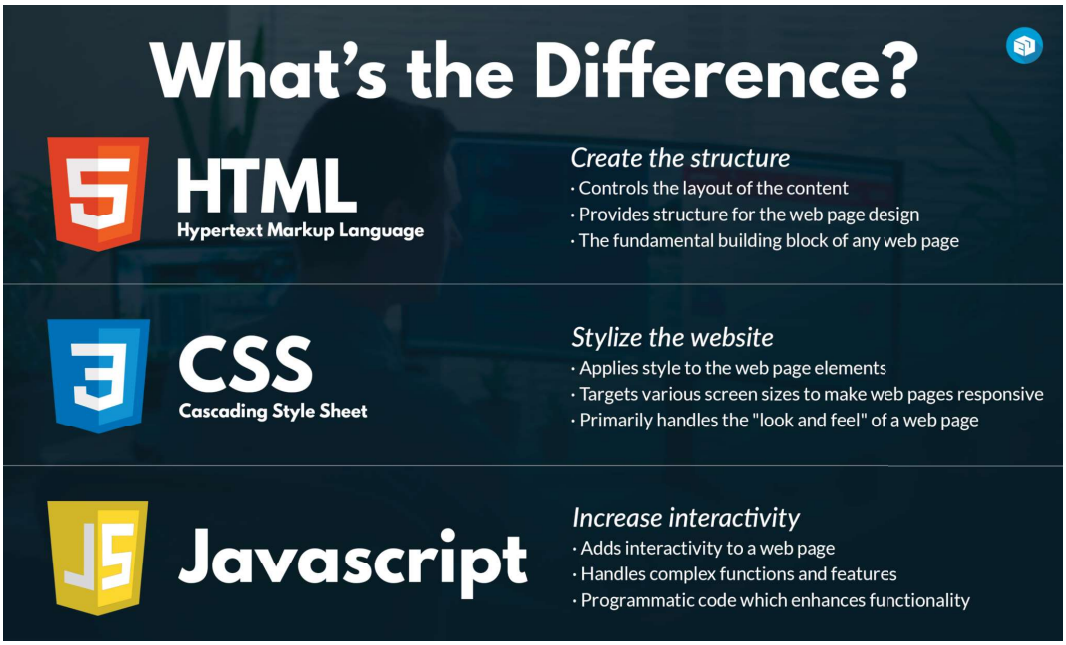
CONQUERBLOCKS

¿Por qué aprender HTML hoy en día?



HTML-C01

¿Por qué aprender HTML hoy en día?



CONQUERBLOCKS

¿Por qué aprender HTML hoy en día?

Razones por las que aprender HTML

- Es el responsable de estructurar todos los contenidos de una web
- Trabajes en lo que trabajes debes conocer HTML
- Toda la web está hecha de HTML
- Son los cimientos de toda web o webapp

HTML-C01

¿Por qué aprender HTML hoy en día?

HTML

Características

1. Texto plano
2. Se puede editar con cualquier editor
3. No necesita ser compilado ni traspilado
4. No es un lenguaje de programación
5. Es un lenguaje de marcado de la información
6. Nació con internet

CONQUERBLOCKS

¿Por qué aprender HTML hoy en día?

HTML-C01

HTML se enseña mal

CONQUERBLOCKS

¿Por qué aprender HTML hoy en día?

HTML-C01

- Mucho tutorial de un solo uso
- Documentación anticuada
- Se estudia programación web sin saber las bases que sustenta toda la web
- Se necesite un aprendizaje incremental y en detalle
- No mezclar tecnologías, al menos al principio

CONQUERBLOCKS

¿Por qué aprender HTML hoy en día?

Perfiles profesionales en la web

- Frontend
- Backend
- Maquetador
- FullStack
- SEO
- Analista web
- SEM
- DevOps
- QA

- Diseñador
- UI/UX
- Márketing Digital
- Automatizaciones
- Accesibilidad
- Redactor
- Creador de contenido

¿Por qué aprender HTML hoy en día?

Para todo lo anterior necesitas entender HTML

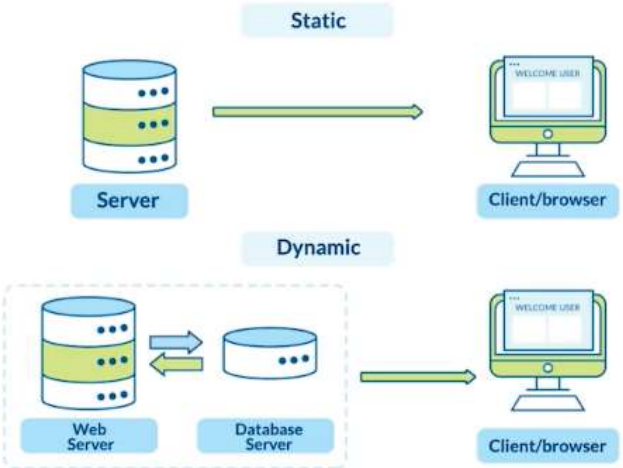
¿Por qué aprender HTML hoy en día?

Perfiles profesionales en la web

- Frontend
- Backend
- Maquetador
- FullStack
- SEO
- Analista web
- SEM
- DevOps
- QA
- Diseñador
- UI/UX
- Márketing Digital
- Automatizaciones
- Accesibilidad
- Redactor
- Creador de contenido

Estático vs Dinámico

Estático vs Dinámico



Estático vs Dinámico

Página estática

- No dispone de base de datos
- No dispone de panel de administración
- Sólo dispone de ficheros **HTML**, CSS y JS
- Solo puede ser actualizada por un desarrollador

Estático vs Dinámico

Página estática

- No suele disponer de lenguaje de programación de parte del servidor
- Se mostrarán siempre iguales para todos los usuarios
- No hay registro/login ni sesión
- Sin interacción o sólo en una vía

HTML-C01

Estático vs Dinámico

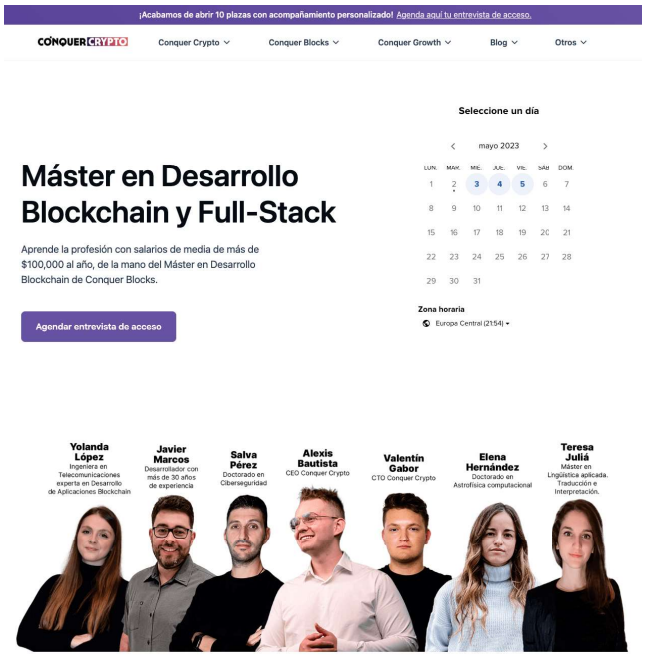
Página estática

- Páginas corporativas
- Landing pages
- Páginas informativas
- Blog (*)

CONQUERBLOCKS

Estático vs Dinámico

Página estática Ejemplo



HTML-C01

Estático vs Dinámico

Página dinámica

- Dispone de motor de base de datos
- Dispone de lenguaje de programación de parte del servidor
- Dispone de lógica asociada a los contenidos

CONQUERBLOCKS

Estático vs Dinámico

HTML-C01

Página dinámica

- También tiene **HTML**, CSS y JS
- Sesión de usuarios/login/registro
- Interactividad en las dos vías
- Contenidos únicos por usuario

Estático vs Dinámico

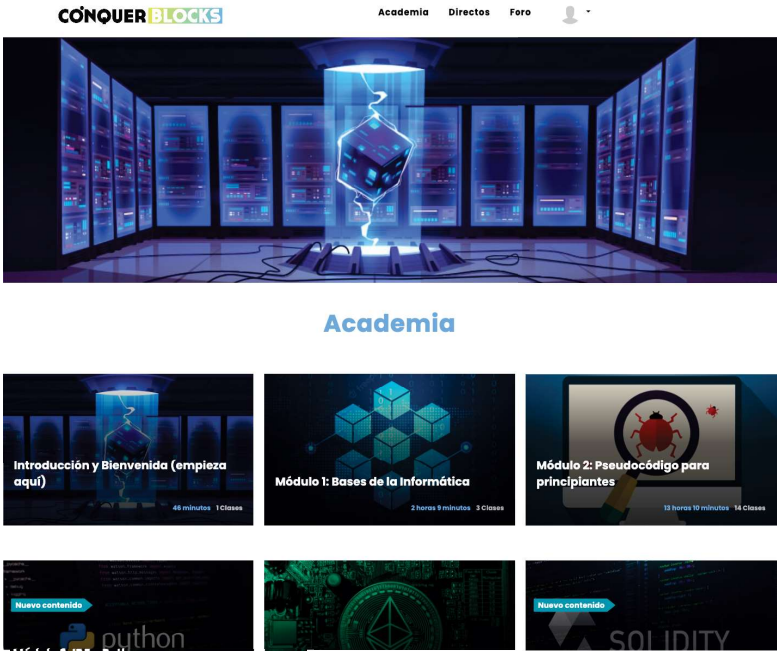
HTML-C01

Página dinámica ejemplos

- Twitter
- Página de entidad bancaria
- Netflix
- Canva

Estático vs Dinámico

Página dinámica ejemplo

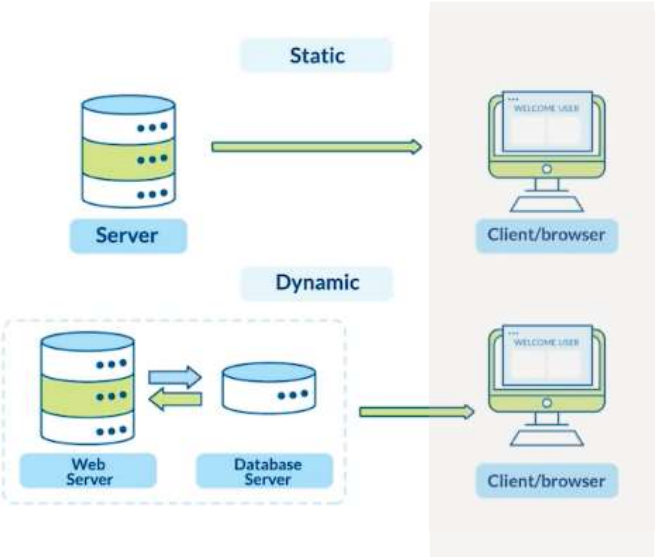


Frontend

Frontend

¿Qué es el Frontend?

Frontend



Frontend

- Será el encargado de todo lo que ve el cliente/navegador
- Sólo existen tres tecnologías principales **HTML**, CSS y JS
- Controlará que, según el dispositivo, se muestre la información de la manera más apropiada
- Enviará información al Backend

Frontend

Un profesional Frontend debe saber:

- Estructurar la información con HTML
- Maquetar la información y mostrarla de la forma correcta aplicando estilos de CSS
- Programar la interactividad deseada con el usuario desde el único lenguaje de programación que entiende el navegador JS

Frontend

HTML

- HyperText Markup Language
- Añadió la funcionalidad de poder saltar de un documento a otro
- El lenguaje más importante de todos, sin él no existiría la web tal y como la conocemos
- Nos da las herramientas para organizar la información semántica y ordenadamente
- Al que se le suele dar menos cariño en las formaciones

Frontend

CSS

- 2º Lenguaje más importante en la web
- Se encarga de presentar la información marcada previamente mediante HTML
- Ayuda a poder separar la información de la presentación
- Evoluciona junto a HTML y los diferentes motores de los navegadores

Frontend

JavaScript

- Único lenguaje de programación que entiende el navegador
- Es un lenguaje de programación completo que es capaz de interactuar con el DOM
- Evoluciona muy rápido debido a su gran uso
- Todos los navegadores saben interpretarlo
- Se puede usar también en el backend con Node.JS

Frontend

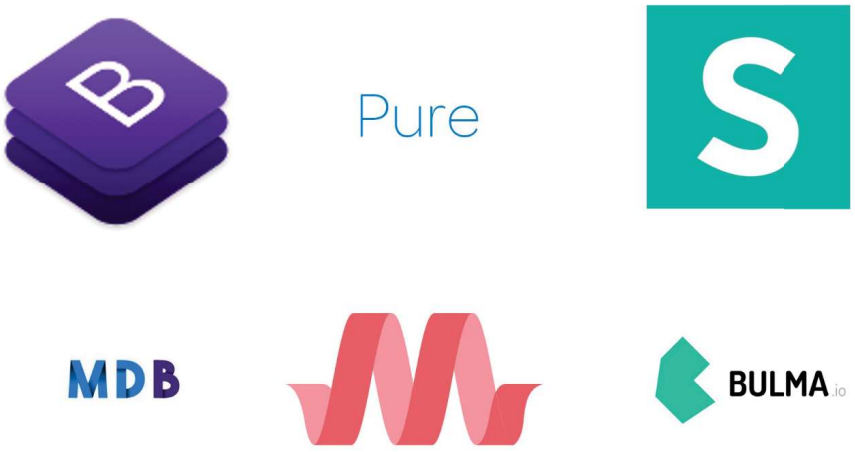
No solo de HTML, CSS y JavaScript vive el Frontend

Frontend

Tecnologías del Frontend

Frontend

Framework CSS

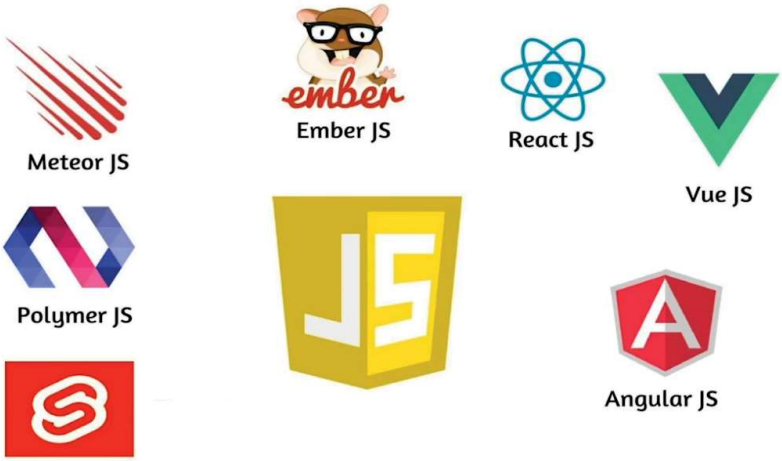


Frontend

Preprocesadores CSS

Frontend

Framework JS



Frontend

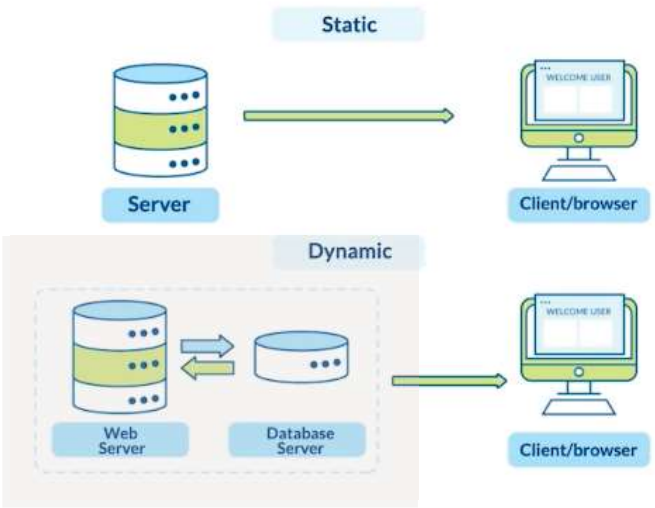
Luego os explico

Backend

Backend

¿Qué es el Backend?

Backend



Backend

- Será el encargado de toda la lógica de la aplicación y negocio de la misma
- Controla la comunicación con el Frontend, es decir con el usuario/cliente que interactúe
- Controla la comunicación con la base de datos

Backend

- Gestiona la seguridad e integridad de la información de la aplicación web
- Controla los usuarios y pemisos de acceso
- Provee mediante API, información a otra aplicaciones web
- Gestión con servicios externos o interconexiones

Backend

HTML-C01

Componentes

- Lenguaje de parte del servidor
- Framework Backend
- Motor de base de datos
- Servidor Web

CONQUERBLOCKS

Backend

HTML-C01

Lenguajes de parte del servidor

- PHP
- Python
- Java
- Node.JS
- Ruby

CONQUERBLOCKS

Backend

Frameworks del Backend

- PHP => Laravel
- Python => Django
- Java => Spring
- Node.JS => Express
- Ruby => Rails

Backend

Luego os explico

Backend

HTML-C01

Motor de Base de Datos

- Persistencia de datos
- Seguridad
- Control de usuarios
- Copias de seguridad
- Rapidez e integridad

CONQUERBLOCKS

Backend

HTML-C01

Motor de Base de Datos

- Relacionales
- No-Relacionales

CONQUERBLOCKS

Backend

Motor de Base de Datos

Relacionales

- MariaDB
- PostgreSQL

No-Relacionales

- Mongo DB
- Redis

Backend

Luego os explico

Backend

Servidor Web

- Encargado de recibir las peticiones
- Procesa y gestiona las rutas
- Controla las sesiones de navegador
- Interpreta el lenguaje de parte del servidor y devuelve la respuesta

HTML-C01

Backend

Servidor Web

- Apache
- NginX
- Tomcat
- IIS

CONQUERBLOCKS

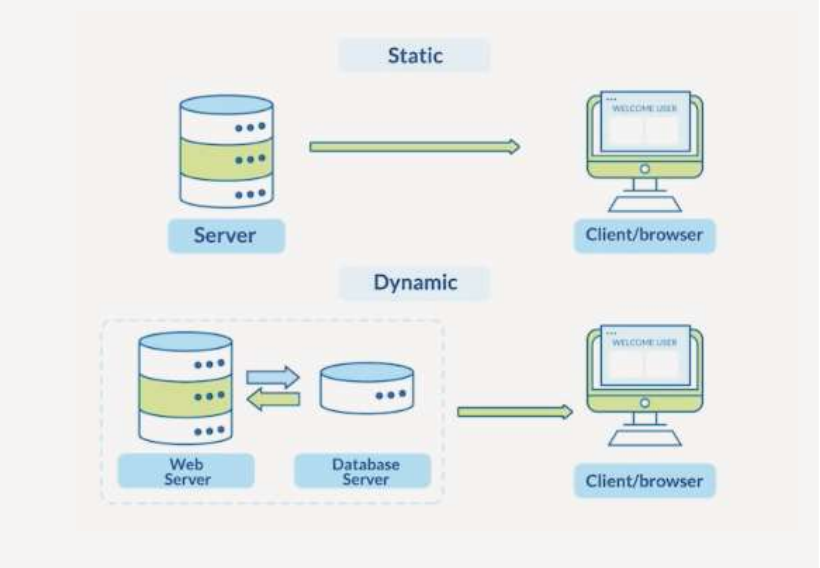
Backend

Va cogiendo forma

Web Backend	node.js, Express, METEOR, GO, RAILS, .NET, .NET Core, Java, php, python, django
Databases	MySQL, Microsoft SQL Server, PostgreSQL, mongoDB, OrientDB, Cassandra, DynamoDB, neo4j, elastic, redis

FullStack

¿Qué es el FullStack?



FullStack

Ya no basta con esto

FullStack

- Comprende tanto de Backend como de Frontend
- Normalmente siempre tiene un perfil predilecto
- Toma deciones de diseño y arquitectura de la solución
- Entender cómo interactúan Frontend y Backend
- Toma decisiones de arquitectura de la aplicación en las dos vías
- Despliega el stack en el servidor
- Controla además los recursos del mismo
- Gestiona la escalabilidad de la aplicación web
- Gestiona las automatizaciones
- ...

FullStack

HTML-C01

Web Frontend	HTML, CSS, JS, Bootstrap, jQuery, ANGULAR, React, Vue.js, BACKBONE.JS, ELECTRON
Web Backend	node.js, Express, METEOR, GO, RAILS, .NET, .NET Core, Java, php, python, django
Databases	MySQL, Microsoft SQL Server, PostgreSQL, mongoDB, OrientDB, Cassandra, DynamoDB, neo4j, elastic, redis
Cloud/ Integration	amazon web services, Google Cloud Platform, Microsoft Azure, DigitalOcean, HEROKU

CONQUERBLOCKS

FullStack

HTML-C01

CONQUERBLOCKS

<Despedida>

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

<HTML>

Clase 02

CONQUER BLOCKS

HTML-C02

<Índice>

Clase 02 Un poco de historia

Tim Bernes Lee

¿Cómo funciona la web?

¿Qué son las aplicaciones web?

SGML => XML => HTML

¿Qué es una URL?

HTTP

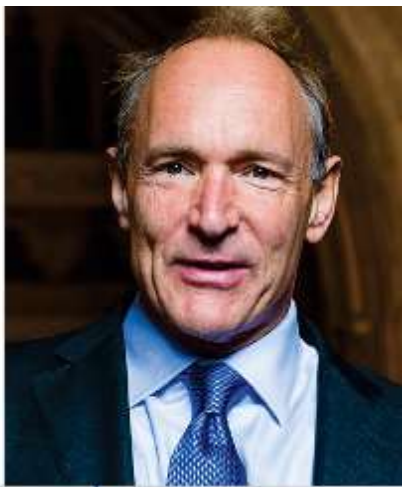
CONQUER BLOCKS

Tim Bernes Lee

Tim Bernes Lee

¿Quién es?

Timothy "Tim" John Berners-Lee, es un científico de la computación británico, conocido por ser el padre de la World Wide Web. Estableció la primera comunicación entre un cliente y un servidor usando el protocolo HTTP en diciembre de 1990. En octubre de 1994 fundó el



Tim Bernes Lee

Curiosidades

- Nació en el 55 (67 años)
- Sus padres se conocieron en el proyecto Mark I (primer ordenador de programa almacenado que se comercializó)
- Graduado en artes...

Tim Bernes Lee

Curiosidades

- En 1989, trabajó en el CERN, el nodo de Internet más grande de Europa y Berners-Lee vio la oportunidad de unir Internet y el hipertexto (HTTP y HTML), de lo que surgiría la World Wide Web.
- La repercusión de su inventiva la podemos notar a día de hoy

Tim Bernes Lee

Curiosidades

- En 1990 construyó el primer navegador (llamado WorldWideWeb y desarrollado con NEXTSTEP) y el primer servidor Web al que llamó httpd (HyperText Transfer Protocol daemon).

HTML-C02

Tim Bernes Lee

Curiosidades

- El primer servidor Web se encontraba en el CERN y fue puesto en línea el 6 de agosto de 1991. Esto proporcionó una explicación sobre lo que era el World Wide Web, cómo uno podría tener un navegador y cómo establecer un servidor Web. Este fue también el primer directorio Web del mundo, ya que Berners-Lee mantuvo una lista de otros sitios Web aparte del suyo.

CONQUERBLOCKS

Tim Bernes Lee

Curiosidades

- Se liberó el código fuente del servidor web así como del cliente/navegador
- Su difusión creció exponencialmente
- El número de servidores Web pasó de veintiséis en 1992 a doscientos en octubre de 1995 lo que refleja cual fue la velocidad de la difusión de internet.

HTML-C02

Tim Bernes Lee

- ¿Os imagináis el mundo sin internet?
- Internet y la web democratizó el acceso a la información
- Los investigadores podían ayudarse los unos a los otros en el campo científico para adelantar tiempo y costes en procesos de creación

CONQUERBLOCKS

Tim Bernes Lee

W3C

- En el 1994 se traslada al MIT y funda lo que hoy conocemos como la W3C
- Una Web única, universal y accesible por cualquier persona y desde cualquier dispositivo
- Quería evitar que Internet se dividiera en áreas que sólo sirvieran a intereses comerciales o académicos.

HTML-C02

Tim Bernes Lee

W3C

El W3C es un organismo internacional de estandarización de tecnologías Web dirigido conjuntamente por el Instituto Tecnológico de Massachusetts, el ERCIM francés y la Universidad de Keiō en Japón.

CONQUERBLOCKS

Tim Bernes Lee

Tim Bernes Lee

En su libro Tejiendo la red, publicado en 1999, Berners-Lee explica por qué la tecnología web es libre y gratis.

Se considera al mismo tiempo el inventor y el protector de la web.

Fue uno de los pioneros a favor de la **neutralidad de red**.

Fue el primer desarrolador web de la historia

Tim Bernes Lee

Inventor de la Web

- **HTML:** HyperText Markup Language.
  El lenguaje de marcado (formato) para la web.
- **URL:** Uniform Resource Locator. Un tipo de "dirección" que es única y que se utiliza para identificar cada recurso en la web.
- **HTTP:** Hypertext Transfer Protocol.
  Permite la recuperación de recursos vinculados de toda la web.

¿Cómo funciona la web?

HTML-C02

¿Cómo funciona la web?

- ARPANET
- IP
- TCP
- ISP
- Router
- Servidor
- DNS
- Dominio

CONQUERBLOCKS

Arpanet

¿Cómo funciona la web?

HTML-C02

ARPANET

- Red de la Agencia de Proyectos de Investigación Avanzada
- Nace como método de interconexión de universidades y edificios estatales para compartir información
- Departamento Defensa EEUU

CONQUERBLOCKS

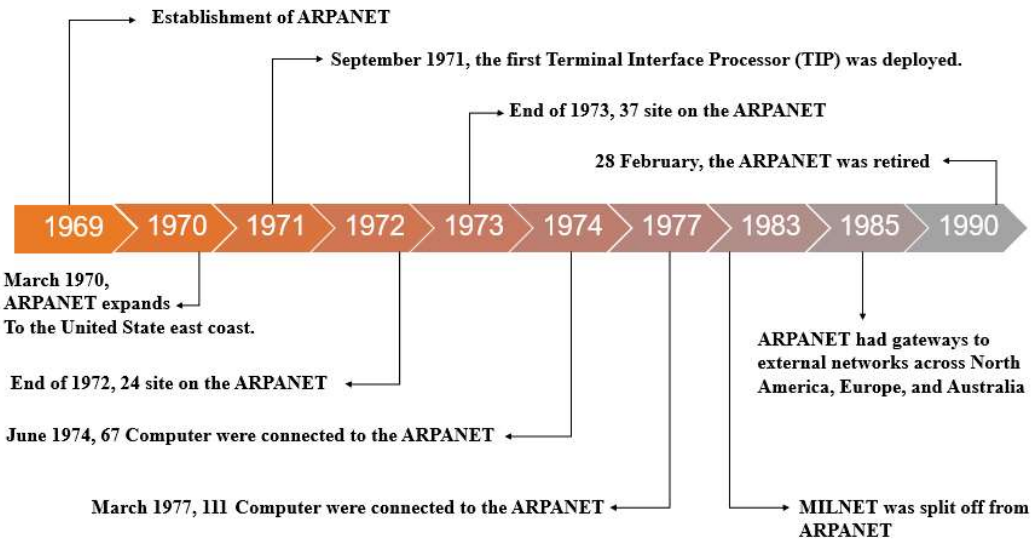
¿Cómo funciona la web?

ARPANET

- Nace con el objetivo de tener una red descentralizada de información
- Con múltiples caminos para llegar a interconectar dos nodos
- Evitar nodo central de comunicación en la guerra nuclear (mito o verdad)

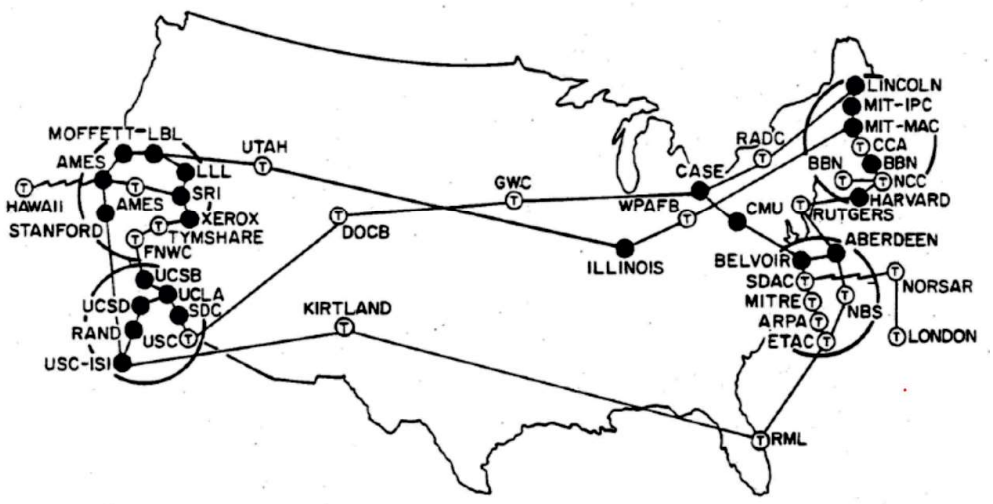
HTML-C02

¿Cómo funciona la web?



CONQUERBLOCKS

¿Cómo funciona la web?

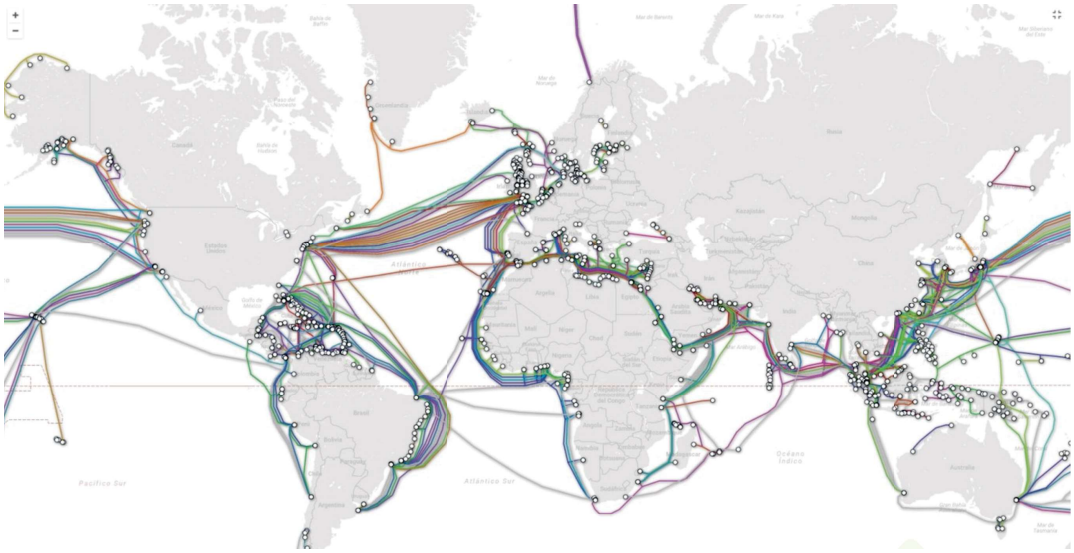


HTML-C02

¿Cómo funciona la web?

CONQUERBLOCKS

¿Cómo funciona la web?



HTML-C02

¿Cómo funciona la web?

- ARPANET
- IP
- TCP
- ISP
- Router
- Servidor
- DNS
- Dominio

CONQUERBLOCKS

¿Cómo funciona la web?

IP

¿Cómo funciona la web?

IP

- Siglas de Internet Protocol
- Forma parte del conjunto de protocolos de internet
- También hace de identificador de un dispositivo en una red

Dirección IP

¿Cómo funciona la web?

HTML-C02

Dirección IP

Dos funciones principales:

- Identificar un dispositivo dentro de una red
- Direccionar un paquete a su ubicación dentro de una red jerárquica

CONQUERBLOCKS

¿Cómo funciona la web?

Dirección IP

- Número en binario de 32 bits
- Se muestra como 4 octetos que van del 0 al 255
- Ejemplos:
  - 10.128.1.253
  - 192.168.1.50

HTML-C02

¿Cómo funciona la web?

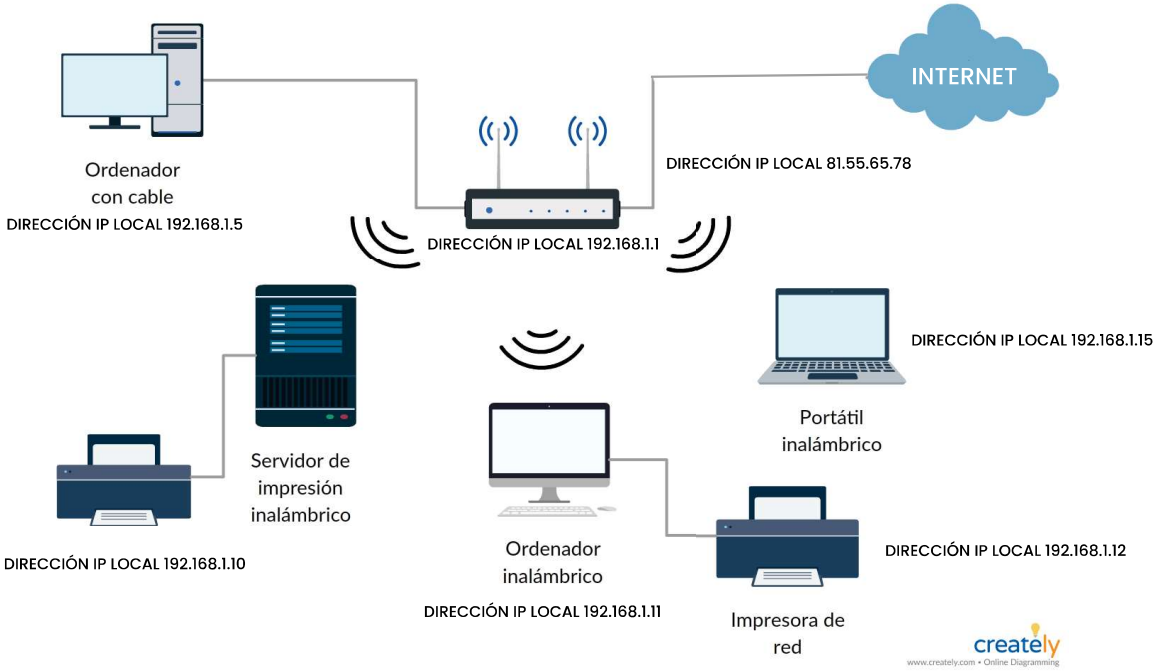
Dirección IP

- El ejemplo que más se parece son los números de teléfono
- Existen 2 grandes tipos de direcciones IP dependiendo del tipo de red en la que estemos
- Direcciones IP públicas y Privadas

CONQUERBLOCKS

HTML-C02

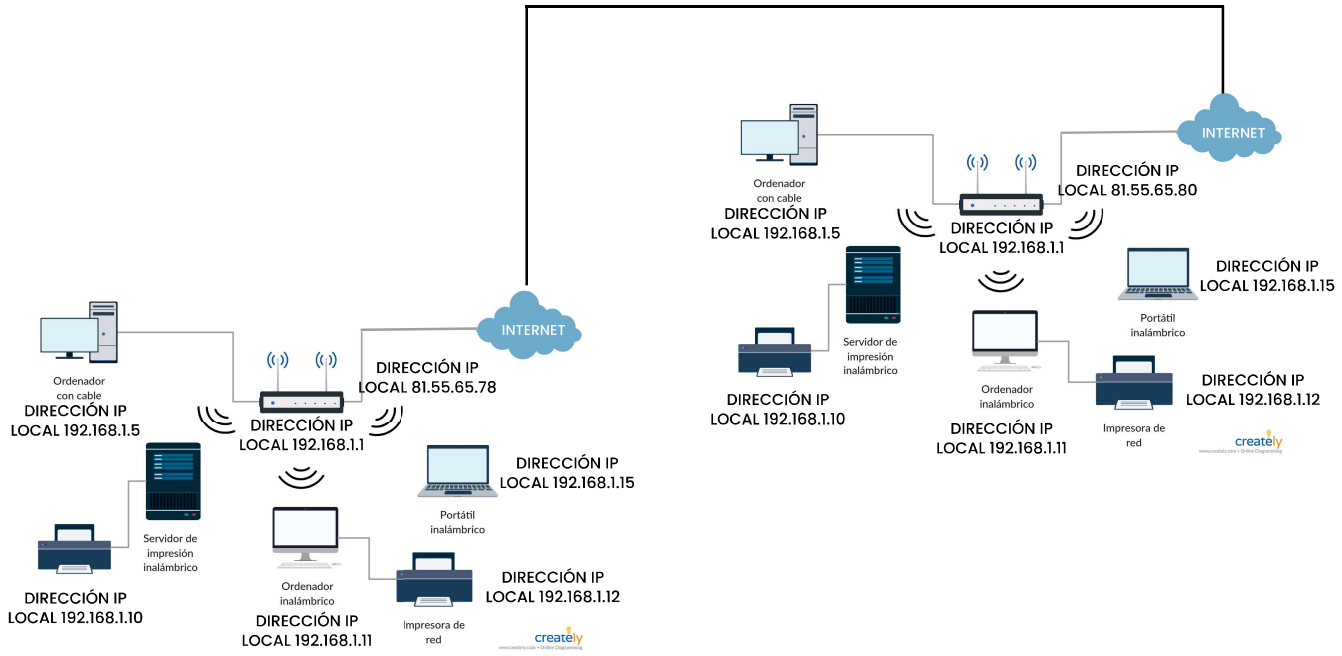
¿Cómo funciona la web?



CONQUERBLOCKS

HTML-C02

¿Cómo funciona la web?



CONQUERBLOCKS

¿Cómo funciona la web?

IPv4 / IPv6

¿Cómo funciona la web?

- ARPANET
- IP
- TCP
- ISP
- Router
- Servidor
- DNS
- Dominio

¿Cómo funciona la web?

ISP

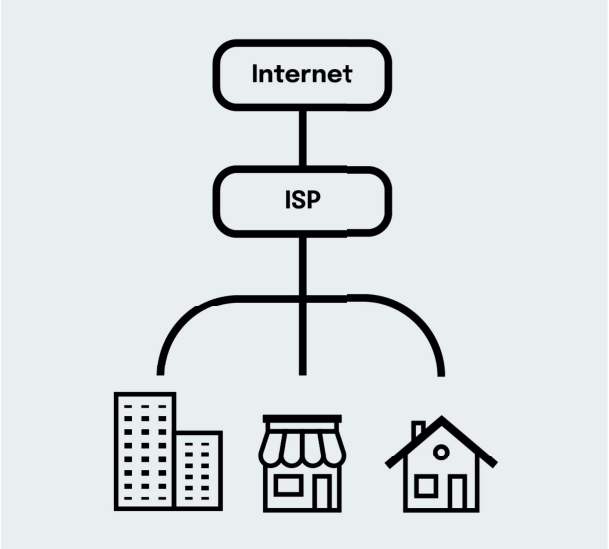
¿Cómo funciona la web?

ISP

- Empresa privada encargada de conectarnos a Internet
- Cuando se abrió ARPANET al público surgió la necesidad de este tipo de empresas
- Cuanto más se democratiza más competitivos son los precios

¿Cómo funciona la web?

ISP



HTML-C02

¿Cómo funciona la web?

- ARPANET
- IP
- TCP
- ISP
- Router
- Servidor
- DNS
- Dominio

CONQUERBLOCKS

Router

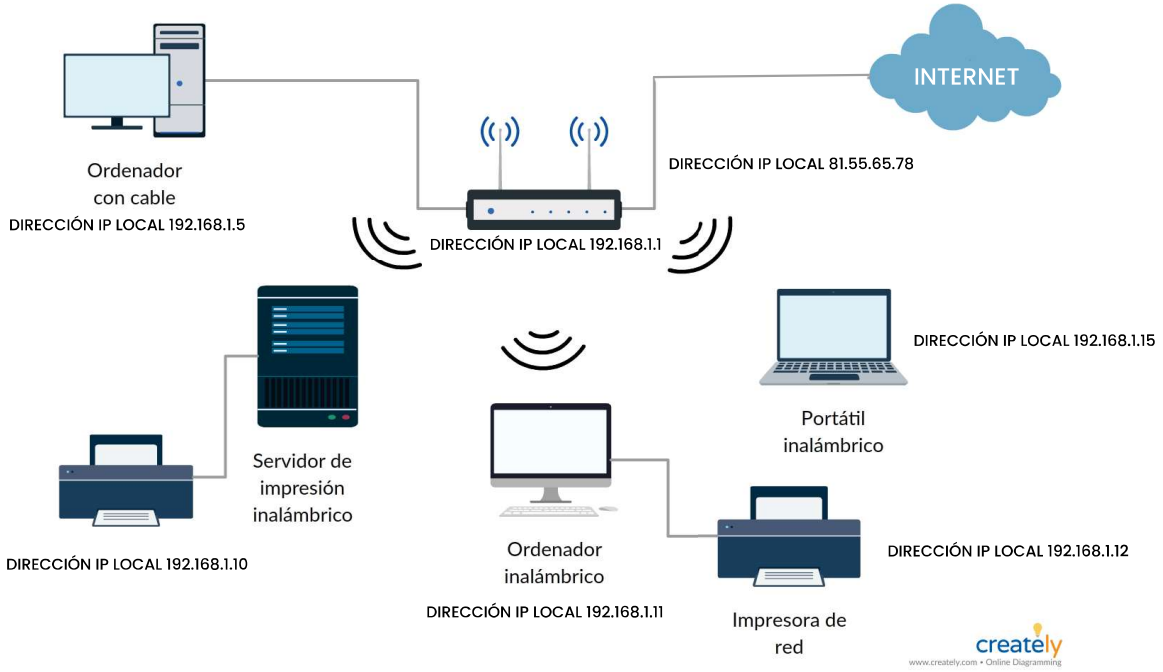
¿Cómo funciona la web?

Router

- Será el encargado de comunicar al menos dos redes
- Encaminará los paquetes enviados por la vía correspondiente
- Pueden existir tanto en red local como a nivel de ISP

HTML-C02
CONQUERBLOCKS

¿Cómo funciona la web?



¿Cómo funciona la web?

- ARPANET
- IP
- TCP
- ISP
- Router
- Servidor
- DNS
- Dominio

Servidor

¿Cómo funciona la web?

HTML-C02

Servidor

- Es un equipo informático que está encendido 24/7 y que está conectado a internet
- Será el responsable de recibir peticiones por parte de clientes o de otros servidores
- Procesará la petición y enviará una respuesta apropiada a dicha petición

CONQUERBLOCKS

¿Cómo funciona la web?

Servidor

- Servidores de correo
- Servidores web
- Servidores de streaming
- etc...

HTML-C02

¿Cómo funciona la web?

Servidor

- Tendrá una identificación de red o una Dirección IP Pública
- También puede únicamente tener una dirección privada si no se necesita acceso externo
- Su principal objetivo es escuchar peticiones y dar respuesta

CONQUERBLOCKS

¿Cómo funciona la web?

Servidor

- Por ejemplo la plataforma de Conquer Blocks
- Vive en un servidor
- Accedemos a los contenidos de dicho servidor vía Web
- A su vez el servidor de Conquer se conecta a Vimeo para que veas este video

¿Cómo funciona la web?

- ARPANET
- IP
- TCP
- ISP
- Router
- Servidor
- DNS
- Dominio

DNS

Domain Name Server

Llama a tu madre ¿Cómo lo harías?

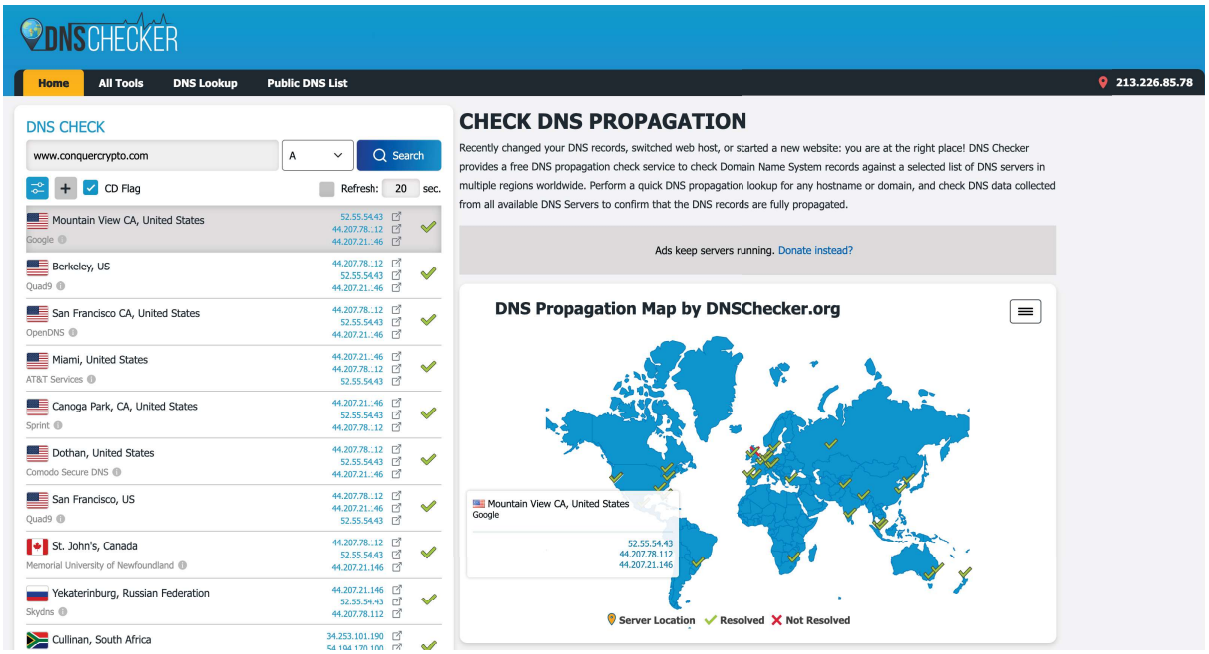
Imagina una agenda global

¿Cómo funciona la web?

DNS

- Traductor de nombres de dominio a direcciones IP a nivel mundial
- Nos olvidamos de aprendernos direcciones IP
- Registro global/mundial replicado

¿Cómo funciona la web?



HTML-C02

Cada cambio puede tardar hasta 48 horas en replicarse

HTML-C02

¿Os acordáis del ataque de DNS que le hicieron a Pancakeswap?

¿Cómo funciona la web?

- ARPANET
- IP
- TCP
- ISP
- Router
- Servidor
- DNS
- Dominio

¿Cómo funciona la web?

Dominio

HTML-C02

CONQUERBLOCKS

¿Cómo funciona la web?

Dominio

- Es un nombre registrado en internet con una extensión determinada
- Existe un registro global de nombres de dominio a nivel mundial
- Cada extensión tiene su propio encargado de controlarlo

HTML-C02

¿Cómo funciona la web?

Dominio

- Por ejemplo .es lo lleva Red.es
- El coste de registro dependerá de la extensión
- Se compran normalmente por un año

CONQUERBLOCKS

¿Cómo funciona la web?

Dominio

- .es .com .edu .net .en .uk. .us .it

HTML-C02

¿Cómo funciona la web?

Dominio

- Cada dominio queda registrado en un registrador de dominios donde podemos establecer los registros DNS que tiene asignados
- Esta será la forma de asignar una dirección IP a un dominio para que sepa a qué servidor debe apuntar

CONQUERBLOCKS

HTML-C02

¿Cómo funciona la web?

¿Ya sabemos qué pasa cuando escribimos un dominio en internet?

CONQUERBLOCKS

HTML-C01

¿Qué son las aplicaciones web?

CONQUERBLOCKS

¿Qué son las aplicaciones web?

HTML-C02

Co mpuestas

- Servidor web
- Lenguaje de parte del servidor
- Sistema gestor de base de datos

¿Qué son las aplicaciones web?

HTML-C02

Ofrecen

- Información mediante páginas web
- Información en algún lenguaje de marcas: json, xml, etc...
- Videos en streaming
- Música en streaming etc...

¿Qué son las aplicaciones web?

¿Cómo accedemos a ellas?

- Mediante un dominio en internet
- Conociendo la dirección ip del servidor que la aloja
- Acceso público
- Acceso privado

HTML-C02

¿Qué son las aplicaciones web?

142.250.185.14

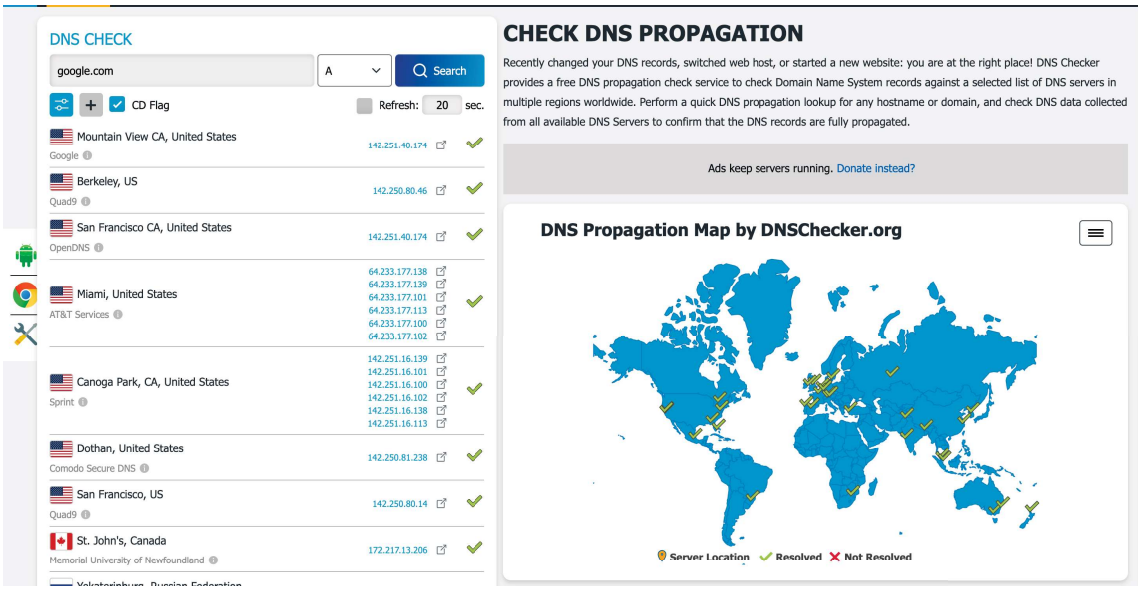
CONQUERBLOCKS

¿Qué son las aplicaciones web?

¿Google?

HTML-C02

¿Qué son las aplicaciones web?



CONQUERBLOCKS

¿Qué son las aplicaciones web?

HTML-C02

CDN: Content Delivery Network

CONQUERBLOCKS

HTML-C01

SGML => XML => HTML

CONQUERBLOCKS

SGML => XML => HTML

HTML-C01

HTML

- Es una mezcla de xml, xhtml, sgml y demás lenguajes de marcas
- Lo estudiaremos con detenimiento
- Es el lenguaje más importante de la web

CONQUERBLOCKS

SGML => XML => HTML

HTML-C01

HTML

- Gracias a él las Web hoy día es lo que es
- Surge con la necesidad de mostrar información mediante internet
- Es el lenguaje más importante de la web

CONQUERBLOCKS

SGML => XML => HTML

HTML

- Siempre debe ser servido mediante un servidor web
- Apache, Nginx, Tomcat...
- Ha hecho que la información se democratice

¿Qué es una URL?

¿Qué es una URL?

Uniform Resource Locator: dirección única y específica que se asigna a cada uno de los recursos disponibles de la World Wide Web para que puedan ser localizados por el navegador y visitados por los usuarios.

¿Qué es una URL?

No confundir con un dominio

¿Qué es una URL?

Cada página, cada recurso, va enlazado a una URL, porque de otra manera los navegadores no podrían localizar la información demandada por el usuario.

¿Qué es una URL?

http://www.example.com:80/path/to/myfile.html?key1=value1&key2=value2#SomewhereInTheDocument

¿Qué es una URL?

http://www.example.com:80/path

Scheme

¿Qué es una URL?

tp://www.example.com:80/path/to/my

Domain Name

¿Qué es una URL?

com:80/path/to/myfile.html?key1=valu

Port

HTML-C01

¿Qué es una URL?

path/to/myfile.html?key1=v

Path to resource

CONQUERBLOCKS

¿Qué es una URL?

?key1=value1&key2=value2

Parameters

¿Qué es una URL?

#SomewhereInTheDocument

Anchor

HTTP

HTML-C01

HTTP

Protocolo de Transferencia de Hipertexto

CONQUERBLOCKS

HTTP

Define y estandariza la forma en la que nos comunicamos con un servidor web

HTTP

Métodos HTTP

GET	→	Leer
POST	→	Crear
PUT	→	Modificar
PATCH	→	Modificar
DELETE	→	Eliminar

<Despedida>

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

<HTML>

Clase 03

CONQUER BLOCKS

<Índice>

Clase 03 Los lenguajes de marcas

Los lenguajes de marcas

Información vs Presentación

¿Qué es una etiqueta?

Atributos comunes y atributos específicos

CONQUER BLOCKS

Los lenguajes de marcas

Los lenguajes de marcas

- Existen desde que comenzó la informática
- Son necesarios para intercambiar información entre sistemas
- Con internet se convirtieron en piedras angulares para la difusión del conocimiento

Los lenguajes de marcas

- Permiten representar información
- Sirven además para estructurar de la información
- Y también para marcar cómo presentar la información
- NO SON LENGUAJES DE PROGRAMACIÓN

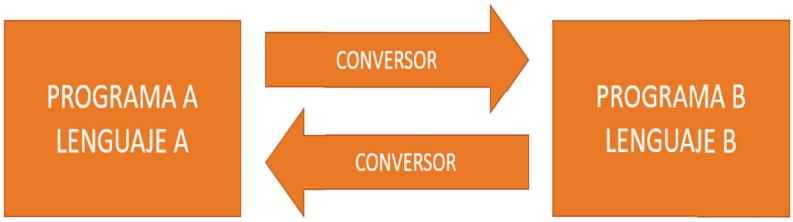
Los lenguajes de marcas

Tipos

- Orientados al almacenamiento y procesado de la información: **HTML**, XML, JSON, DTD...
- Orientados a presentación: **HTML**, CSS

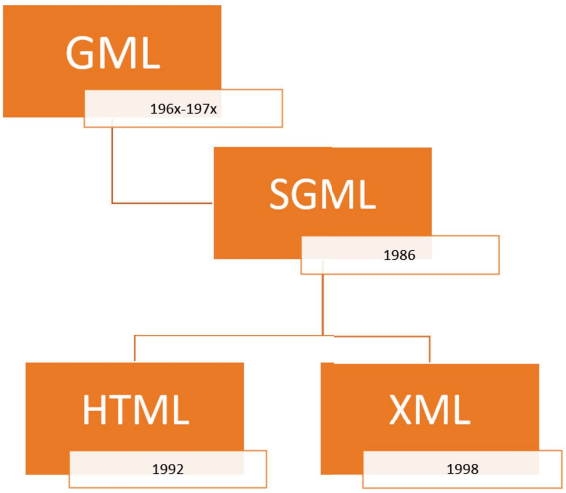
Los lenguajes de marcas

Nenesarios para evitar por cada par de sistemas una conversión



Los lenguajes de marcas

Evolución histórica



Los lenguajes de marcas

Características de los lenguajes de marcas

- Independencia de la infraestructura
- Formados por texto plano
- Flexibles, admiten infinitas formas de estructurar la información
- Compactabilidad > Texto plano
- Facilidad de procesamiento

Los lenguajes de marcas

Ejemplo

- Los elementos delimitados por la marca <#> son títulos de películas y deberán representarse en color azul y utilizando la tipografía Times New Roman con un tamaño de 16 puntos.
- Los elementos delimitados por el símbolo <%> son nombres de director y deberán representarse en color rojo y utilizando la tipografía Times New Roman con un tamaño de 14 puntos

Los lenguajes de marcas

Ejemplo

```
<#>Tron<#>
<%>Steven Lisberger<%>
<#>Juegos de guerra<#>
<%>John Badham<%>
```


Los lenguajes de marcas

Resultado

Tron
Steven Lisberger
Juegos de guerra
John Badham

CONQUERBLOCKS

Los lenguajes de marcas

HTML

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Mi página personal</title>
</head>
<body>
    <h1>Mi lectura favorita</h1>
    <hr>
    <p>Mi libro favorito es...</p>
</body>
</html>
```


Los lenguajes de marcas

XML

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<videojuegos>
    <videojuego multijugador="si">
        <titulo>Call Of Duty</titulo>
        <plataforma>Play Station 5</plataforma>
    </videojuego>
    <videojuego multijugador="si">
         <titulo>Fortnite</titulo>
         <plataforma>XBox One</plataforma>
    </videojuego>
    <videojuego multijugador="no">
        <titulo>Manic Miner</titulo>
        <plataforma>Amstrad CPC</plataforma>
    </videojuego>
</videojuegos>
```

Los lenguajes de marcas

JSON

```
{
     "asunto": "Lenguaje de Marcas",
     "publicacion": "2021",
     "editorial":
     "Paraninfo","lenguajes": [
          {
               "nombre": "HTML",
               uso": "Web"
          },
          {
               "nombre": "XML",
               "uso": "Intercambio de datos"
          },
          {
               "nombre": "JSON",
               "uso": "Intercambio de datos"
          }
     ]
}
```


Información vs Presentación

Te reto

Imagina un lenguaje que codifique el las letras del abecedario

Información vs Presentación

¿Sencillo no?
Sólo necesitamos un código por letra y a escribir

Información vs Presentación

MORSE CODE

A	B	C	D	E	F
·–	–···	–·–·	–··	·	··–·
G	H	I	J	K	L
––·	····	··	·–––	–·–	·–··
M	N	O	P	Q	R
––	–·	–––	·––·	––·–	·–·
S	T	U	V	W	X
···	–	··–	···–	·––	–··–
Y	Z	0	1	2	3
–·––	––··	–––––	·––––	··–––	···––
4	5	6	7	8	9
····–	·····	–····	––···	–––··	––––·

CONQUERBLOCKS

Información vs Presentación

36 elementos
¡A formar palabras!

Información vs Presentación

¿Le ponemos colores?

CONQUERBLOCKS

Información vs Presentación

Hagamos la prueba con 8 colores

Tenemos 36 códigos * 8 colores
288 códigos diferentes

Información vs Presentación

Si quisiera es espectro de los 65535 colores RGB

¿Cuántos códigos necesitamos?

CONQUERBLOCKS

Información vs Presentación

De locos...

Información vs Presentación

Imagínate si además metemos, negrita, cursiva etc...

Crece exponencialmente las variantes

CONQUERBLOCKS

Información vs Presentación

Lenguajes de marcas al rescate

CONQUERBLOCKS

Información vs Presentación

<rojo>hola soy una cadena roja</rojo> y yo una cadena normal. Y yo <negrita>soy una cadena en negrita <rojo>y esto negrita y rojo</rojo></negrita>

CONQUERBLOCKS

Información vs Presentación

No hagáis esto en casa
no mezcléis presentación con
información

HTML como especificación de XML

HTML como especificación de XML

Características

- El lenguaje de marcas más popular
- Es el lenguaje de la web
- Gratuito
- Gestionado por el W3c
- Orientado a las personas
- Su objetivo era estético pero evolucionó a un lenguaje semántico

HTML como especificación de XML

Primera página web de la historia

http://info.cern.ch - home of the first website

From here you can:

- Browse the first website
- Browse the first website using the line-mode browser simulator
- Learn about the birth of the web
- Learn about CERN, the physics laboratory where the web was born

HTML como especificación de XML

HML

- Democratiza la información
- Añade algo nuevo, hipervínculo
- Acceso público
- Acceso privado

HTML como especificación de XML

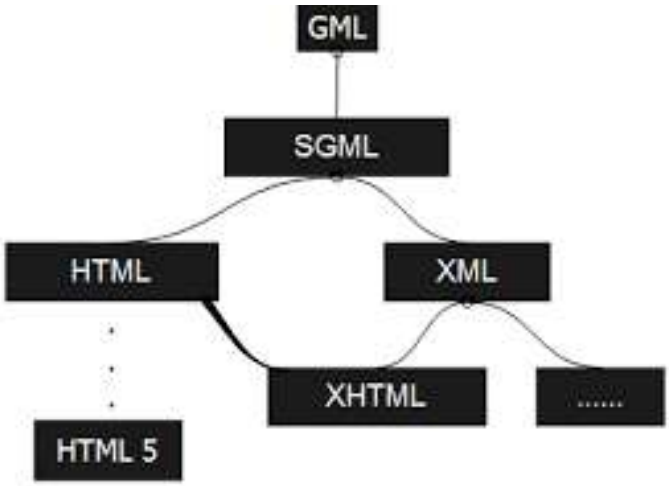
Año	Versión	Observaciones
1991	Diseño inicial	Incluye 18 etiquetas. Se considera como un dialecto de SGML. Se plasma en un documento interno e informal del CERN (Conseil Européen pour la Recherche Nucléaire u Organización Europea para la Investigación Nuclear, en español).
1992	HTML 1.1	Primer borrador informal.
1995	HTML 2.0	Publicado por el IETF (Internet Engineering Task Force o Grupo de Trabajo de Ingeniería de Internet, en español).
1997	HTML 3.2	Publicado como recomendación del W3C (World Wide Web Consortium).
1997	HTML 4.0	Publicado como recomendación del W3C.
1999	HTML 4.1	Publicado como recomendación del W3C.
2000	XHTML	Publicado como recomendación del W3C. Tenía como objetivo sustituir a HTML. En HTML5 se incorpora como parte de la recomendación y deja de desarrollarse por separado.
2014	HTML 5.0	Publicado como recomendación del W3C.
2016	HTML 5.1	Publicado como recomendación del W3C.
2017	HTML 5.2	Publicado como recomendación del W3C.

HTML como especificación de XML

HTML + XML = XHTML

HTML5

HTML como especificación de XML



HTML como especificación de XML

¿Qué es una etiqueta?

¿Qué es una etiqueta?

Etiqueta

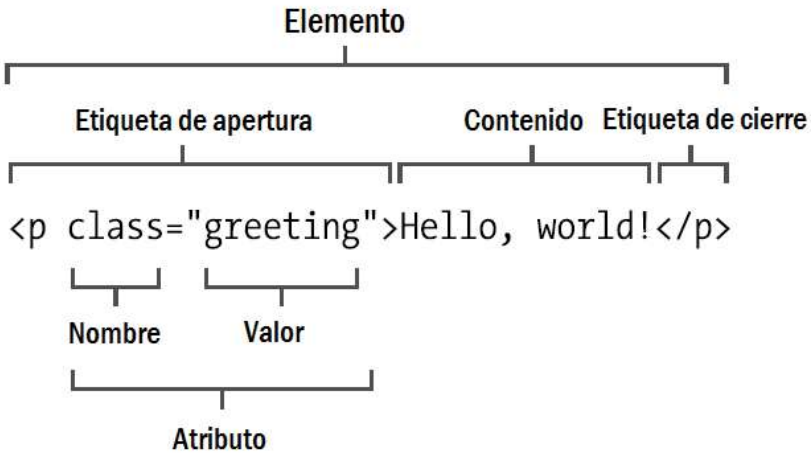
- Elemento más importante en HTML
- Nos servirá para marcar la información
- Compuesta por varios elementos

¿Qué es una etiqueta?

```
<p class="greeting">Hello, world!</p>
```

CONQUERBLOCKS

¿Qué es una etiqueta?



¿Qué es una etiqueta?

Etiquetas vacías

Existen etiquetas vacías cuyo contenido se tiene mediante los atributos

```
<img src="logo.jpg" />

<br />

<hr />
```

¿Qué es una etiqueta?

Contenido de una etiqueta

- **Información en texto**
- **Más etiquetas**
- **Etiquetas vacías**

¿Qué es una etiqueta?

¿Qué es una etiqueta?

Contenido: texto

```
<a href="https://pancakeswap.finance/" target="_blank">PancakeSwap</a>
```


¿Qué es una etique ta?

Contenido: otras etiquetas

```
<p class="principal">
    <a href="https://pancakeswap.finance/" target="_blank">PancakeSwap</a>
</p>
```

¿Qué es una etique ta?

Contenido: otras etiquetas

```
<p class="principal">
    <a href="https://pancakeswap.finance/" target="_blank">PancakeSwap</a>
    <img src="conejo.jpg" alt="Imagen del conejo de PancakeSwal">
</p>
```


¿Qué es una etiqueta?

Cuestiones importantes de XHTML que nos sirven como buenas prácticas en HTML

¿Qué es una etiqueta?

1) Las etiquetas se tienen que cerrar
de acuerdo a como se abren:

Ejemplo correcto en XHTML:

```
<p>Este es un párrafo con <a>un enlace</a></p>
```

Ejemplo incorrecto en XHTML (pero correcto en HTML):

```
<p>Este es un párrafo con <a>un enlace</p></a>
```


¿Qué es una etiqueta?

2) Los nombres de las etiquetas
y atributos siempre se escriben en minúsculas:

Ejemplo correcto en XHTML:

```
<p>Este es un párrafo con <a href="http://
www.google.com">un enlace</a></p>
```

Ejemplo incorrecto en XHTML (pero correcto en HTML):

```
<P>Este es un párrafo con <A HREF="http://
www.google.com">un enlace</A></P>
```

¿Qué es una etiqueta?

3) El valor de los atributos
siempre se encierra con comillas:

Ejemplo correcto en XHTML:

```
<p>Este es un párrafo con <a href="http://
www.google.com">un enlace</a></p>
```

Ejemplo incorrecto en XHTML (pero correcto en HTML):

```
<p>Este es un párrafo con <a href=http://www.google.com>un
enlace</a></p>
```


¿Qué es una etiqueta?

4) Todas las etiquetas deben cerrarse

Ejemplo correcto en XHTML:

```
<img src="logo.jpg" />
```

Ejemplo incorrecto en XHTML (pero correcto en HTML):

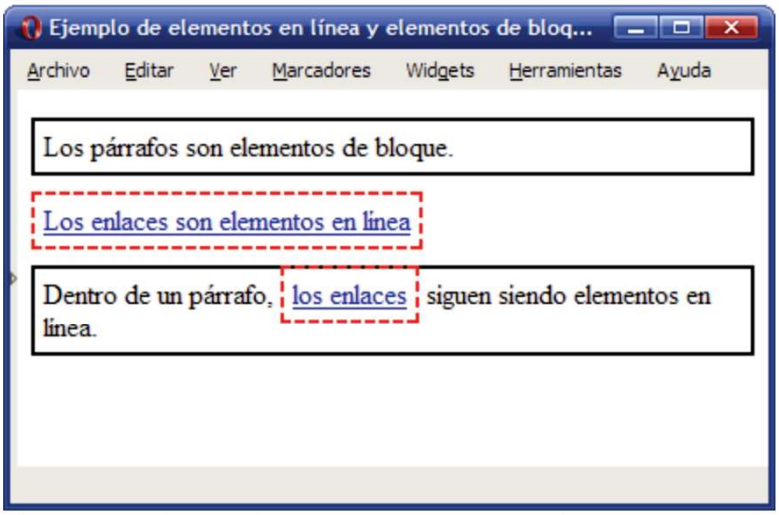
```
<img src="logo.jpg">
```

Tipos de etiqueta

- Etiquetas de línea
- Etiquetas de bloque

Atributos comunes y atributos específicos

Tipos de etiqueta



CONQUERBLOCKS

Atributos comunes y atributos específicos

CONQUERBLOCKS

Atributos

- Proveen de información a la etiqueta
- Una etiqueta puede tener o no tener
- Una etiqueta puede tener tantos atributos como necesite, siempre que sean válidos

CONQUERBLOCKS

Atributos comunes y atributos específicos

Tipos

- Atributos comunes
- Atributos específicos de la etiqueta
- Atributos obligatorios y opcionales

Atributos comunes y atributos específicos

Algunos atributos comunes

Atributo	Descripción
class	Permite agrupar elementos. El valor de este atributo es una cadena de caracteres definida por el autor de la página. Si varios elementos comparten el mismo valor, se dice que son de la misma clase. Se utiliza para asignar estilos y durante la programación con JavaScript.
contenteditable	Si existe, el contenido del elemento es editable.
hidden	Si existe, el elemento no se representa en el navegador.
id	Identifica al elemento de manera única en el documento. El valor es una cadena de caracteres definida por el autor de la página. Se utiliza para asignar estilos y durante la programación con JavaScript.
lang	Determina el idioma en el que está escrito el contenido del elemento.
spellcheck	Determina si el elemento debe ser analizado ortográfica y gramaticalmente por el navegador.
style	Permite asignar un estilo al componente.
tabindex	Determina el orden de selección de elementos cuando se pulsa el tabulador.
title	Permite especificar información extra al elemento.
translate	Determina si el elemento debe ser traducido o no.

CONQUERBLOCKS

Atributos comunes

- Atributos básicos
- Atributos de internacionalización
- Atributos de eventos
- Atributos de foco

Atributos básicos

Atributo	Descripción
`id = "texto"`	Establece un identificador único a cada elemento dentro de una página HTML
`class = "texto"`	Establece la clase CSS que se aplica a los estilos del elemento
`style = "texto"`	Establece de forma directa los estilos CSS de un elemento
`title = "texto"`	Establece el título a un elemento (mejora la accesibilidad y los navegadores lo muestran cuando el usuario pasa el ratón por encima del elemento)

Atributos internacionalización

Atributo	Descripción
`lang = "codigo de idioma"`	Indica el idioma del elemento mediante un código predefinido
`xml:lang = "codigo de idioma"`	Indica el idioma del elemento mediante un código predefinido
`dir`	Indica la dirección del texto (útil para los idiomas que escriben de derecha a izquierda)

Atributos de eventos

Atributo	Descripción	Elementos que pueden usarlo
`onblur`	Deseleccionar el elemento	`<button>`, `<input>`, `<label>`, `<select>`, `<textarea>`, `<body>`
`onchange`	Deseleccionar un elemento que se ha modificado	`<input>`, `<select>`, `<textarea>`
`onclick`	Pinchar y soltar el ratón	Todos los elementos
`ondblclick`	Pinchar dos veces seguidas con el ratón	Todos los elementos
`onfocus`	Seleccionar un elemento	`<button>`, `<input>`, `<label>`, `<select>`, `<textarea>`, `<body>`
`onkeydown`	Pulsar una tecla (sin soltar)	Elementos de formulario y `<body>`
`onkeypress`	Pulsar una tecla	Elementos de formulario y `<body>`
`onkeyup`	Soltar una tecla pulsada	Elementos de formulario y `<body>`

Atributos de foco

Atributo	Descripción
`accesskey = "letra"`	Establece una tecla de acceso rápido a un elemento HTML
`tabindex = "numero"`	Establece la posición del elemento en el orden de tabulación de la página. Su valor debe estar comprendido entre `0` y `32.767`
`onfocus, onblur`	Controlan los eventos JavaScript que se ejecutan cuando el elemento obtiene o pierde el foco

Atributos vacíos

¿Qué es una etiqueta?

Atributos vacíos

```
<form action="server.php">
    <input type="text name="nombre" required>
    <button>Enviar</button>
</form>
```

CONQUERBLOCKS

‹Despedida›

Email

bienvenidosaez@gmail.com

Instagram

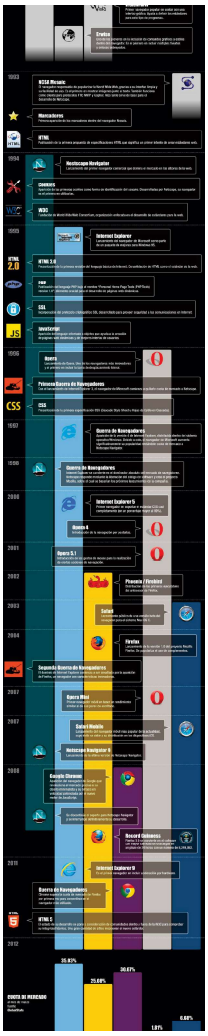
@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

HTML como especificación de XML


Viola
Primer navegador capaz de crear sitios web. Viola también incluía el primer lenguaje de hojas de estilos para que los no programas.
Erwise
Navegador Internet con la novedosa incorporación de elementos gráficos y además contaba con navegación. Fue el pionero en incluir marcadores y formularios HTML.
1993
NCSA Mosaic
El navegador responsable de popularizar la World Wide Web, gracias a su interfaz intuitiva y fácil de usar. Mosaic fue el primer navegador que mostraba imágenes en línea junto con otros elementos gráficos. Fue también el navegador que sirvió de base para el desarrollo de Netscape.
Marcadores
Permitía guardar los marcadores dentro del navegador Mosaic.
HTML
Publicación de la primera propuesta de especificaciones HTML que significa un primer intento de estandarizar este elemento web.
1994
Netscape Navigator
Lanzamiento del primer navegador comercial que dominó en Internet en los últimos lustros web.
CGI/BINS
Mecanismo de interfaz archivos como forma de identificación del usuario. Desarrollado por Netscape, su navegador es el pionero en utilizarlo.
W3C
Fundación de World Wide Web Consortium, organización encargada el desarrollo de estándares para la web.
1995
Internet Explorer
Lanzamiento del navegador de Microsoft como parte de un paquete de archivos del Windows 95.
HTML 2.0
HTML 3.0
Presentación de la primera versión del lenguaje basada en Internet. Se modificaron las HTML para el estándar de la web.
PDF
Lanzamiento del lenguaje PDF (siglas de nombre Personal Device Page Tools) (PDF/Tools) versión 1.0 diseñado en un principio para el desarrollo de páginas web. Adobe.
SSL
Aceleración de protocolo criptográfico SSL desarrollado para proveer seguridad y los intercambiamos en Internet.
JS
JavaScript
Lenguaje de programación orientado a objetos que ejecuta a ejecutar de programas web dinámicos y así interactuamos de usuarios.
1996
Opera
Lanzamiento de Opera. Una de los navegadores, más demandados y el pionero en incluir la opción de navegación mediante tabs.
Primera Guerra de Navegadores
Lucha de navegadores de Internet Explorer 3.0 y navegador de Microsoft contienen su plan contra el navegador de Netscape.
CSS
Presentación de la primera versión de CSS (Cascade Style Sheets, Hojas de Estilos en Cascada).
1997
Guerra de Navegadores
Guerra entre el navegador 4.0 de Netscape y Internet Explorer 4.0 de Microsoft. Ambos navegadores incluyen soporte de estándares web, como HTML, CSS y DOM. Aparecen también nuevas características propietarias, incompatibles entre ellas, como el elemento <blink> de Netscape y el elemento <marquee> de Microsoft.
1998
Guerra de Navegadores
Lanzamiento de la versión de la Netscape Navigator 4.0 que incluye el navegador Mozilla, cuyo fin es la creación de un navegador de código abierto. Mozilla, cuyo fin es la creación de un navegador de código abierto. Mozilla, cuyo fin es la creación de un navegador de código abierto.

<HTML>

Clase 04 - Tema 3

CONQUER BLOCKS

<Índice>

Tema 3: Estructura de una página web

Esqueleto de cualquier página web

Etiquetas de cabecera y metadados

Etiquetas de contenido

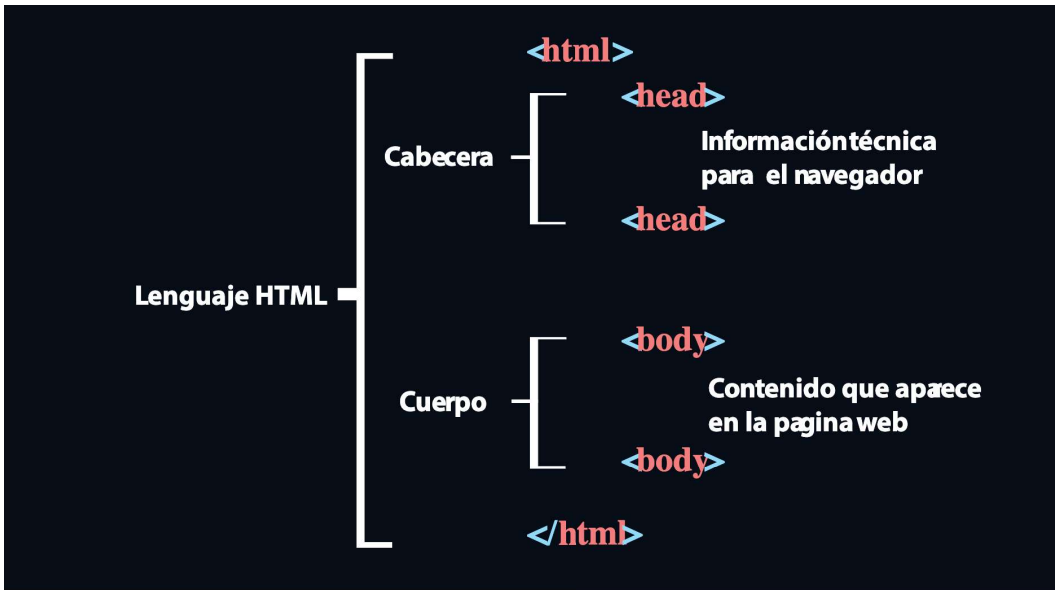
Comentarios

HTML5 Boilerplate

CONQUER BLOCKS

Esqueleto de cualquier página web

Esqueleto de cualquier página web



Esqueleto de cualquier página web

html

<html> </html>

Este elemento engloba todo el contenido de la página y es conocido en ocasiones como el elemento raíz.

head

<head> </head>

Este elemento actúa como contenedor para todos los parámetros que quieras incluir en el documento HTML que no será visible a los visitantes de la página. Incluye cosas como palabras clave y la descripción de la página que quieras mostrar en los resultados de búsqueda, así como la hoja de estilo para formatear nuestro contenido, declaraciones de codificación de caracteres y más.

body

<body> </body>

Contiene todo el contenido que quieres mostrar a los usuarios cuando visitan tu página, ya sea texto, imágenes, vídeos, juegos, pistas de audio reproducibles o cualquier otra cosa.

Esqueleto de cualquier página web

```html
<!DOCTYPE html>

<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Mi primera pagina web</title>
</head>
<body>

    <!--Aqui va el contenido-->

</body>
</html>
```

Esqueleto de cualquier página web

1 <!DOCTYPE html>

En su primera línea, es siempre conveniente indicar el tipo de documento a utilizar. Para indicar que se trata de un documento HTML5.

3 <html lang="en"> </html>

En su tercera línea, en estas etiquetas debemos especificar la lengua en la que está redactado el contenido del documento, en nuestro caso se trata de "en" ingles segun aparece en el ejemplo.

6 <meta charset="UTF-8">

En su sexta línea, este elemento establece que tu documento HTML usará la codificación UTF-8, que incluye la gran mayoría de caracteres de todos los idiomas humanos escritos. En resumen: puede gestionar cualquier contenido textual que pongas en tu documento. No hay razón para no configurarlo y puede te ayudar a evitar problemas más adelante.

7 <meta name="viewport" content="width=device-width, initial-scale=1.0">

En su septima línea, Un elemento de ventana `<meta>` le da al navegador instrucciones sobre cómo controlar las dimensiones y la escala de la página. Si estás diseñando flexiblemente, usa la etiqueta meta viewport te prepara para adaptar tu pagina a multi-dispositivo.

Esqueleto de cualquier página web

Primeras etiquetas importantes

8 <title> </title>

En su octava línea, este elemento establece el título de la página, que aparece en la pestaña/ventana de tu navegador cuando la página se carga y se utiliza para describir la página cuando la añades a tus marcadores o la marcas como favorita.

10 <body> </body>

En su decima línea, en esta zona se incluyen todas las etiquetas HTML que si tienen repercusión visual en el navegador del usuario, generalmente utilizadas para dibujar secciones o partes de una página.

Esqueleto de cualquier página web

Vamos a desmenuzar esto un poco más

Etiquetas de cabecera y metadatos

Etiquetas de cabecera y metadatos

```
<!DOCTYPE html>
```


Etiquetas de cabecera y metadatos

doctype

- Define el tipo de documento
- Debe ir en la primera línea del documento
- Es obligatoria, pero si no se indica se visualizará correctamente
- Si no se establece el navegador entra en modo no estándar y posiblemente no cargará algunas de las APIS de html5

Etiquetas de cabecera y metadatos

```
<head></head>
```


Etiquetas de cabecera y metadatos

head

- Información relevante para el navegador
- No confundir con la cabecera de un documento web a nivel visual
- Es más relevante de lo que parece

CONQUERBLOCKS

Ejemplos de etiquetas que van en el head

Etiquetas de cabecera y metadatos

```html
<!DOCTYPE html>
<html lang="es">

<head>
    <!— Título para el navegador y los buscadores —>
    <title>Mi primera página web</title>
    <!— Descripción de la página para los buscadores —>
    <meta name="description" content="Descripción corta de nuestra web para Google">
    <!— Palabras clave para los buscadores —>
    <meta name="keywords" content="palabra1, palabra2, palabra3">
    <!— Autor de la página —>
    <meta name="author" content="Nombre del autor">
    <!— Favicon de la página —>
    <link rel="icon" href="favicon.ico" type="image/x-icon">
    <!— Codificación—>
    <meta charset="UTF-8">
    <!— Para que la web sea responsive —>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!— Para algunos navegadores móviles —>
    <meta name="theme-color" content="#1e7bbf">
</head>

<body>
    <h1>Mi primera web</h1>
    <p>Este es el contenido de mi primera página web en Conquer</p>
</body>

</html>
```

CONQUERBLOCKS

Etiquetas de cabecera y metadatos

Revisemos el head

Etiquetas de cabecera y metadatos

```html
<head>
    <!— Título para el navegador y los buscadores —>
    <title>Mi primera página web</title>
    <!— Descripción de la página para los buscadores —>
    <meta name="description" content="Descripción corta de nuestra web para Google">
    <!— Palabras clave para los buscadores —>
    <meta name="keywords" content="palabra1, palabra2, palabra3">
    <!— Autor de la página —>
    <meta name="author" content="Nombre del autor">
    <!— Favicon de la página —>
    <link rel="icon" href="favicon.ico" type="image/x-icon">
    <!— Codificación—>
    <meta charset="UTF-8">
    <!— Para que la web sea responsive —>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!— Para algunos navegadores móviles —>
    <meta name="theme-color" content="#1e7bbf">
</head>
```

Etiquetas de cabecera y metadatos

title

Define el título que aparecerá en la pestaña del navegador

Su función más importante es que será el texto por lo que será indexada nuestra página en todos los índices (Google, Bing...)

Es muy importante que cada página disponga de su title personalizado

```
<!— Título para el navegador y los buscadores —>
<title>Mi primera página web</title>
```


Etiquetas de cabecera y metadatos

description

Define el descripción general de esa página en concreto

También aparecerá en los resultados de búsqueda donde se muestre nuestra página

```
<!— Descripción de la página para los buscadores —>
<meta name="description" content="Descripción corta de nuestra web para Google">
```

Etiquetas de cabecera y metadatos

Etiquetas de cabecera y metadatos

Esto genera lo anterior

```
<meta charset="utf-8">
<title>▷ Máster en Desarrollo Blockchain (100% Online)</title>
<meta content="La programación blockchain es un trabajo con alta demanda y altos salarios ✅ Descubre nuestra formación y maximiza tu valor com
o profesional." name="description">
<meta content="▷ Máster en Desarrollo Blockchain (100% Online)" property="og:title">
```

Etiquetas de cabecera y metadatos

¿Cómo se verá nuestro ejemplo de nuestra primera página web?

Etiquetas de cabecera y metadatos

charset

Define la tabla de codificación
que debe usarse para mostrar el contenido

Si en alguna ocasión has visto textos como cÃ³digo o tambiÃ©n es porque probablemente tenga una codificación incorrecta o no se le haya establecido esta etiqueta

¿Qué son las tablas de codificación?

```
<!— Codificación —>
<meta charset="UTF-8">
```

Etiquetas de cabecera y metadatos

keywords

Anteriormente se usaba como las palabras que usaba Google para indexar nuestra página en su índice de resultados

Su abuso ha hecho que:
Hoy en día no sirvan para nada

Sólo buscadores antiguos usarán esta meta

```
<!— Palabras clave para los buscadores —>
<meta name="keywords" content="palabra1, palabra2, palabra3">
```


Etiquetas de cabecera y metadatos

author

Esta etiqueta se utiliza para indicar quien es el author de dicho documento. No tiene mucho más uso

```
<!— Autor de la página —>
<meta name="author" content="Nombre del autor">
```

Etiquetas de cabecera y metadatos

favicon

Es la etiqueta donde indicaremos el favicon de nuestra página.

Es posible tener varias versiones dependiendo de las diferentes plataforma

Lo veremos con detenimiento en un tema aparte

```
<!— Favicon de la página —>
<link rel="icon" href="favicon.ico" type="image/x-icon">
```


Etiquetas de cabecera y metadatos

viewport

Es la etiqueta donde indicaremos el favicon de nuestra página.

Es posible tener varias versiones dependiendo de las diferentes plataforma

Lo veremos con detenimiento en un tema aparte

```
<!— Para que la web sea responsive —>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Etiquetas de cabecera y metadatos

theme-color

Se usará pra algunos navegadores móviles que disponen de la funcionalidad de poner la cabecera del navegador de un color determinado.

Ayudará a mejorar el branding de nuestras webs en mobile

```
<!— Para algunos navegadores móviles —>
<meta name="theme-color" content="#1e7bbf">
```


Etiquetas de cabecera y metadatos

Esto es solo el comienzo, dentro del head tendremos muchas más etiquetas que iremos viendo conforme avance el curso

CONQUERBLOCKS

Etiquetas de contenido

Etiquetas de contenido

“Los hijos de body”

Etiquetas de contenido

Etiquetas de contenido

- Son las etiquetas que va a renderizar nuestro navegador (elementos visuales)
- Definirán todo el contenido visual de nuestras páginas

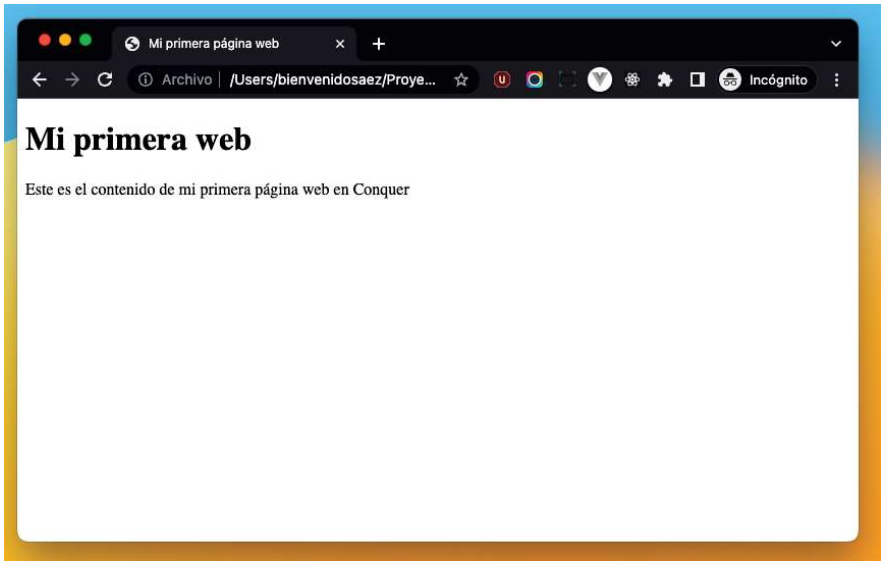
Etiquetas de contenido

Etiquetas de contenido

```html
<!DOCTYPE html>
<html lang="es">

<head>
    <!— Título para el navegador y los buscadores —>
    <title>Mi primera página web</title>
    <!— Descripción de la página para los buscadores —>
    <meta name="description" content="Descripción corta de nuestra web para Google">
    <!— Palabras clave para los buscadores —>
    <meta name="keywords" content="palabra1, palabra2, palabra3">
    <!— Autor de la página —>
    <meta name="author" content="Nombre del autor">
    <!— Favicon de la página —>
    <link rel="icon" href="favicon.ico" type="image/x-icon">
    <!— Codificación—>
    <meta charset="UTF-8">
    <!— Para que la web sea responsive —>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!— Para algunos navegadores móviles —>
    <meta name="theme-color" content="#1e7bbf">
</head>

<body>
    <h1>Mi primera web</h1>
    <p>Este es el contenido de mi primera página web en Concuer</p>
</body>

</html>
```

Etiquetas de contenido

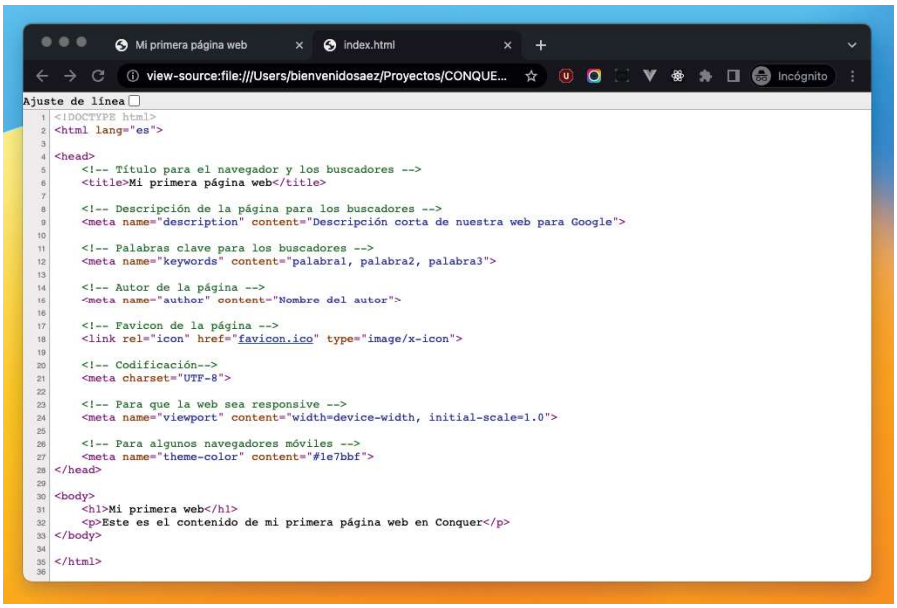
Así se verá



CONQUERBLOCKS

Etiquetas de contenido

Ver código fuente



CONQUERBLOCKS

Tipos de etiquetas de contenido

Etiquetas de bloque

Etiquetas de contenido

Los elementos de bloque definidos por HTML son:
address, blockquote, center, dir, div, dl, fieldset, form, h1, h2, h3, h4, h5, h6, hr, isindex, menu, noframes, noscript, ol, p, pre, tab le, ul, dd, dt, frameset, li, tbody, td, tfoot, th, thead, tr.

Etiquetas de contenido

Etiquetas de línea

Etiquetas de contenido

Los elementos en línea definidos por HTML son:
a, abbr, acronym, b, basefont, bdo, big, br, cite, code, dfn, em, font, i, img, input, kbd, label, q, s, samp, select, small, span, strike, strong, sub, sup, textarea, tt, u, var.

Etiquetas de contenido

Modifiquemos nuestro contenido

```
<body>
    <h1>Mi primera web</h1>
    <p>Esta es primera página web en <a href="https://www.conquercrypto.com/">Conquer</a></p>
</body>
```

En este ejemplo hemos usado 2 etiquetas de bloque y una etiqueta de línea.
También hemos hecho uso del atributo href

Etiquetas de contenido

Luego os voy a presentar a vuestro mejor amigo de aquí en adelante en el desarrollo web

INSPECTOR DE ELEMENTOS

Comentarios

Comentarios

Comentarios

- Elemento común en todo lenguaje de programación y de marcas
- Algunos lenguajes de marcas no lo soportan
- Mira por el tú del futuro

Comentarios

```
<!-- Hola soy un comentario -->
```

Comentarios

Ejemplos de comentarios

```html
<!DOCTYPE html>
<html lang="es">

<head>
    <!— Título para el navegador y los buscadores —>
    <title>Mi primera página web</title>
    <!— Descripción de la página para los buscadores —>
    <meta name="description" content="Descripción corta de nuestra web para Google">
    <!— Palabras clave para los buscadores —>
    <meta name="keywords" content="palabra1, palabra2, palabra3">
    <!— Autor de la página —>
    <meta name="author" content="Nombre del autor">
    <!— Favicon de la página —>
    <link rel="icon" href="favicon.ico" type="image/x-icon">
    <!— Codificación—>
    <meta charset="UTF-8">
    <!— Para que la web sea responsive —>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!— Para algunos navegadores móviles —>
    <meta name="theme-color" content="#1e7bbf">
</head>

<body>
    <h1>Mi primera web</h1>
    <p>Este es el contenido de mi primera página web en Concuer</p>
</body>

</html>
```


Comentarios

Cuidado con lo que escribes lo podrá ver todo el mundo

HTML5 Boilerplate

HTML5 Boilerplate

- Necesitamos una estructura base para comenzar
- No queremos tener que escribir siempre la base de nuestros documentos
- Esto luego lo realizaremos con nuestro editor VSCode
- Están revisados por miles de desarrolladores

HTML5 Boilerplate

Varias posibilidades

- HTML5 Boilerplate
- htmlboilerplates.com
- Solucionan el mismo problema

HTML5 Boilerplate

Podemos aprender mucho de HTML5 Boilerplate

HTML5 Boilerplate

Aprendizajes

- Estructura de carpetas
- Forma de nombrar a los ficheros
- Forma de organizar los assets:
  - img
  - css
  - js

Demo

Demo

DEMO

- ¿Cómo abrir nuestro primer fichero html?
- Mostrar código fuente
- Abrir inspector de elementos
- Comparar etiquetas de bloque con etiquetas de línea

Demo

DEMO

- Importancia del meta charset
- Title
- Mostrar comentarios

Tareas

Tareas

Tarea 4.1

Visualizando la clase y viendo un poco la estructura de un primer documento html, crear vuestro primer documento html haciendo mucho hincapié en la estructura del mismo. El contenido es lo de menos, con un título y un párrafo será más que suficiente. Eso si, el contenido deberá ir acorde a lo que queremos mostrar, es decir, title, description, h1 y párrafo deben tener un significado.

Tareas

Tarea 4.2

Realizar la misma tarea de antes pero en lugar de hacer un único fichero html, hacedla utilizando la plantilla boilerplate html5 que os he dejado en el sistema de archivos

CONQUERBLOCKS

‹Despedida›

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

<HTML>

Clase 05 - Tema 4

CONQUER BLOCKS

<Índice>

Tema 4: Texto en HTML

Etiquetas Hx

p, blockquote, cite, code, strong y em

Etiquetas no recomendadas

Entidades en HTML

Enlaces

Espacios en blanco en HTML

CONQUER BLOCKS

Etiquetas hx

CONQUERBLOCKS

Etiquetas hx

‹hx›

- Sustituimos la x por 1,2,3,4,5 y 6
- Se utilizan para establecer cuáles son los títulos de nuestros contenidos
- Al igual que en los libros, podemos dividirlos en capítulos, subcapítulos, secciones etc...

CONQUERBLOCKS

Etiquetas hx

<hx>

- <h1> Nivel 1. Equivalente al título del documento.
- <h2> Nivel 2. Equivalente al tema del documento.
- <h3> Nivel 3. Equivalente a la sección de un tema.
- <h4> Nivel 4. Equivalente a un apartado de la sección.
- <h5> Nivel 5. Equivalente a un ejemplo del apartado.
- <h6> Nivel 6. Equivalente a un subapartado del ejemplo.

Etiquetas hx

<h1>

- Es el texto más importante de cada uno de nuestros documentos
- Solo debe haber una etiqueta h1 por cada documento
- Debe indicar el tema principal de lo que va nuestro documento
- SEO

Etiquetas hx

‹hx›

- Del resto de etiquetas h no existe una limitación en cuanto al número que puede hacer en un mismo documento
- Si debemos tener en cuenta: orden de aparición ya que no tiene sentido que desde un h2 saltemos a un h4 sin haber de por medio una etiqueta h3

Etiquetas hx

‹hx›

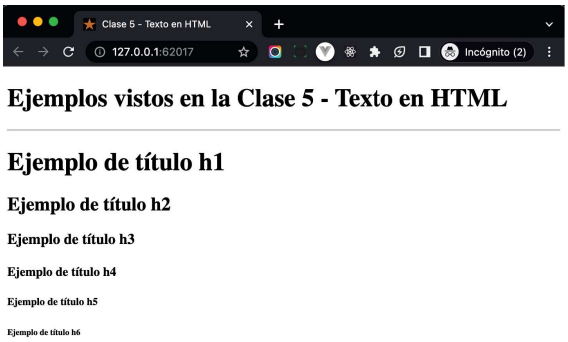
- Presentación por defecto: tamaño, márgenes y grosor
- Jerarquía de importancia
- Son **etiquetas de bloque**
- **No deben contener dentro otras etiquetas de bloque**

Etiquetas hx

Demo <hx>

Etiquetas hx

```
<body>

    <h1>Ejemplos vistos en la Clase 5 - Texto en HTML</h1>
    <hr />

    <h1>Ejemplo de título h1</h1>
    <h2>Ejemplo de título h2</h2>
    <h3>Ejemplo de título h3</h3>
    <h4>Ejemplo de título h4</h4>
    <h5>Ejemplo de título h5</h5>
    <h6>Ejemplo de título h6</h6>

</body>
```



Ejemplo

p, blockquote, cite, code, em y strong

CONQUERBLOCKS

p, blockquote, cite, code, em y strong

```
<p>Hola soy un párrafo</p>
```

CONQUERBLOCKS

p, blockquote, cite, code, em y strong

<p>

- Se utiliza para crear párrafos de texto
- Es una etiqueta de bloque
- Atributos comunes
- Atributos de eventos

p, blockquote, cite, code, em y strong

<p>

- Visualización por defecto: separación entre ellos por arriba y por abajo
- Control del interlineado
- Empieza y termina siempre con línea nueva
- Puede contener texto u otras etiquetas
- No debería contener otras etiquetas de bloque

Demo <p>

Especificación oficial

<blockquote>

p, blockquote, cite, code, em y strong

<blockquote>

- Se utiliza para indicar que el texto pertenece a otro sitio externo del cual se indica la fuente
- Es una etiqueta de bloque
- Atributos comunes y eventos
- Atributo específico cite de tipo url para indicar el origen de la cita

p, blockquote, cite, code, em y strong

<blockquote>

- No genera automáticamente comillas alrededor del contenido. Se deben añadir de forma explícita en el propio contenido o mediante contenido generado a través de CSS.
- En versiones anteriores de HTML, el contenido de <blockquote> debía escribirse dentro de párrafos <p>. En HTML5 este requisito ha sido eliminado, pero es conveniente respetarlo.
- El atributo cite no sirve realmente para nada

Demo ‹blockquote›

Especificación oficial

p, blockquote, cite, code, em y strong

<cite>

CONQUERBLOCKS

p, blockquote, cite, code, em y strong

<cite>

- cite es uno de los pocos identificadores de HTML que se emplea tanto como etiqueta como atributo
- Es una etiqueta de línea, se renderiza en cursiva
- Atributos comunes y eventos
- Como etiqueta, <cite> se emplea para referenciar un trabajo creativo como un libro, una película, un programa de televisión o una canción

p, blockquote, cite, code, em y strong

Demo <cite>

Especificación oficial

CONQUERBLOCKS

p, blockquote, cite, code, em y strong

<code>

p, blockquote, cite, code, em y strong

<code>

- Se utiliza para indicar al navegador que el contenido de dicha etiqueta es un bloque de código fuente de alguno de los lenguajes de programación o marcado
- Es una etiqueta de línea, se renderiza con la fuente predeterminada monospace que tenga el navegador
- Atributos comunes y eventos
- Lo normal será darle estilos css asociados al tipo de lenguajes

p, blockquote, cite, code, em y strong

Demo <code>

Especificación oficial

p, blockquote, cite, code, em y strong

CONQUERBLOCKS

p, blockquote, cite, code, em y strong

strong

- Se utiliza para marcar texto como importante dentro de un párrafo o bloque de contenido
- Es una etiqueta de línea
- Normalmente el navegador renderiza el contenido en negrita
- Además de la parte visual, strong tiene un significado semántico

p, blockquote, cite, code, em y strong

**Demo **

Especificación oficial

CONQUERBLOCKS

p, blockquote, cite, code, em y strong

p, blockquote, cite, code, em y strong

‹em›

- Se utiliza para marcar una parte del texto con más énfasis o como si fuera
- Tiene menos relevancia que strong
- Atributos comunes
- El navegador lo renderizará por defecto como texto en cursiva en la mayoría de los casos

p, blockquote, cite, code, em y strong

Demo

Especificación oficial

Entidades

Entidades

Caracteres protegidos o especiales de html.

Lista de entidades

Enlaces

Enlaces

Es la etiqueta que hizo que html tuviera la repercusión que tuvo en su día

Enlaces

<a>

La utilizaremos para enlazar a:

- Otros documentos internos
- Documentos externos
- Sección dentro del mismo documento

- Atributos básicos y de eventos

- Atributos específicos obligatorios

Enlaces

Atributo href

El valor de este atributo será una url
o un segmento de una url

Es un atributo obligatorio

Enlaces

Atributo href

NO NOS COMPLIQUEMOS
LA VIDA CON LAS URLS

Enlaces

Atributo href

- No espacios
- No caracteres tipo entidad
- No mezclar mayúsculas y minúsculas
- Cualquier cosa que se pueda simplificar, hacedlo
- Esto hará que nombremos los ficheros de forma correcta

Enlaces

Atributo download

Es un atributo que puede ir vacío o puede contener el nombre del fichero con el que se descargará el documento

Para que funcione, la página debe ser servida por un servidor web no abriendo el archivo en local

Enlaces

Atributo target

Este atributo indicará dónde quieres que se abra en enlace

Opciones:

_blank: se abrirá el destino en una pestaña nueva

_self: Abre el enlace en la pestaña o <iframe> actual.

Enlaces

Atributo hreflang

Establece el idioma del documento al que se accederá

Valores posibles

cualquier opción del estándar ISO 639-1

Estados de un enlace

- Enlaces sin visitas: Un enlace que no ha sido visitado aún es representado por los navegadores como un enlace de color azul y subrayado.

- Enlaces visitados: Un enlace que ha sido visitado previamente es representado por los navegadores como un enlace de color lila y subrayado.

- Enlaces activos: Un enlace está en estado activo cuando el usuario está pulsando sobre el enlace con el ratón y a la misma vez no ha dejado de pulsar en él.

Demo enlaces

Especificación oficial

Enlaces

Rutas relativas y absolutas

enlace al video

Espacios en blanco en HTML

Espacios en blanco en HTML

Espacios en blanco

- Html no procesa más de un espacio segudio
- Tampoco tiene en cuenta los saltos de linea en texto
- Propiedades propias de html

Espacios en blanco en HTML

** **

- Espacio manual
- Al ser una entidad, no es tratado como un espacio
- No abuses... recuerda separar información y presentación

Espacios en blanco en HTML

**
**

- Salto de línea manual en cualquier bloque de texto
- Es una etiqueta vacía
- Sin control en responsive
- No abuses... recuerda separar información y presentación

CONQUERBLOCKS

Espacios en blanco en HTML

<wbr />

- Oportunidad de salto de línea
- Para palabras demasiado largas
- Controlamos por donde partirá la línea cuando sea necesario

CONQUERBLOCKS

Espacios en blanco en HTML

<pre>

- Bloque de código donde se respetan los espcios
- Nada recomendable salvo para mostrar código y que se guarden las identaciones
- Controlamos por donde partirá la línea cuando sea necesario

Tareas

Tareas

Tarea 5.1

Utilizando las etiquetas que hemos visto en esta clase, realizar un documento html usando la plantilla boilerplate del tema pasado, que tenga el siguiente aspecto.

Tareas

HTML

HTML son las siglas de *HyperText Markup Language*, que puede traducirse como lenguaje de marcas o marcado de hipertexto.

El lenguaje **HTML** se emplea para crear las páginas web. Es muy fácil ver el código **HTML** de una página web, la opción exacta cambia de un navegador a otro y también puede cambiar de una versión a otra de un mismo navegador, pero suelen tener un nombre similar.

HTML se compone de etiquetas que se escriben entre los símbolos menor que y mayor que.

Historia de HTML

Los inicios del lenguaje **HTML** se remontan al año 1990, cuando *Tim Berners-Lee* creó la primera página web.

Versiones de HTML

Tim Berners-Lee definió la primera versión de **HTML** en el año 1991.

En la actualidad, la última versión de **HTML** es HTML5.

Tareas

Tarea 5.2

Utilizando las etiquetas vistas en clase y los enlaces, intentar crear un documento con ejemplos de cada tipo de etiqueta y enlazarlos en un documento inicial parecido a los que os enseño yo en la clase. Enlazar el documento creado en la tarea 5.1 también en la lista de enlaces

CONQUERBLOCKS

‹Despedida›

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

<HTML>

Clase 06 - Tema 5

CONQUER BLOCKS

<Índice>

Tema 5: Etiquetas semánticas y su importancia

Etiquetas semánticas de estructura

Header, Footer y Main

Nav, Section y Aside

Etiquetas semánticas de texto: strong, em, time ...

Etiquetas no recomendadas: i, b, u, small

CONQUER BLOCKS

Etiquetas semánticas de estructura

CONQUERBLOCKS

Etiquetas semánticas de estructura

¿Qué son?

- Etiquetas de agrupación
- Etiquetas de bloque
- Dan mucho valor a nuestros documentos HTML

CONQUERBLOCKS

Etiquetas semánticas de estructura

- En versiones anteriores a HTML5, al crear la estructura de una página, normalmente utilizabamos etiquetas para ir agrupando secciones diferentes. Unido a esto, ibamos añadiendo los atributos id o class dependiendo de nuestro interés, para que quedase más claro.

- De hecho, hoy en día aún es muy habitual crear una estructura de etiquetas basada en simplemente etiquetas HTML que no proporcionan ninguna información adicional, aunque existan etiquetas alternativas mucho más apropiadas.

Etiquetas semánticas de estructura

- No hay más que navegar y mirar con el inspector de elementos para comprobar que multitud de sitios aun no las utilizan

- SEO

- Accesibilidad

Etiquetas semánticas de estructura

- Ejemplo de estructura no semántica
- Ejemplo de estructura semántica

Etiquetas semánticas de estructura

```
<article></article>
```

Etiquetas semánticas de estructura

article

- Una de las etiquetas semánticas de sección más importantes de HTML5
- No se trata de una etiqueta para definir artículos
- Sirve para definir partes con la suficientemente importancia como para considerarse una entidad destacable con información a su alrededor.

Etiquetas semánticas de estructura

article

- La naturaleza del <article> dependerá del tipo de página: un artículo en un blog, una receta de cocina, una canción o video, un canal de Twitch, un widget personalizado en la página, etc...
- Es posible encontrar etiquetas article dentro de otras etiquetas article. Por ejemplo las noticias recomendadas a una noticia

Article

Especificación oficial


```
<nav></nav>
```

Etiquetas semánticas de estructura

nav

- Englobará a una lista de enlaces que normalmente será el menú de navegación de la web
- Dentro del nav podrá haber una sucesión de etiquetas a o una lista de elementos (ya los veremos)

Etiquetas semánticas de estructura

nav

- Ayudará a los robots y navegadores sin gráficos a navegar por nuestra web
- Es posible encontrarnos varios elementos nav en nuestras páginas webs

Nav

Especificación oficial


```
<header></header>
```

Etiquetas semánticas de estructura

header

- Encerrará a la zona común de la parte superior de nuestra web
- Normalmente estará compuesto por el logotipo y demás componentes que se repiten en todas las páginas

Etiquetas semánticas de estructura

header

- No confundir el header con el head
- Es posible encontrar también dentro de una sección o un article a otra etiqueta header. Por ejemplo en la cabecera de un post de un blog

Header

Especificación oficial


```
<footer></footer>
```

Etiquetas semánticas de estructura

footer

- Encerrará a la zona común de la parte inferior de nuestra web
- Normalmente estará compuesto por el menú inferior, páginas legales, copyright y demás componentes que se repiten en todas las páginas

Etiquetas semánticas de estructura

footer

- Es posible encontrar también dentro de una sección o un article a otra etiqueta header. Por ejemplo en la cabecera de un post de un blog

Footer

Especificación oficial


```
<section></section>
```

Etiquetas semánticas de estructura

section

- La etiqueta <section> nos permite agrupar una sección de nuestro documento web.
- Una forma fácil de saber si una parte de nuestra web es una sección, es cuando pensamos en dicha agrupación y colocamos un encabeza para identificarla

Etiquetas semánticas de estructura

section

Ejemplos de section

- Cada una de las partes de una página web.
- Cada capítulo de un documento.
- Cada una de las pestañas o tabs de un cuadro de diálogo.
- Cada una de las secciones numeradas de un documento.

Section

Especificación oficial


```
<main></main>
```

Etiquetas semánticas de estructura

main

- La etiqueta <main> la utilizaremos para agrupar el conjunto de contenido de nuestra página
- A grandes rasgos es como si fuera la parte central y relevante de la misma
- Suele ser elemento hijo de etiquetas como <body>, o

Etiquetas semánticas de estructura

main

Especificación oficial

CONQUERBLOCKS

Etiquetas semánticas de estructura

```
<aside></aside>
```


Etiquetas semánticas de estructura

aside

- Es todo lo opuesto a la etiqueta main
- Normalmente lleva dentro la parte más irrelevante de la web
- Permite agrupar contenido que no es considerado el principal de un documento, es decir, el contenido secundario o que no tiene relación directa con el tema principal del documento.

Etiquetas semánticas de estructura

aside

Ejemplos de aside

- Menús o sidebars
- Publicidad
- Zonas de navegación
- Enlaces relacionados

Etiquetas semánticas de estructura

aside

Especificación oficial

CONQUERBLOCKS

Etiquetas semánticas de estructura

- Ejemplo de estructura no semántica
- Ejemplo de estructura semántica

Etiquetas semánticas de texto

Etiquetas semánticas de texto

¿Qué son?

- Nos ayudan a dar relevancia a partes del texto
- Suelen ser etiquetas de línea
- Aportan valor semántico a nuestros textos

Etiquetas semánticas de texto

- Estas etiquetas no tienen significado gráfico ni visual. Otra cosa es que los navegadores por defecto dispongan su presentación de forma resaltada
- Debemos centrarnos en el significado semántico de las mismas

Etiquetas semánticas de texto

- SEO
- Accesibilidad

Etiquetas semánticas de estructura

```
<strong></strong>
```

Etiquetas semánticas de estructura

strong

- Sirve para dar más relevancia a unas partes del texto que otras
- Normalmente será mostrado en negrita
- Es una etiqueta de línea

Etiquetas semánticas de estructura

strong

Especificación oficial

CONQUERBLOCKS

Etiquetas semánticas de estructura

Etiquetas semánticas de estructura

em

- Sirve para dar énfasis a una parte del texto comparado con el resto
- Normalmente será mostrado en cursiva
- Es una etiqueta de línea

em

Especificación oficial


```
<time></time>
```

Etiquetas semánticas de estructura

time

- Lo usaremos para marcar una fecha o una hora
- Accesibilidad
- Dispone de un atributo datetime en formato timestamp en formato string

Etiquetas semánticas de estructura

time

Especificación oficial

CONQUERBLOCKS

Etiquetas semánticas de estructura

```
<address></address>
```


Etiquetas semánticas de estructura

address

- Lo usaremos para indicar la forma de contacto o dirección que mostraremos en nuestra web
- Accesibilidad
- Puede ser desde una url, una dirección, un email etc...

Etiquetas semánticas de estructura

address

Especificación oficial

Etiquetas no recomendadas

Etiquetas semánticas de estructura

- Debido a las diferentes versiones de HTML es normal encontrar usos de etiquetas que no deben aplicarse hoy en día
- Vamos a ver algunos ejemplos

Etiquetas semánticas de estructura

<b>

Sirve para poner el texto en negrita sin darle la importancia que tiene un strong.
Se desaconseja su uso por mezclar información y presentación

Etiquetas semánticas de estructura

<i>

Sirve para poner el texto en otra voz.

Mejor usar

Etiquetas semánticas de estructura

<small> <u>

Demo y tareas

CONQUERBLOCKS

‹Despedida›

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

<HTML>

Clase 07 - Tema 6

CONQUER BLOCKS

<Índice>

Tema 6: Etiquetas de agrupación

Etiquetas vistas en el tema anterior

Div

Span

Blockquote

hr

Listas: ul, ol y dl

Figure

CONQUER BLOCKS

Etiquetas de agrupación

Etiquetas de agrupación

¿Qué son?

- En html existen dos tipos de etiquetas: de línea y de bloque
- Además existen otros dos tipos según lo que contienen:
  - Etiquetas que contienen solo texto
  - **Etiquetas que contienen otras etiquetas**

Etiquetas de agrupación

En esta ocasión, vamos a centrarnos en el segundo grupo, que son las etiquetas que se utilizan para agrupar y asociar información relacionada

Etiquetas de agrupación

¿Para qué agrupar?

- Identificarla como una sección específica
- Para darle estilo visual (CSS)
- Para darle funcionalidad (Javascript)

Etiquetas de agrupación

Hay que encontrar el equilibrio entre ser ordenados y tener etiquetas de más. Aunque ante la duda, mejor que sobre que que falte, te será más fácil darle estilos después

Etiquetas vistas en el tema anterior

Etiquetas vistas en el tema anterior

section, main, header, footer, aside, article...

Etiquetas vistas en el tema anterior

Todas las etiquetas semánticas
son al la vez de agrupación.

Se encargan precisamente de eso
de agrupar contenido semántico

CONQUERBLOCKS

Etiquetas vistas en el tema anterior

¿Os acordáis de la tarea en Figma?
Grupos de información es lo que hacíais
sin apenas daros cuenta

‹div›

‹div›

‹div›

- De las etiquetas más conocidas
- De las etiquetas que menos sirven
- De las etiquetas que más abusamos
- Atributos comunes y eventos

‹div›

‹div›

La etiqueta ‹div› es una etiqueta muy simple que se suele utilizar para crear una agrupación de una o varias etiquetas y organizarlas en una división específica.

‹div›

‹div›

```html
<div class="noticia">
    <h3>Título de la noticia 4</h3>
    <p>Descripción de la noticia 4</p>
</div>
```

CONQUERBLOCKS

‹div›

‹div›

```html
<div class="noticias-container">
    <h2>Noticias de Deportes</h2>
    <div class="noticia">
        <h3>Título de la noticia 1</h3>
        <p>Descripción de la noticia 1</p>
    </div>
    <div class="noticia">
        <h3>Título de la noticia 2</h3>
        <p>Descripción de la noticia 2</p>
    </div>
    <div class="noticia">
        <h3>Título de la noticia 3</h3>
        <p>Descripción de la noticia 3</p>
    </div>
    <div class="noticia">
        <h3>Título de la noticia 4</h3>
        <p>Descripción de la noticia 4</p>
    </div>
</div>
```

CONQUERBLOCKS

‹div›

‹div›

- Es una etiqueta que no aporta significado especial de ningún tipo.
- Es necesario tener una lógica a la hora de agrupar la información
- No ser redundante

CONQUERBLOCKS

‹div›

‹div›

- Es una etiqueta de bloque
- Los navegadores por defecto no aportan ninguna información gráfica de más
- Mucho cuidado al agrupar etiquetas

CONQUERBLOCKS

Recurso

Can I include

Cuidado

```
<div class="x9f619 x1n2onr6 x1ja2u2z">
  <div class="x78zum5 xdt5ytf x1n2onr6 x1ja2u2z">
    <div class="x78zum5 xdt5ytf x1n2onr6">
      <div class="x78zum5 xdt5ytf x1n2onr6 xat3117 xxzkxad">
        <div class="x78zum5 xdt5ytf x10cihs4 x1t2pt76 x1n2onr6 x1ja2u2z">
          <!— varios elementos div anidados más —>
        </div>
      </div>
    </div>
  </div>
</div>
```

Cuidado

- Accesibilidad
- SEO
- El código HTML ocupará más tamaño
- Al ser más grande, tardará más en descargarse
- El navegador tardará más tiempo en renderizar la página (dibujarla)
- Hay más probabilidades de que sea menos accesible

Demo ‹div›

Especificación oficial

‹span ›

CONQUERBLOCKS

‹span›

‹span›

- De las etiquetas más conocidas
- De las etiquetas que menos sirven
- De las etiquetas que más abusamos
- Atributos comunes y eventos

CONQUERBLOCKS

- Se puede decir que la etiqueta span es como la etiqueta div, pero por el contrario es una etiqueta de línea.
- La usaremos para marcar el texto para posteriormente dar estilos o interactividad con JS.

- No aporta semántica de ningún tipo
- No cambia el flujo natural del texto


```
<p>
  Este <strong>párrafo</strong> de texto tiene <span>algunos fragmentos de texto</span>
  que mantienen su <span>disposición original</span>.
</p>
```


Recuerda

- Una etiqueta de línea puede contener más etiquetas de línea
- Una etiqueta de línea no puede contener etiquetas de bloque

‹span›

```
<p>
  Esto es un <span>fragmento de <div>texto</div> inválido</span>.
</p>
```


‹span›

Demo ‹span›

Especificación oficial

‹blockquote ›

CONQUERBLOCKS

‹blockquote›

‹blockquote›

- Se utiliza para indicar que el texto pertenece a otro sitio externo del cual se indica la fuente
- Es una etiqueta de bloque
- Atributos comunes y eventos
- Atributo específico cite de tipo url para indicar el origen de la cita

CONQUERBLOCKS

‹hr ›

‹hr›

‹hr›

- Se utiliza para separar bloques de contenido de otros bloques de contenido
- Inserta una línea horizontal
- Es un elemento de bloque
- Cada navegador la pintará a su manera hasta que no le digamos lo contrario

<hr>

<hr>

- La podemos encontrar además como etiqueta de separación semántica
- Aporta semántica, la separación de un tema con otro
- Lo visual no es relevante porque se puede personalizar

<hr>

Demo <hr>

Especificación oficial

Listas
ul, ol y dl

ul, ol

ul, ol

- Son elementos de bloque
- Se utilizan para organizar información en formato de listado
- Disponen de etiquetas hijas li
- Ambas funcionan de la misma manera, lo único que cambia es el “orden”

ul

ul

- Siglas de (lista sin orden)
- Se utilizará para listar una serie de elementos en los que no importa el orden
- Lista de personas, ingredientes...

‹ul›

Demo ‹ul›

Especificación oficial

ol

ol

- Orderer list
- Se utilizará para listar una serie de elementos en los que importa el orden
- Lista de pasos a seguir
- Diferentes tipos de numeración

‹ol›

Demo ‹ol›

Especificación oficial

li

li

- List item
- Elemento de bloque
- Dependerá del padre de su presentación si con viñeta o con número

CONQUERBLOCKS

dl

dl

- Definition/Description List
- Se utiliza para listar elementos y definirlos
- Muy usada en diccionarios

CONQUERBLOCKS

dl

dl

- Este tipo de lista no dispone de etiquetas hijas li
- Tiene sus propias etiquetas hijas
- Muy usada en diccionarios

dl

dl

- dt: Término a definir
- dd: Definición del término

dl

Demo <dl>

Especificación oficial

Listas anidadas

Listas anidadas

- Inicio
- Sobre mí
- servicios
  - Fontanería
  - Electricidad
  - Pintura
    - Parkings
    - Trabajos en altura
    - Domicilios privados
- Testimonios
- Contacto

Listas anidadas

- Una lista puede contener otra lista en su interior
- SIEMPRE DENTRO DE UN LI
- Hay que tener especial cuidado a la hora de abrir y cerrar las etiquetas
- De lo contrario será inmanejable
- Pueden mezclarse los tipos de listas

Práctica de listas anidadas

Figure y Figcaption

Figure y Figcaption

- Figure es una etiqueta de agrupación para imágenes
- Encerrará a una etiqueta img básica
- Se manejará mucho mejor con CSS en el futuro

Figure y Figcaption

- Recuerda que en la ilustración o figura, se pueden contener todo tipo de elementos, no solo imágenes. Ejemplos válidos podrían ser videos, párrafos de texto o incluso contenido de otro tipo.

Figure y Figcaption

- Figcaption añade significado semántico a la imagen
- Ayudará al atributo alt del img
- Normalmente se utiliza como el texto pie de la imagen

Figure y Figcaption

```
<figure>
  <img src="http://lenguajehtml.com/img/html5-logo.png" alt="Logotipo de HTML5">
  <figcaption>Logotipo oficial del lenguaje de marcas HTML5.</figcaption>
</figure>
```

Figure y Figcaption

Demo Figure y Figcaption

Especificación oficial

Demo y tareas

<Despedida>

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

<HTML>

Clase 08 - Tema 7

CONQUER BLOCKS

<Índice>

Tema 7: Tablas

Antiguos usos de las tablas

Table

Elementos básicos: tr, th, td y caption

Colspan y Rowspan

Tablas semánticas

thead, tbody y tfoot

CONQUER BLOCKS

Tablas

Tablas

¿Qué son?

- Las tablas están incluidas en HTML desde sus primeras versiones y son una forma fantástica de mostrar datos claramente
- Se utilizan para mostrar datos tabulados y listados de información organizada

Tablas

Se deben utilizar **únicamente** para mostrar datos organizados de forma tabular con encabezados

Tablas

Una tabla puede ser sencilla o compleja, dependiendo de nuestro objetivo y la cantidad de etiquetas o atributos a utilizar.

Antiguos usos de las tablas

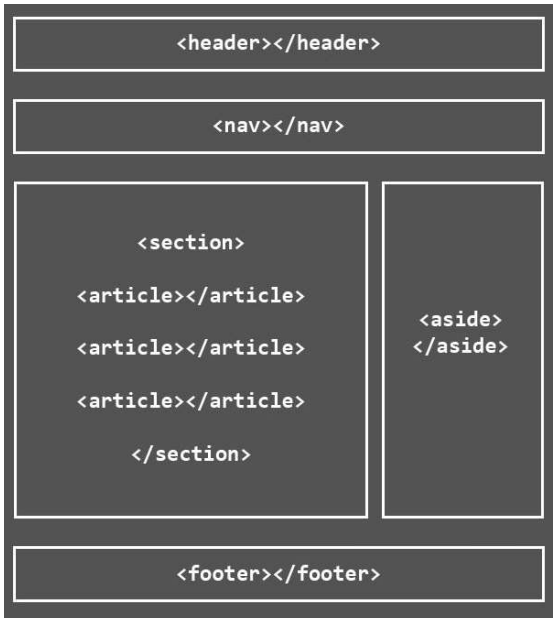
Antiguos usos de las tablas

Aunque parezca ciencia ficción, antiguamente se usaban para maquetar estructuras de página complejas cuando CSS no era tan potente y sobre todo cuando los navegadores no lo soportaban todos por igual.

La cabecera de la página era una fila de una gran tabla, el pie de página era otra fila de esta tabla y la zona de contenidos estaba formada por varias columnas dentro de esa gran tabla.

Antiguos usos de las tablas

¿Os acordáis de las etiquetas semánticas?



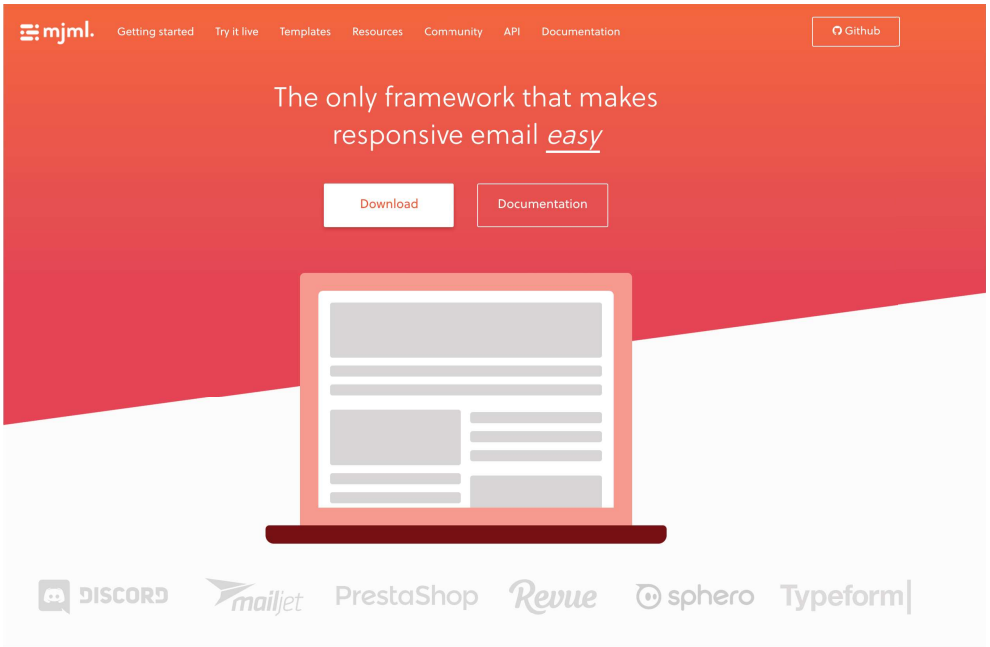
Antiguos usos de las tablas

A día de hoy se siguen usando para maquetar cosas muy específicas como por ejemplo los correos electrónicos que tienen diseño y estructura

Ejemplo

Antiguos usos de las tablas

MJML



<table>

‹table›

‹table›

- Englobará todo el contenido de la tabla
- Etiqueta de bloque
- Atributos comunes y eventos
- Atributo propio: summary = "texto" que Permite describir el contenido de la tabla para buscadores y accesibilidad

CONQUERBLOCKS

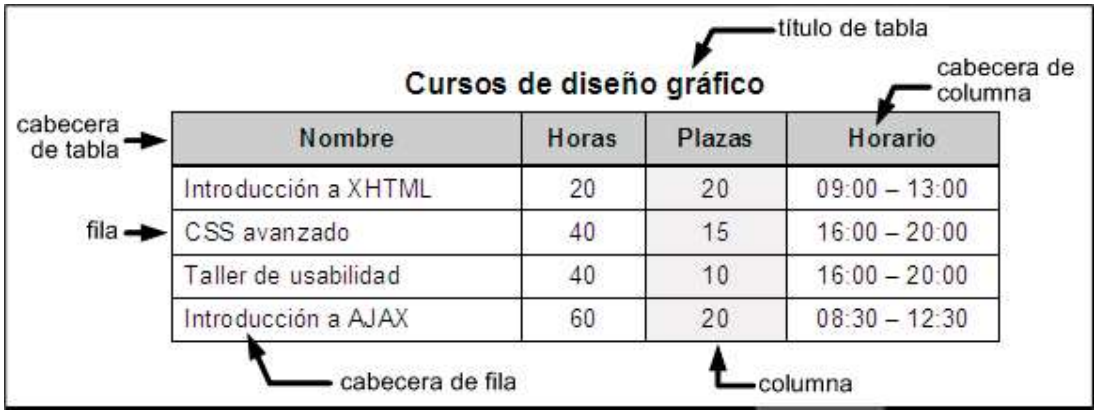
‹table›

‹table›

- Atributo border **solo para este momento**
- Atributo
- Atributos comunes y eventos
- Atributo propio: summary = "texto" que Permite describir el contenido de la tabla para buscadores y accesibilidad

CONQUERBLOCKS

‹table›



Nombre	Horas	Plazas	Horario
Introducción a XHTML	20	20	09:00 – 13:00
CSS avanzado	40	15	16:00 – 20:00
Taller de usabilidad	40	10	16:00 – 20:00
Introducción a AJAX	60	20	08:30 – 12:30

‹tr›

‹tr›

‹tr›

- Modela una fila de la tabla en la que esté incluída
- Etiquetas de bloque

‹table›

Cursos de diseño gráfico

Nombre	Horas	Plazas	Horario
Introducción a XHTML	20	20	09:00 – 13:00
CSS avanzado	40	15	16:00 – 20:00
Taller de usabilidad	40	10	16:00 – 20:00
Introducción a AJAX	60	20	08:30 – 12:30

‹th›

CONQUERBLOCKS

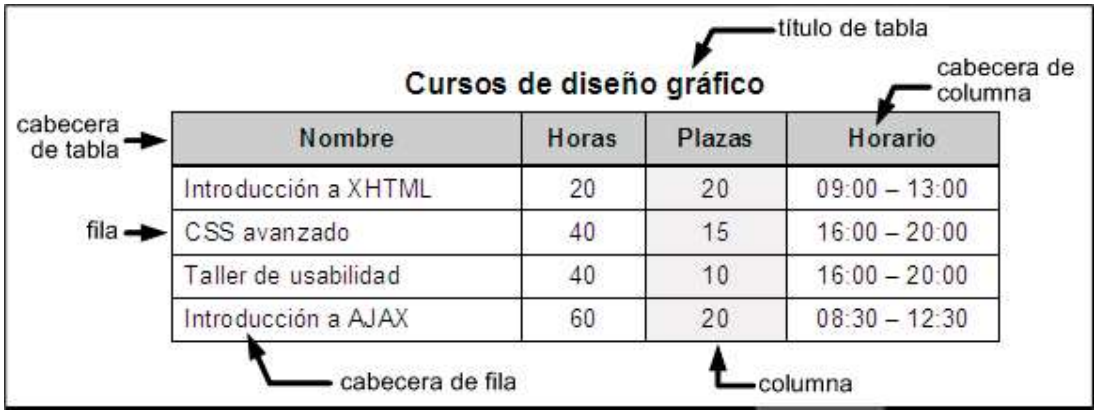
‹tr›

‹th›

- Celda de cabecera de fila o de columna
- Etiquetas de tipo nuevo, ni es de línea de es de bloque
- Los navegadores suelen renderizar su contenido en negrita
- Aporta semántica a nuestras tablas

CONQUERBLOCKS

<tr>



Nombre	Horas	Plazas	Horario
Introducción a XHTML	20	20	09:00 – 13:00
CSS avanzado	40	15	16:00 – 20:00
Taller de usabilidad	40	10	16:00 – 20:00
Introducción a AJAX	60	20	08:30 – 12:30

<td>

‹td›

‹td›

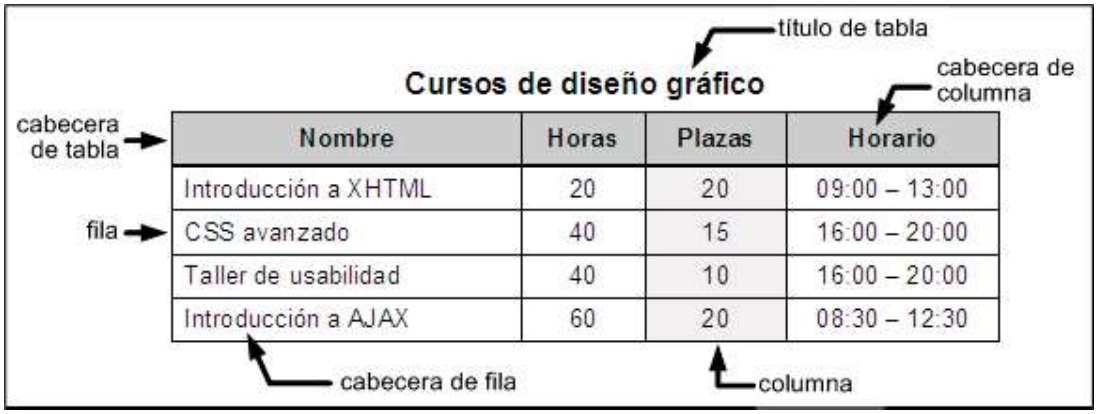
- Celda común de tabla
- Etiquetas de tipo nuevo, ni es de línea de es de bloque
- Son las que contendrán los datos regulares de la tabla

‹td›

‹td›

- Atributos específicos de fusión de celdas
- Colspan: para fusionar columnas
- Rowspan: para fusionar filas
- Luego veremos ejemplos

‹tr›


título de tabla
Cursos de diseño gráfico
cabecera de columna
cabecera de tabla
Nombre
Horas
Plazas
Horario
Introducción a XHTML
20
20
09:00 – 13:00
fila
CSS avanzado
40
15
16:00 – 20:00
Taller de usabilidad
40
10
16:00 – 20:00
Introducción a AJAX
60
20
08:30 – 12:30
cabecera de fila
columna

‹caption›

‹td›

‹caption›

- Lo usaremos para indicar en texto el contenido de la tabla
- Los navegadores accesibles lo usarán para dar una descripción de lo que pueden encontrar en la tabla
- El navegador lo renderizará a modo de título de la tabla

‹tr›

Cursos de diseño gráfico

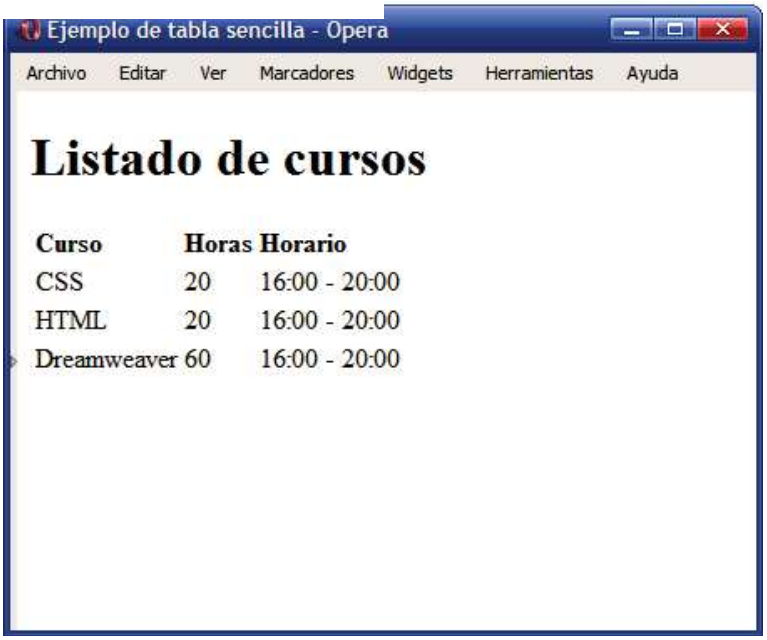
Nombre	Horas	Plazas	Horario
Introducción a XHTML	20	20	09:00 – 13:00
CSS avanzado	40	15	16:00 – 20:00
Taller de usabilidad	40	10	16:00 – 20:00
Introducción a AJAX	60	20	08:30 – 12:30

Tabla sencilla 1

CONQUERBLOCKS

‹td›

Tabla sencilla 1 histórica ;-)



CONQUERBLOCKS

Tabla sencilla 2

CONQUERBLOCKS

‹td›

Tabla sencilla 2

Nombre producto	Precio unitario	Unidades	Subtotal
Reproductor MP3 (80 GB)	192.02	1	192.02
Fundas de colores	2.50	5	12.50
Reproductor de radio & control remoto	12.99	1	12.99
TOTAL	-	7	**207.51**

CONQUERBLOCKS

Tabla sencilla 3

CONQUERBLOCKS

Tabla sencilla 3

AÑO	Producto A	Producto B	Producto C	Producto D
N-3	-	-	-	-
N-2	3	5	8	4
N-1	4	4	7	3
N	5	7	6	2

CONQUERBLOCKS

<Despedida>

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

<HTML>

Clase 09 - Tema 7b

CONQUER BLOCKS

<Índice>

Tema 7b: Tablas Avanzadas

Antiguos usos de las tablas

Table

Elementos básicos: tr, th, td y caption

Colspan y Rowspan

Tablas semánticas

thead, tbody y tfoot

CONQUER BLOCKS

Colspan

Colspan

¿Qué son?

- Atributos de la celdas
- th y td

Colspan

Nos servirán para
fusionar celdas adyacentes
por columnas

Colspan

Una tabla puede ser sencilla o compleja, dependiendo de nuestro objetivo y la cantidad de etiquetas o atributos a utilizar.

Colspan

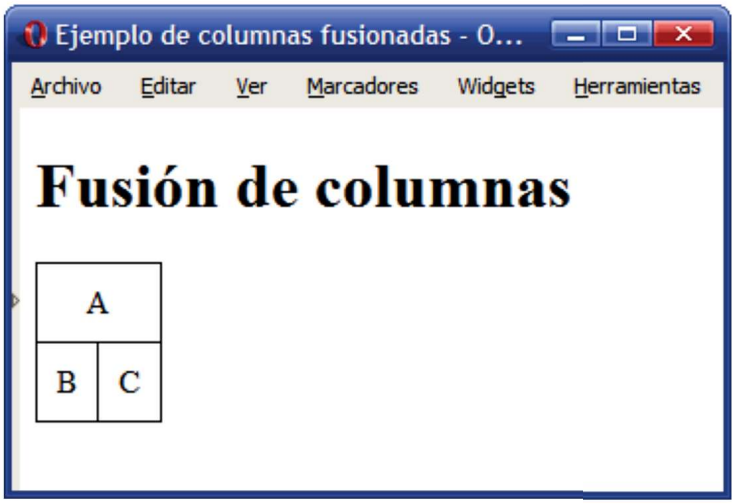


Figura 7.5 Ejemplo sencillo de fusión de columnas

Colspan

Demo colspan

Rowspan

Rowspan

¿Qué son?

- Atributos de la celdas
- th y td

Rowspan

Nos servirán para
fusionar celdas adyacentes
por filas

Rowspan

Una tabla puede ser sencilla o compleja, dependiendo de nuestro objetivo y la cantidad de etiquetas o atributos a utilizar.

Colspan

Figura 7.6 Ejemplo sencillo de fusión de filas

Colspan

Demo colspan

Colspan + Rowspan

colspan + rowspan

- Contabilizar total de filas
- Contabilizar total de columnas
- Comprobar que la suma de td's + colspan dan el total de columnas por fila
- Comprobar que la suma de tr + rowspan dan el total de filas por columna

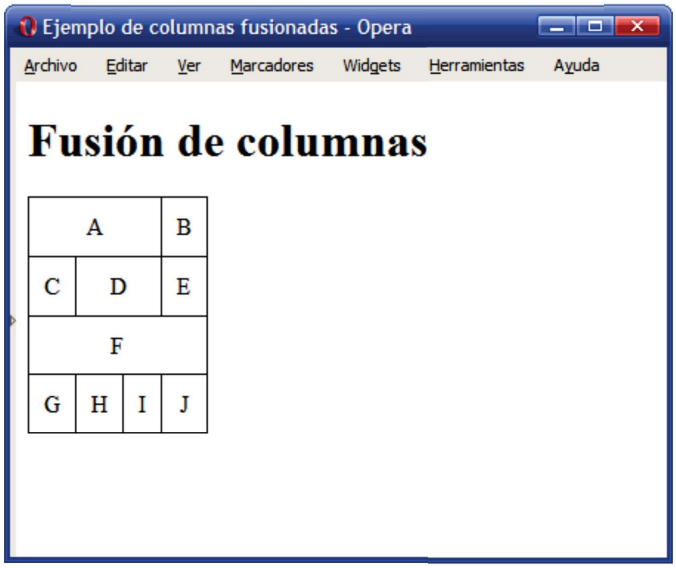


Figura 7.7 Ejemplo complejo de fusión de columnas

colspan + rowspan

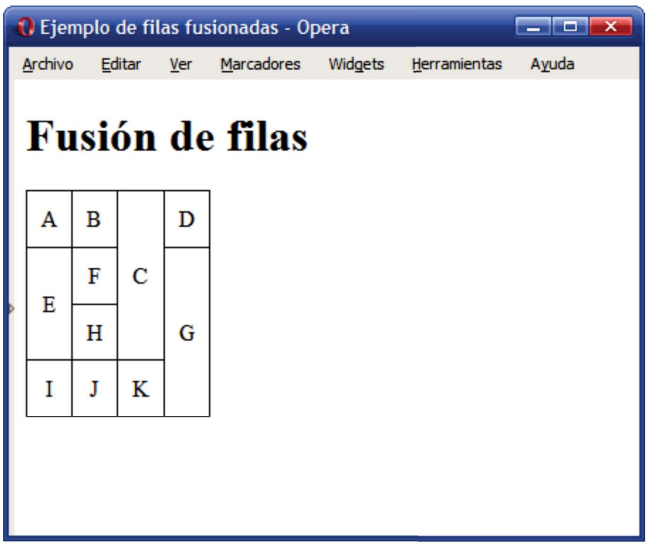


Figura 7.8 Ejemplo complejo de fusión de filas

CONQUERBLOCKS

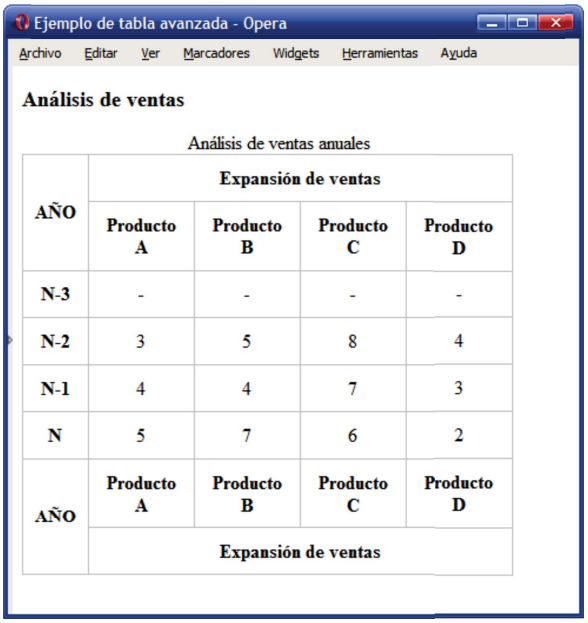
colspan + rowspan

Demo

colspan + rowspan

Juntemos todo

colspan + rowspan



Análisis de ventas

Análisis de ventas anuales

AÑO	Expansión de ventas			
	Producto A	Producto B	Producto C	Producto D
N-3	-	-	-	-
N-2	3	5	8	4
N-1	4	4	7	3
N	5	7	6	2
AÑO	Producto A	Producto B	Producto C	Producto D
	Expansión de ventas			

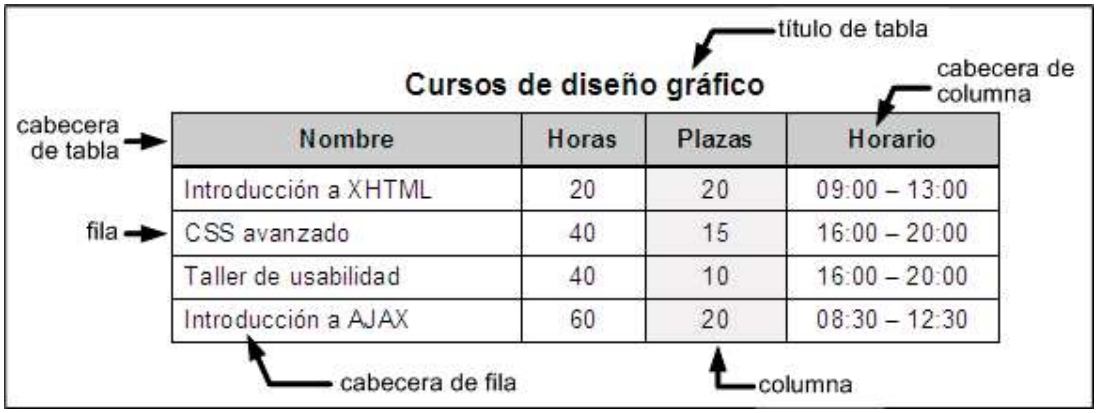
Figura 7.11 Ejemplo de tabla avanzada con cabecera, pie y secciones

Tablas semánticas

Tablas semánticas

- Ayudan al igual quer las etiquetas semánticas, a añadir significado a los datos tabulados
- Dividiremos las tablas en tres partes

Tablas semánticas



Cursos de diseño gráfico

Nombre	Horas	Plazas	Horario
Introducción a XHTML	20	20	09:00 – 13:00
CSS avanzado	40	15	16:00 – 20:00
Taller de usabilidad	40	10	16:00 – 20:00
Introducción a AJAX	60	20	08:30 – 12:30

‹thead›

‹thead›

‹thead›

- Aglutinará todas las filas con celdas de cabecera
- Etiqueta hija de tbody
- Aporta semántica a nuestras tablas

‹thead›

Nombre	Horas	Plazas	Horario
Introducción a XHTML	20	20	09:00 – 13:00
CSS avanzado	40	15	16:00 – 20:00
Taller de usabilidad	40	10	16:00 – 20:00
Introducción a AJAX	60	20	08:30 – 12:30

‹tbody›

‹tbody›

‹tbody›

- Contendrá todas los datos de la tabla
- Después de thead siempre y cuando no exista tfoot

‹tbody›

Cursos de diseño gráfico

Nombre	Horas	Plazas	Horario
Introducción a XHTML	20	20	09:00 – 13:00
CSS avanzado	40	15	16:00 – 20:00
Taller de usabilidad	40	10	16:00 – 20:00
Introducción a AJAX	60	20	08:30 – 12:30

‹tfoot›

CONQUERBLOCKS

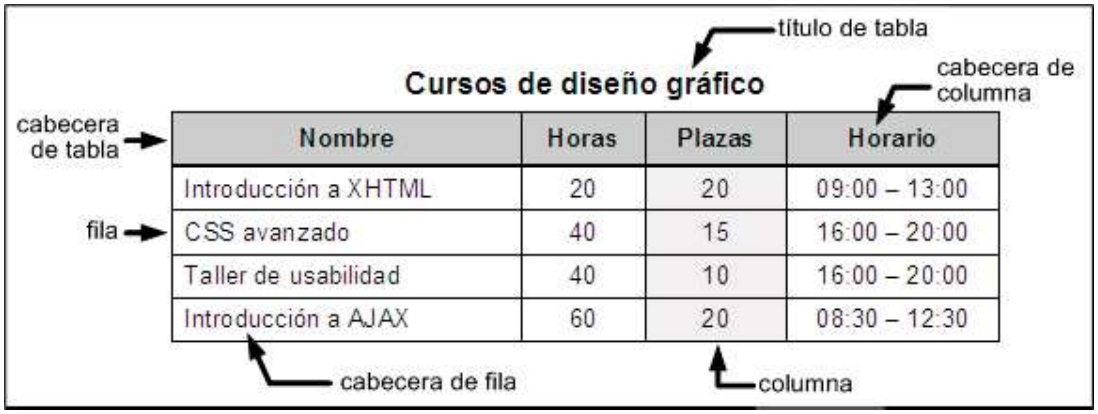
‹tfoot›

‹tfoot›

- Siempre antes de tbody
- Sirve para acumulados, contabilizaciones o repetir la cabecera

CONQUERBLOCKS

‹tfoot›

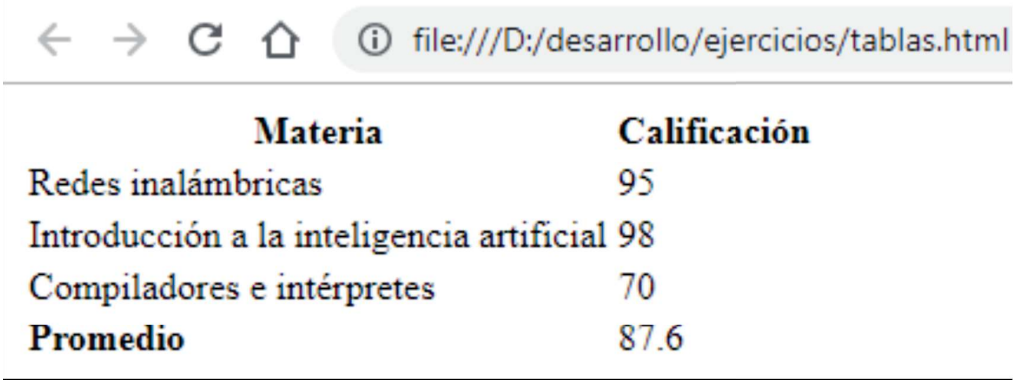


Cursos de diseño gráfico

Nombre	Horas	Plazas	Horario
Introducción a XHTML	20	20	09:00 – 13:00
CSS avanzado	40	15	16:00 – 20:00
Taller de usabilidad	40	10	16:00 – 20:00
Introducción a AJAX	60	20	08:30 – 12:30

‹tfoot›

Demo



Materia	Calificación
Redes inalámbricas	95
Introducción a la inteligencia artificial	98
Compiladores e intérpretes	70
Promedio	87.6

Tablas dentro de tablas

‹tfoot›

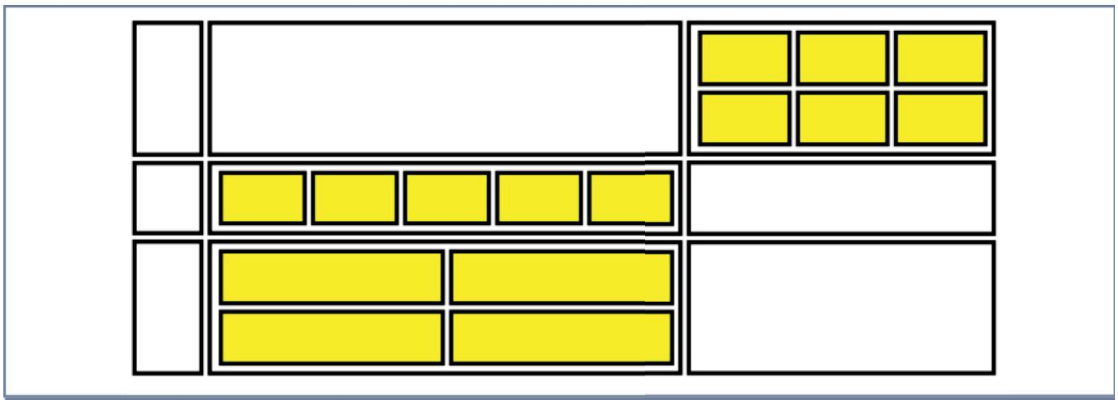


Ilustración 84. Maquetación compleja conseguida con tablas internas insertadas en celdas

<Despedida>

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

<HTML>

Clase 10 - Tema 8

CONQUER BLOCKS

<Índice>

Tema 10: Formularios I

¿Qué es un formulario?

La etiqueta form

Métodos de envío: GET vs POST

Inputs

Tipos: Text, Number, Email, Date, Range, Hidden...

CONQUER BLOCKS

¿Qué es un formulario?

Formularios

¿Qué son?

- Bloque con un conjunto de elementos de entrada de información
- Es de las únicas formas en las que podemos enviar datos a un servidor

Se conoce como **formulario** a los mecanismos para enviar información por parte del usuario, a través de unos campos visuales de forma sencilla e intuitiva, que determinan la naturaleza del tipo de información que se va a enviar y como hacerla llegar a otro extremo donde se procesará esa información.

¿Cual es el formulario más sencillo que recuerdas?

Formularios

Objetivos

- Hacer lo más sencillo posible el proceso de inserción de datos por parte del usuario.
- Intentar que la experiencia de usuario sea lo más agradable y cómoda posible.
- Intentar que los datos introducidos estén en un formato predecible y esperado.

Objetivos

- Reducir los errores al introducir datos en la medida de lo posible.
- Comunicar clara y rápidamente si ocurren errores al introducir datos.

Etiqueta form

‹form›

‹form›

- Etiqueta de bloque
- Atributos propios
- Englobará todo el contenido del formulario
- Pueden coexistir varios formularios en una misma página

‹form›

Atributos propios

Atributo	Descripción	Más información
action	Dirección URL del backend al cuál se enviará la información obtenida en el formulario.	
method	Método HTTP de envío. GET a través de URL, POST para envío extenso de información.	
name	Nombre del formulario. Útil para identificar el formulario y procesar posteriormente.	
target	Nombre del lugar donde se abrirá el formulario. _blank para nueva pestaña.	
enctype	Codificación para el envío del formulario. Importante para envío de archivos.	Envío de archivos
accept-charset	Fuerza a utilizar una codificación en los parámetros de texto del formulario.	
autocomplete	Activa on o desactiva off el autocompletado para todos los campos del formulario.	
novalidate	BOOLEAN Desactiva la validación HTML5 del formulario.	

Información que puedes enviar en un formulario

Texto
Emails
Cantidades
Numéricas
Fechas
etc...

‹form›

Información a obtener	Ejemplos	Etiqueta y atributo a utilizar
Información de texto corto	Nombres, apellidos, textos cortos...	`<input>`
Información de texto largo	Comentarios, observaciones...	`<textarea>`
Números o cantidades	Edades, precios, cuantías...	`<input type="number">` ó `<input type="range">`
Fechas u horas	Fecha de nacimiento, inicio de evento...	`<input type="date">` o derivados
Verdadero/falso	Si/No, Opción A/B, ON/OFF...	`<input type="checkbox">`
Opción única	(Elegir una opción de 2 ó más posibles)	`<input type="radio">` ó `<select>`
Varias opciones	(Elegir varias opciones de 2 ó más posibles)	`<select multiple>` ó `<input type="checkbox">`
Opción única abierta	(Elegir una opción o indicar una propia)	`<datalist>`
Selección de color	(Escoger un color o tonalidad)	`<input type="color">`
Selección de archivo	(Escoger un archivo para enviar)	`<input type="file">`

CONQUERBLOCKS

No os preocupéis por esto que lo vamos a ver todos poco a poco

Sintaxis básica de un formulario

```
<form action="/registrar_cliente/" name="registro" method="get">
    Nombre:<br />
    <input type="text" name="nombre">
    <button>Enviar</button>
</form>
```

Métodos de envío

CONQUERBLOCKS

Métodos de envío

get

- Concatenación de variables en URL
- Como máximo el envío de unos 500 bytes de información
- No permite el envío de archivos adjuntos con el formulario
- Los datos enviados se ven en la barra de direcciones del navegador

CONQUERBLOCKS

Métodos de envío

get

- El símbolo ? indica dónde empiezan los parámetros que se reciben desde el formulario que ha enviado los datos a la página.
- Después del símbolo ? aparecen parejas de datos con su nombre y valor separadas por el símbolo &. Las parejas dato1=valor1, dato2=valor2, dato3=valor3... reflejan el nombre y el valor de los campos enviados por el formulario.

Métodos de envío

get

- Tener en cuenta que para separar la primera pareja de la dirección web en sí se usa el símbolo '?' y para separar las restantes parejas entre sí se usa el símbolo '&'.
- Determinados caracteres no son recibidos en la URL de la misma forma exactamente en que fueron escritos en el formulario

Métodos de envío

www.facebook.com/registro?firstname=Bienve&lastname=Saez

CONQUERBLOCKS

¿Qué es una URL?

http://www.example.com:80/path/to/myfile.html?

key1=value1&key2=value2#SomewhereInTheDocument

CONQUERBLOCKS

¿Qué es una URL?

http://www.example.com:80/path

Scheme

¿Qué es una URL?

tp://www.example.com:80/path/to/my

Domain Name

¿Qué es una URL?

com:80/path/to/myfile.html?key1=valu

Port

HTML-C01

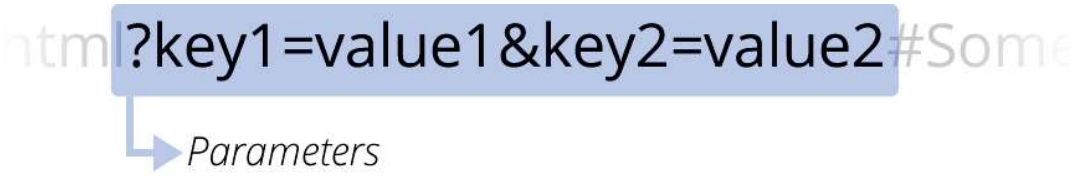
¿Qué es una URL?

path/to/myfile.html?key1=v

Path to resource

CONQUERBLOCKS

¿Qué es una URL?



Métodos de envío

Demo en Google

Métodos de envío

get

- Se utilizan para formularios de consulta
- No se usan para modificar ni crear registros

CONQUERBLOCKS

Métodos de envío

post

- Se envían de forma “oculta”
- Se pueden enviar una gran cantidad de datos a la vez
- Permite enviar documentos adjuntos

CONQUERBLOCKS

Métodos de envío

post

- Se usarán para acciones de creación, edición y borrado

Métodos de envío

CRUD

Create

Read

Update

Delete

input

input

- Elemento principal donde pondremos escribir la información que queremos enviar al servidor
- Dependiendo de la naturaleza de los datos utilizaremos input de un tipo u otro

input

button, checkbox, color, date, datetime-locale, mail, file, hidden, image, month, number, password, radio, range, reset, search, submit, tel, text, time, url, week

CONQUERBLOCKS

input

button, checkbox, color, date, datetime-local, **email**, **file**, **hidden**, image, month, **number**, **password**, radio, range, reset, search, **submit**, tel, **text**, time, **url**, week

CONQUERBLOCKS

input

W3C input

input

La importancia del name y el value

input

Demo

text

number

password

url

mail

button, reset

file

hidden

atributos propios

input

Attr

type

value

placeholder

<Despedida>

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

<HTML>

Clase 11 - Tema 8

CONQUER BLOCKS

<Índice>

Clase 11: Formularios II

Resto de inputs por ver

Label

Textarea

Select

Button vs Submit

Fieldset

Modificadores de validación y obligatoriedad

Chuleta

CONQUER BLOCKS

input II

CONQUERBLOCKS

input

checkbox, color, date, datetime-local, image, month, radio, range, reset, search, tel, time, week

CONQUERBLOCKS

input

La importancia del name y el value

CONQUERBLOCKS

input

search, image, color, date, time,
datetime-local, month, week, tel, range

CONQUERBLOCKS

input

Demo

search

image

color

date - time - datetime-local

month

week

tel

range

input

checkbox, radio

input

Demo

checkbox (ingredientes pizza)

radio (color de zapatillas)

input

¿Los hacemos accesibles?

label

label

- Uso del atributo for para unirlos
- Varias formas de agruparlo, con contenedor, envolviendo o de cualquier otra forma

label

label vs placeholder vs text simple

label

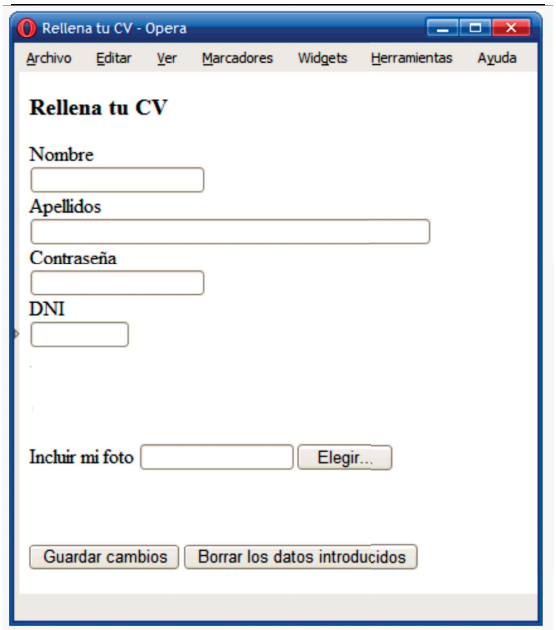


Figura 8.12 Formulario con controles de varios tipos

textarea

textarea

- Textos largos
- Atributos col y row

textarea

Demo textarea

select

select

- Para elegir entre varios
- Típico desplegable
- Atributos: multiple, required, size

select

Demo select

select

Demo múltiple

CONQUERBLOCKS

select

Demo select obligatorio

CONQUERBLOCKS

button vs submit

button vs submit

- Yo prefiero button
- Estilos básicos de botón
- Son igualmente válidos

button vs submit

Demo

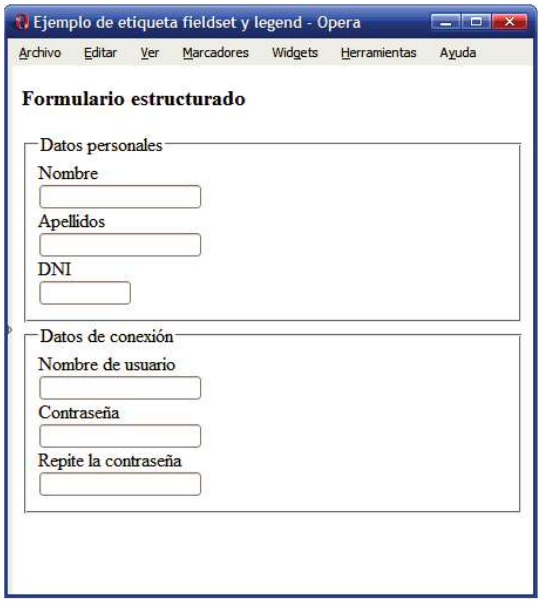
fieldset y legend

fieldset y legend

- Ordenación de campos
- Semántica con legend
- No obligatorio pero si recomendable

fieldset y legend

Demo



Modificadores de validación y obligatoriedad

CONQUERBLOCKS

Validación y Obligatoriedad

- Uso siempre aconsejado
- Nos ayudan pero no nos salvan
- Cuando estés en el back, nunca confíes en el front

CONQUERBLOCKS

Validación y Obligatoriedad

- required
- minlength, maxlength
- min, max
- type: email, url, etc...
- pattern

chuleta

chuleta

Tabla 2.6. Principales valores del atributo «type» del elemento <input>

Tipo	Tipo de entrada	Representación visual
button	Botón de acción.	Activar
checkbox	Casilla para la selección de un elemento dentro de un conjunto de opciones.	
color	Color.	
date	Fecha.	03/09/2025
datetime-local	Fecha y hora.	24/12/2028 15:12
email	Dirección de correo electrónico.	info@paraninfo.es
file	Selección de un fichero del sistema de archivos.	Seleccionar archivo Lenguaje de marcas.docx
hidden	No es visible pero internamente existe a todos los efectos.	
image	Tiene el mismo comportamiento que el componente **<input>** de tipo **submit,** pero en lugar de texto muestra una imagen.	
month	Mes y año.	septiembre de 1971
number	Valor numérico.	35000
password	Contraseña. Visualmente sustituye los caracteres introducidos por asteriscos o puntos.	••••••
radio	Casilla para la selección de un único elemento dentro de un conjunto de opciones.	
range	Selección visual de un valor numérico dentro de un rango.	
reset	Restaura el estado inicial de los campos del formulario.	Restablecer

Tipo	Tipo de entrada	Representación visual
search	Texto para realizar una búsqueda. Tiene el mismo comportamiento que el elemento **<input>** de tipo **text,** pero permite establecer una diferenciación para realizar un tratamiento distinto.	
submit	Verifica y envía los datos del formulario al destino indicado en el atributo **action** de este.	Enviar
tel	Número de teléfono.	+34630001100558899
text	Texto libre.	
time	Hora.	18:15
url	URL.	http://www.paraninfo.es
week	Año y ordinal de la semana de dicho año.	Semana 30, 2032

<Despedida>

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

<HTML>

Clase 13 - Imágenes en la web

CONQUER BLOCKS

<Índice>

Clase 13: Imágenes en la web

Etiqueta <img>

Etiqueta <figure> y <figcaption>

Formatos admitidos

Imágenes responsive y alternativas

Picture

Source

CONQUER BLOCKS

Contenido vs Presentación

Contenido vs Presentación

- Imágenes que pretenden mostrarse como parte del contenido de la web.
- Imágenes que se muestran como parte de decoración de la web.

CONQUERBLOCKS

Contenido vs Presentación

- Contenido => <img>
- Decoración => CSS (background-image)

Etiqueta img

‹img›

- Imágenes que pretenden mostrarse como parte del contenido de la web.
- Imágenes que se muestran como parte de decoración de la web.

‹img›

Atributos

Atributo	Descripción
`src`	Indica el nombre o la URL de la imagen a mostrar. **Atributo obligatorio**.
`alt`	Establece un texto alternativo que describa la imagen a mostrar. **Atributo obligatorio**.
`width`	Indica el ancho de la imagen en píxels (sin la unidad). Se puede hacer desde CSS.
`height`	Indica el alto de la imagen en píxels (sin la unidad). Se puede hacer desde CSS.

‹img›

Indicar el width y height

‹img›

La importancia de las urls

- Diferencia entre una url absoluta y url relativa
- Cómo enlazar correctamente ficheros dentro de nuestro proyecto
- Directorio . y directorio ..

Demo

Todo esto también aplica a los enlaces

‹img›

Atributo Loading

- Carga perezosa
- Valores:
  - eager: el navegador descarga inmediátamente
  - lazy: el navegador pospone la descarga hasta que es visible

‹img›

Demo

CONQUERBLOCKS

Figure y Figcaption

Figure y Figcaption

- Figure es una etiqueta de agrupación para imágenes
- Encerrará a una etiqueta img básica
- Se manejará mucho mejor con CSS en el futuro

CONQUERBLOCKS

Figure y Figcaption

- Recuerda que en la ilustración o figura, se pueden contener todo tipo de elementos, no solo imágenes. Ejemplos válidos podrían ser videos, párrafos de texto o incluso contenido de otro tipo.

Figure y Figcaption

- Figcaption añade significado semántico a la imagen
- Ayudará al atributo alt del img
- Normalmente se utiliza como el texto pie de la imagen

Figure y Figcaption

```
<figure>
  <img src="http://lenguajehtml.com/img/html5-logo.png" alt="Logotipo de HTML5">
  <figcaption>Logotipo oficial del lenguaje de marcas HTML5.</figcaption>
</figure>
```


Figure y Figcaption

Demo Figure y Figcaption

CONQUERBLOCKS

Formatos admitidos

Formatos admitidos

Formato	Características	¿Recomendado?
Formatos tradicionales		
PNG	Soporta transparencia. Compresión sin pérdidas. Imágenes «lisas».	✅
JPG	Compresión con pérdidas. Ideal para imágenes con texturas.	✅
SVG	Formato vectorial. Ideal para imágenes escalables.	✅
GIF	Imágenes pequeñas. Soporta animación. Pesado. Poca calidad.	⚠️ Evitar. Usar MP4 o APNG.
Formatos modernos		
WEBP	Alternativa libre de Google al JPEG, PNG y GIF. Transparencias y animaciones.	✅ Buen soporte
JPEG2000	Formato evolución de JPEG. Poco utilizado.	❌ Solo Safari
JPEG-XR	Alternativa libre de Microsoft al JPEG. Poco utilizado.	❌ Solo IE
APNG	Alternativa libre a GIF. Compatible con PNG. Soporta animaciones.	✅ Buen soporte
Formatos de nueva generación		
AVIF	Formato de imagen basado en AV1. No confundir con videos AVI.	⚠️ Con precaución
JPEG-XL	Alternativa competidora a AVIF. Evolución moderna de JPEG.	⚠️ Aún poco soporte

CONQUERBLOCKS

Formatos admitidos

SVG vs PNG vs JPG vs WEBP

Formatos admitidos

Cualquier otro formato no mostrado en esta lista no está recomendado para utilizar en web. Formatos como BMP, TIFF, RAW, NEF, PSD, CDR son formatos no orientados para su uso en la web, además de ser propietarios y muy pesados. Asegúrate utilizar uno de los recomendados de la tabla anterior.

Imágenes responsive y alternativas

Imágenes responsive

- En algunas ocasiones nos podría interesar indicar al navegador que debe descargarse una imagen concreta si el dispositivo tiene una cierta resolución (por ejemplo, dispositivo móvil), pero en el caso de ser un dispositivo de alta resolución (por ejemplo, escritorio) descargue una imagen de mayor resolución. De esta forma no desperdiciamos recursos o tiempo de descarga. Lo mismo con pantallas que tengan una densidad de píxeles mayor, y pueda mostrar imágenes a mejores resoluciones.

Imágenes responsive

Atributos

Atributo	Descripción
srcset	Indica un conjunto de imágenes y cargará una cuando se cumplan ciertos criterios.
sizes	Indica un conjunto de condiciones de medios para aplicar al srcset.

Imágenes responsive

Atributo srcset

Mediante el atributo srcset se puede indicar una lista de recursos (imágenes) separados por coma, que se utilizarán en situaciones concretas cuando se cumplan determinados criterios como los indicados anteriormente. Los valores de srcset tendrían el siguiente formato:

```
imagen_400w.jpg 400w, imagen_900w.jpg 900w, imagen_1200w.jpg 1200w
```

Imágenes responsive

Atributo srcset

Criterio	Descripción	Ejemplos
Ancho	Ancho de pantalla en píxeles, seguido de la letra w.	400w, 800w
Densidad	Densidad de píxeles con decimales, seguido de la letra x.	3x, 2.5x

Imágenes responsive

```
<img src="imagen_400w.jpg" alt="Logotipo HTML5"
     srcset="imagen_400w.jpg 400w, imagen_900w.jpg 900w, imagen_1200w.jpg 1200w">
```

Picture y Source

Picture

- HTML 5.1 incorpora un nuevo sistema para utilizar imágenes en nuestros documentos HTML de forma mucho más flexible que la etiqueta <img> tradicional. Esta nueva etiqueta nos permitirá mostrar imágenes dependiendo de nuestras necesidades.

Picture

Etiqueta	Descripción
`<picture>`	Agrupa una serie de imágenes. Etiqueta contenedora.
`<source>`	Mostrará la imagen que cumpla una serie de criterios opcionales.

Picture

Formato	Características	¿Recomendado?
Formatos tradicionales		
PNG	Soporta transparencia. Compresión sin pérdidas. Imágenes «lisas».	✅
JPG	Compresión con pérdidas. Ideal para imágenes con texturas.	✅
SVG	Formato vectorial. Ideal para imágenes escalables.	✅
GIF	Imágenes pequeñas. Soporta animación. Pesado. Poca calidad.	⚠️ Evitar. Usar MP4 o APNG.
Formatos modernos		
WEBP	Alternativa libre de Google al JPEG, PNG y GIF. Transparencias y animaciones.	✅ Buen soporte
JPEG2000	Formato evolución de JPEG. Poco utilizado.	❌ Solo Safari
JPEG-XR	Alternativa libre de Microsoft al JPEG. Poco utilizado.	❌ Solo IE
APNG	Alternativa libre a GIF. Compatible con PNG. Soporta animaciones.	✅ Buen soporte
Formatos de nueva generación		
AVIF	Formato de imagen basado en AV1. No confundir con videos AVI.	⚠️ Con precaución
JPEG-XL	Alternativa competidora a AVIF. Evolución moderna de JPEG.	⚠️ Aún poco soporte

Picture

- Uno de los principales problemas que teníamos al utilizar la etiqueta <img> es que si utilizábamos un formato de imagen moderno, que puede no estar soportado en algunos navegadores, la imagen no se verá en dichos navegadores.
- Nos falta la posibilidad de establecer alternativas, para que en el caso de estar soportado pueda utilizarla, y en caso contrario utilice un formato alternativo más antiguo, y así poder aprovecharnos de las ventajas de formatos nuevos si hay soporte (menor peso, mayor calidad, etc...).

Picture

```
<picture>
  <!— Aquí colocaremos posibles imágenes candidatas <source> —>
  <!— Aquí colocaremos la imagen <img> a utilizar por defecto —>
</picture>
```

Picture

```
<picture>
  <source srcset="image.avif">
  <source srcset="image.webp">
  <img src="image.jpg" alt="Descripción breve de la imagen">
</picture>
```


Picture

- El navegador comprobará la primera etiqueta <source> con una imagen en formato .avif.
- Si el navegador la soporta, la insertará en el atributo src de la etiqueta <img> y la usará.
- Si el navegador no la soporta, analizará la siguiente etiqueta <source> con formato .webp.
- Si el navegador la soporta, la insertará en el atributo src de la etiqueta <img> y la usará.
- Si el navegador no la soporta, utilizará la etiqueta <img> indicada al final.

Imágenes responsive II

Imágenes responsive II

- Otra ventaja interesante es que con <picture> podemos crear imágenes responsive que cambien dependiendo de características de las media queries (CSS). Por ejemplo, utilicemos min-width (tamaño mínimo de ancho de la pantalla) en el siguiente ejemplo:

Imágenes responsive II

```html
<picture>
  <source media="(width ≥ 900px)"
          srcset="logo-large.png">
  <source media="(width ≥ 600px) and (width < 900px)"
          srcset="logo-medium.png">
  <source media="(width < 600px)"
          srcset="logo-small.png">
  <img src="logo-medium.png" alt="HTML5 logo">
</picture>
```


Picture

Según tamaño de pantalla

- Dispositivos grandes (igual o más de 900px): Muestra la imagen logo-large.png
- Dispositivos grandes (entre 600-899px): Muestra la imagen logo-medium.png
- Dispositivos pequeños (menos de 600px): Muestra la imagen logo-small.png
- Si no cumple las anteriores (o no soporta estas etiquetas): Muestra la imagen logo-medium.png

Picture

```
<picture>
  <source media="(width ≥ 900px)"
          srcset="logo-large.png,
                  logo-large-hd.png 2x,
                  logo-large-fhd.png 3x">
  <source media="(width ≥ 600px) and (width < 900px)"
          srcset="logo-medium.png,
                  logo-medium-hd.png 2x,
                  logo-medium-fhd.png 3x">
  <source media="(width < 600px)"
          srcset="logo-small.png,
                  logo-small-hd.png 2x,
                  logo-small-fhd.png 3x">
  <img src="logo-medium.png"
       srcset="logo-medium.png,
               logo-medium-hd.png 2x,
               logo-medium-fhd.png 3x"
       alt="HTML5 logo">
</picture>
```


Picture

Según densidad de píxeles

- Observa que en cada etiqueta <source> se establece el atributo srcset para indicar una imagen para una determinada densidad de pantalla (*por ejemplo, 2x o 3x). Si se omite y no se indica ninguna, es lo mismo que indicar 1x.

Imágenes responsive II

CONQUERBLOCKS

‹Despedida›

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

<HTML>

Clase 14 - Etiquetas multimedia

CONQUERBLOCKS

<Índice>

Clase 14: Etiquetas multimedia

Etiquetas de contenido incrustado

Iframe

Embed

Etiqueta video

Etiqueta audio

Etiqueta canvas

Etiquetas map y area

CONQUERBLOCKS

Etiquetas de contenido incrustado

Etiquetas de contenido incrustado

- Normalmente se usarán para cargar contenido externo
- También se puede cargar contenido de la misma web
- Es una referencia a un recurso almacenado en otra ubicación
- Al estar enlazado, garantiza que si el original cambia, nuestra web mostrará la versión actualizada

Etiquetas de contenido incrustado

- Podemos incrustar contenido como objetos multimedia, bloques de código HTML con contenidos diseñados por terceros e incluso webs enteras

- Usaremos dos tipos de etiquetas:
  - <embed>
  - <iframe>

<embed>

<embed>

Podemos incrustar casi cualquier tipo de recurso

- Documentos html => mejor usa iframe
- Imágenes => mejor usa img
- Vídeos => mejor usa video
- Audio => mejor usa audio
- Flash
- Controles ActiveX

<embed>

Atributos

<embed>

Demo

```
<embed type="video/webm" src="trailers/estreno.mov" width="640" height="480">
```

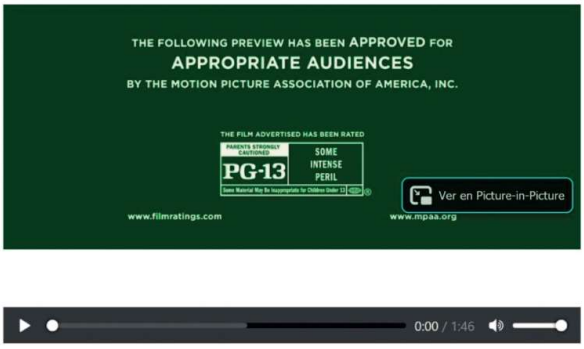


Figura 2.38. *Los vídeos son recursos multimedia muy útiles y fáciles de utilizar.*

<embed>

¿Os digo un secreto?

<embed>

W3c

CONQUERBLOCKS

iframe

CONQUERBLOCKS

iframe

- Representa un contexto de navegación anidado. Es una ventana que permite acceder a un recurso que se encuentra ubicado en un contexto distinto a la página que se está mostrando en el navegador.

- Básicamente, permite incrustar un documento o parte de un documento dentro de una página HTML.

iframe

- src: la URI del recurso incrustado.

- sredoc: contiene el código HTML que mostrar dentro del elemento <iframe>. Si existe este atributo, se impone a la referencia proporcionada por el atributo sr.

- sandbox: permite proporcionar restricciones al documento incrustado en el elemento <iframe> con objeto de mejorar la seguridad de la página.

- allow: directivas referentes a permisos de acceso a determinados elementos del dispositivo físico, tales como el acelerometro, la cámara o la geolocalización.

iframe

- allowfullscreen: si existe, permite al contenido incrustado ocupar la pantalla com- pleta.
- width y height: permiten indicar el ancho y el alto en píxeles.
- loading: con el valor lazy el contenido se carga cuando este se haga visible; con el valor eager (valor por defecto) el contenido se cargará inmediatamente.

iframe

¿Pero qué ocurre si queremos colocar contenido de una web externa? Por ejemplo, servicios como **Youtube, Vimeo, SoundCloud, SlideShare** u otros sitios similares. Se trata de servicios que ofrecen contenido externo para incrustar en nuestra página web, pero lo que nos proporcionan no son imágenes, video o audio en formatos que reconozca el navegador, sino un enlace con contenido HTML muy diverso, que pueden incluir Javascript, CSS, imágenes, videos y muchos otros recursos.

iframe

Demo iframe externo

Content-Security-Policy

iframe

Demo iframe interno

iframe

Demo iframe Youtube

iframe

Demo iframe Google Maps

iframe

El atributo name

Es posible utilizar el atributo name en nuestro <iframe> para darle un nombre al marco y cargar direcciones URL diferentes a través de una etiqueta de enlace <a>, haciendo referencia al marco a través del atributo target:

```
<a href="https://www.bing.com/" target="marco">Abrir buscador Bing</a>

<iframe name="marco" width="100%" height="450"
        src="https://www.youtube.com/embed/Imeq3GeRttw">
</iframe>
```


iframe

El atributo srcdoc

Por otro lado, el atributo srcdoc nos permite indicar directamente el código fuente del iframe en su valor, algo que puede ser interesante para crear contenido autogenerado a demanda:

```
<iframe src="http://pagina.com/apartado/"
        srcdoc="<p>¡Hola! ¡Soy el <strong>texto del iframe</strong>!</p>">
</iframe>
```

iframe

W3c

CONQUERBLOCKS

‹video›

CONQUERBLOCKS

‹video›

- En HTML5 se introduce la interesante posibilidad de mostrar videos directamente desde nuestro navegador. De hecho, si arrastramos un video a la ventana del navegador, comprobaremos que comienza a reproducirse en él. Esto ocurre porque los navegadores son capaces de mostrar imágenes, reproducir video y audio.

‹video›

Atributos básicos

Atributo	Descripción
`src`	Archivo o URL del video a reproducir. Obligatoria si no tiene etiquetas dentro.
`width`	Indica el tamaño de ancho del video.
`height`	Indica el tamaño de alto del video.

‹video›

Igual que con las imágenes, es recomendable establecer ancho y alto por reserva de espacio

‹video›

Atributos de control

Atributo	Descripción
`autoplay`	Comienza a reproducir el video automáticamente. OJO, está sujeto a normas del navegador.
`loop`	Vuelve a iniciar el video cuando finaliza su reproducción (en bucle).
`muted`	Establece que el video se reproduzca con sonido muteado (silenciado).
`controls`	Muestra los controles de reproducción (por defecto no se muestran).
`playsinline`	Reproduce el video **en línea**, es decir, en su área de reproducción.

‹video›

Autoplay con grandes limitaciones

‹video›

Atributos de precarga

Atributo	Descripción
poster	Muestra una imagen a modo de presentación.
preload	Indica como realizar la precarga del video.

<video>

Atributos

Formato	Códecs utilizado	Características	¿Recomendado?
Formatos modernos			
MP4	x264, DivX H264	Alta calidad. El códec x264 es libre.	✅ Buen soporte
WebM	VP8, VP9	Alternativa libre a MP4 de Google.	✅ Buen soporte
Formatos de nueva generación			
AV1	Basado en VP10, Daala y Thor	Compite con HEVC/H.265	❌ Soporte irregular
HEVC	x265, DivX HEVC	Futura evolución de MP4.	❌ Soporte irregular
Formatos más antiguos o no orientados a web			
OGV	Theora	Alternativa libre a MP4.	⚠️ Sin soporte en Safari
MKV	Matroska	Buena compresión. Potente.	❌ Alto consumo CPU
AVI	XviD, DivX 3/5	Menor compresión que MP4.	❌ Anticuado

<video>

Uso de source

```
<video width="640" height="480">
  <source src="video.mp4" type="video/mp4" />
  <source src="video.webm" type="video/webm" />
  <source src="video.ogv" type="video/ogg" />
  <img src="imagen.png" alt="Video no soportado" />
  Su navegador no soporta contenido multimedia.
</video>
```

iframe

W3c

iframe

Demo

‹audio›

CONQUERBLOCKS

‹audio›

Como la etiqueta video, pero sin video :-)

CONQUERBLOCKS

‹audio›

Formatos

Formato	Codec utilizado	Características	¿Recomendado?
MP3	MPEG Layer-3	Buena calidad.	✅ Buen soporte
AAC	Advanced Audio Coding	Mejora el MP3. Usado como audio en MP4.	✅ Buen soporte
OGG	Ogg Vorbis	Buena calidad. Alternativa libre a MP3.	⚠️ Cuidado con Safari
Opus	Opus	Buena calidad. Alternativa libre a MP3.	⚠️ Cuidado con Safari
FLAC	FLAC Audio Lossless	Muy buena calidad (sin pérdidas). Tamaño superior.	✅ Buen soporte.
WAV	Wave sound	Formato de Microsoft. Está soportado.	❌ Muy pesado

‹audio›

Uso de source

```
<audio>
  <source src="audio.opus">
  <source src="audio.ogg">
  <source src="audio.mp3">
</audio>
```

<audio>

W3c

<audio>

Demo

CONQUERBLOCKS

<canvas>

<canvas>

Se utilizará para que librerías gráficas usen este contenedor a modo de lienzo para poder dibujar y montar gráficos

CONQUERBLOCKS

<canvas>

W3c

CONQUERBLOCKS

<map> y <area>

CONQUERBLOCKS

‹map›

Se utilizan para realizar secciones de una imagen que puedan enlazar a diferentes secciones

CONQUERBLOCKS

‹map›

W3c

CONQUERBLOCKS

<Despedida>

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

<HTML>

Clase 15 - Etiquetas interactivas

CONQUERBLOCKS

<Índice>

Clase 15: Etiquetas interactivas

Etiquetas interactivas

Cuidado con usarlas sin sentido

Ocultación semántica

Etiqueta details y summary

Etiqueta dialog

Etiqueta selec2

Etiqueta Wysiwyg

CONQUERBLOCKS

Etiquetas interactivas

Etiquetas interactivas

- Serán un conjunto de etiquetas que sirven para que el usuario puede interactuar con nuestro contenido
- Con ellas se podrá desencadenar algunas acciones en la web
- Vienen a suplir, características que únicamente podíamos hacer anteriormente con JS
- Todavía están bastante verdes, pero poco a poco mejoran

Etiquetas de contenido incrustado

- El ejemplo más claro de etiquetas interactivas pueden ser todas las de formulario
- Select, checkbox, radiobutton etc...
- Atributos requeridos, comprobación de datos etc...
- Su funcionamiento es nativo del navegador

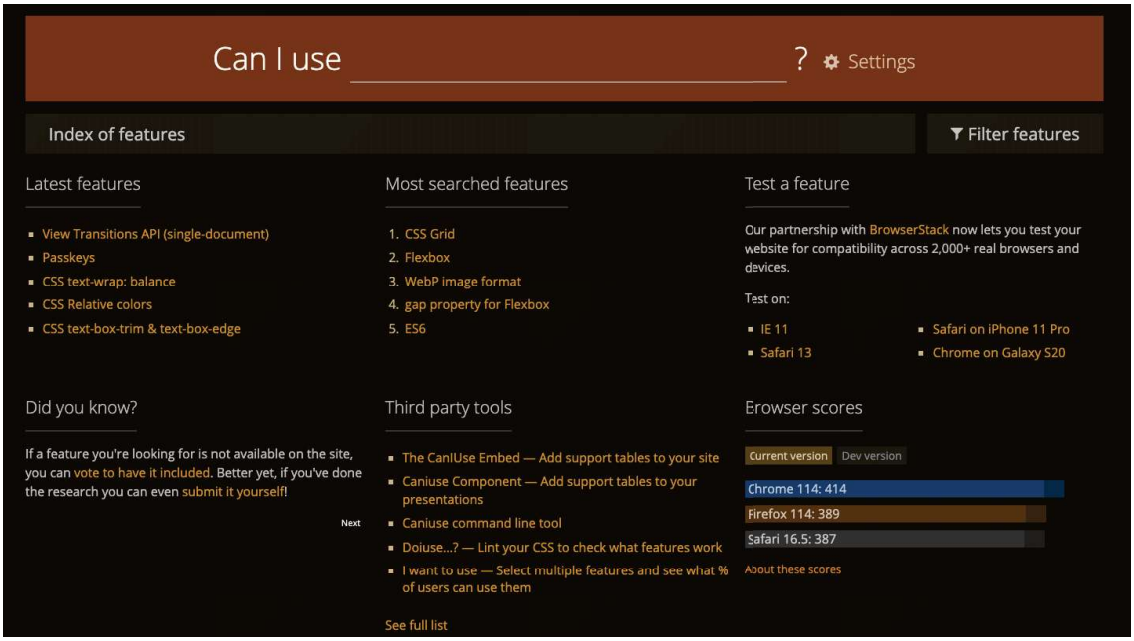
Cuidado con usarlas sin sentido

Can I use

- Ya vimos que antes de usar alguna etiqueta nueva, debemos comprobar su compatibilidad para asegurarnos que tiene el mayor soporte posible

- Nuestro mayor amigo: Can I use

Can I use



Can I use

Demo

Ocultación semántica

Ocultación semántica

- En muchas ocasiones, cuando queremos ocultar un elemento lo hacemos con CSS utilizando las propiedades display: none, visibility: hidden, opacity: 0 o alguna similar.

- Lo ideal sería tener una forma de saber si semánticamente tiene sentido que un elemento esté oculto, por ejemplo, por que estamos enfocados en otra sección, y la sección oculta no es relevante en este momento.

Ocultación semántica

- Con el atributo hidden hacemos desaparecer visualmente un elemento HTML (y todo su contenido, si está en una etiqueta contenedora).

- Hasta ahora, para realizar esta tarea teníamos que recurrir a CSS, ya que se trata de una cuestión de presentación visual.

Ocultación semántica

```
<section>
  <h1>Sección 1</h1>
  <p>En esta sección vamos a hablar de...</p>
</section>

<section hidden>
  <h1>Sección 2</h1>
  <p>En esta sección vamos a hablar de...</p>
</section>
```


Ocultación semántica

El atributo hidden no tiene ninguna magia extraña. Simplemente se trata de un atributo, que cuando un elemento lo incorpore, el mismo tendrá un display: none por defecto. Sin embargo, nosotros podemos cambiar el estilo mediante CSS si lo deseamos.

Ocultación semántica

Demo hidden

Details y summnary

Details y summary

- En HTML5, aparecen algunas nuevas etiquetas que proporcionan características interactivas a una página o documento. En dichas etiquetas, el usuario debe tomar parte de forma activa para activarlas o utilizarlas.

- Una de ellas es el llamado Acordeón HTML o elemento desplegable. Se trata de un elemento que muestra un texto y forma o icono identificativo que el usuario puede pulsar y desplegar una información que por defecto está contraída.

Details y summary

Details

- La etiqueta <details> nos proporciona la posibilidad de crear un elemento desplegable, que el usuario puede expandir o contraer para ver más información que en principio estará oculta

- Se trata de una etiqueta contenedora que actúa como un , salvo que todo el contenido de esta etiqueta no se muestra inicialmente.

Details

Demo

CONQUERBLOCKS

Summary

Podremos personalizar el texto Detalles que sale por defecto a la hora de usar una etiqueta summary

CONQUERBLOCKS

Summary

Demo

Details y summary

ejemplos

Can I use

Dialog y Popover

Dialog y Popover

En HTML5 es posible crear ventanas de diálogo personalizadas sin necesidad de Javascript (o mediante muy poco Javascript), de forma nativa, e incluso crear ventanas modales. Todo ello, se realizará mediante la etiqueta HTML <dialog> en diferentes modalidades:

Dialog y Popover

- **Ventana de diálogo**: Se denomina así a una ventana que comunica el sistema o máquina con el usuario, para anunciarle información y pedirle confirmación, para pedirle datos, etc.

- **Ventana de diálogo modal**: Se denomina así a una variación de la anterior, donde se obliga al usuario a atenderla y responderla, impidiendo hacer ninguna otra tarea mientras.

Dialog y Popover

Dialog

A partir de HTML5.1, se incorpora una etiqueta <dialog> mediante la cuál podemos construir nuestras propias ventanas de diálogo personalizadas. Dichas ventanas pueden contener toda la información HTML que queramos: texto, imágenes, botones, video o cualquier otra cosa, y ser personalizada mediante CSS.

Dialog y Popover

```
<dialog open>
  <p>Esto es un ejemplo de mensaje de diálogo.</p>
</dialog>
```

Demo

Dialog y Popover

Demo, cerrando dialog

```
<dialog id="alert-dialog">
  <img class="icon" src="information.png" alt="Information icon">
  <p>Esto es un ejemplo de mensaje de diálogo.</p>
  <button onClick="this.parentElement.close()">Aceptar</button>
</dialog>
```

Dialog y Popover

Dialog modal

- Por defecto, el fondo se oscurece un poco.
- La ventana se suele centrar también verticalmente, para centrar la atención en ella.
- Sólo puede haber una ventana modal, ventanas normales pueden existir varias.
- La más importante: No deja interactuar con ningún otro elemento de la página.

Dialog y Popover

Popovers: elementos emergentes

Los popover (pop over, salta por encima) son una nueva forma en HTML5 de crear elementos emergentes, es decir, contenido que aparecerá al usuario por encima del contenido que ya estaba visualizando, para así llamar temporalmente su atención.

Popovers: elementos emergentes

Atributo	Descripción
popover	Convierte el elemento HTML en un potencial contenido emergente. Por defecto, valor auto.
popovertarget	Indica el id del elemento que queremos manipular.
popovertargetaction	Indica la acción que se realizará sobre el elemento con el id anterior.

Popovers demo

```
<button popovertarget="message" popovertargetaction="toggle">Mostrar</button>

<div id="message" popover>
  <p><strong>¡Alerta!</strong>: Has recibido un correo de <strong>Afordin</strong>.</p>
</div>
```

Datalist

Datalist

Aunque hemos visto controles como las casillas de verificación, los botones de opción (radio), o las listas de selección hay situaciones en las que puede que ninguno de estos controles nos sirvan. Por ejemplo, una opción de varias posibles, pero también abierta a que se añadan más opciones por parte del usuario. Esto sería una mezcla entre un campo <input> de texto y una lista desplegable <select>.

Datalist

La etiqueta <datalist> debe indicarse con un atributo id para identificarse. En su interior, incluiremos varias etiquetas <option> para añadir opciones, al igual que lo hacemos en una etiqueta <select>:

Datalist

```
<datalist id="items">
  <option value="1">Opción 1</option>
  <option value="2">Opción 3</option>
</datalist>
```

Datalist

```
<form method="post" action="/send/">
  <input type="text" list="items">
  <datalist id="items">
    <option value="manzdev">ManzDev</option>
    <option value="afor_digital">Afor_digital</option>
    <option value="padawanstrainer">PadawansTrainer</option>
    <option value="chrisvdev">ChrisvDev</option>
  </datalist>
</form>
```

CONQUERBLOCKS

Datalist

```
<form method="post" action="/send/">
  <input type="text" list="items">
  <datalist id="items">
    <option value="manzdev">ManzDev</option>
    <option value="afor_digital">Afor_digital</option>
    <option value="padawanstrainer">PadawansTrainer</option>
    <option value="chrisvdev">ChrisvDev</option>
  </datalist>
</form>
```

CONQUERBLOCKS

Datalist

ejemplos

Can I use

CONQUERBLOCKS

Select2

CONQUERBLOCKS

Select2

Se utilizará para tener un select con valores predecibles y con ayuda contextual según se vayan rellenando

Select2

Demo

Wysiwyg

CONQUERBLOCKS

Wysiwyg

Editor de contenido html en línea para nuestros Textarea con formato

CONQUERBLOCKS

Wysiwyg

CKEditor

CONQUERBLOCKS

‹Despedida›

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS