

BIENVENIDOS AL CURSO

(IFCDO110) CONFECCIÓN Y PUBLICACIÓN DE PÁGINAS WEB



22/4963

CONFECCIÓN Y PUBLICACIÓN DE PÁGINAS WEB

**CENTRO DE FORMACIÓN:
TEIDE IV**

SEPE
SERVICIO PÚBLICO
DE EMPLEO ESTADAL

**CONSEJERÍA DE ECONOMÍA,
EMPLEO Y HACIENDA**
Comunidad de Madrid

 GOBIERNO
DE ESPAÑA

**MINISTERIO
DE EMPLEO
Y SEGURIDAD SOCIAL**

 UNIÓN EUROPEA
Fondo Social Europeo
En Fase: Invierte en tu futuro

Presentación Escuela

CENTRO DE FORMACIÓN: TEIDE IV

Tres centros educativos en Madrid dedicados a la Formación Profesional en sus dos subsistemas de Formación Reglada:

- Ciclos Formativos de Formación Profesional Básica, Grado Medio y Superior
- y Formación para el Empleo

COORDINADORA RESPONSABLE:
Andrea Sánchez Cebrián
Dpto. de Coordinación para el Empleo



EXPERIENCIA CURRICULAR DOCENTE

Mar Moreno Álvarez



(IFCDO110) CONFECCIÓN Y PUBLICACIÓN DE PÁGINAS WEB

- Diplomada en Informática
- Diplomada en Informática Hospitalaria
- Diplomada en Metodología general para la gestión de proyectos
- Técnico Superior en Informática de Gestión
- Estudiante del Grado en Criminología

- Certificado de Profesionalidad de Docencia para la FPE
- Diploma SSCE075PO-Formación de formadores en E-Learning
- Diploma SSCE071PO-Experto en E-Learning.
- Diploma SSCE123PO- Orientación socio-laboral On Line
- Diploma SSCE001PO TELEFORMACIÓN PARA DOCENTES
- Diploma SSCE009PO - HERRAMIENTAS DIGITALES EN EL AULA
- Diploma Creación de Blogs y Redes Sociales
- Experto Universitario en Criminología
- Experto Docente en Herramientas de Análisis de Datos, Auditoría Interna y Seguridad
- Certificado de Profesionalidad Implantación de espacios comerciales. Escaparatismo
- Diploma Dirección Comercial y Marketing. Selección y Formación de Equipos

Otros:

- Emprendimiento: Cómo Gestionar el desarrollo tecnológico de una Startup.
- Técnicas de Análisis de Datos y Big Data.
- Certificado ITIL Foundation Certificate in IT Service Management
- Certificado VTSP – Mobility 2016 Accreditation VMware Partner University
- Certificado VSP – Mobility 2016 Accreditation VMware Partner University
- Certificado Poténciate con Redes sociales
- Ingeniería de Servicios
- Comunicación Corporativa

LinkedIn Learning

LinkedIn para la búsqueda de empleo

Curso completado por Mar Moreno
ago 31, 2023 at 05:30PM UTC • 1 hora 25 minutos

Aptitudes que se tratan

Búsqueda de empleo LinkedIn

Mar Moreno Álvarez
Responsable de estrategia de contenido de Learning

El contenido de este curso es propiedad de LinkedIn y se proporciona únicamente para uso personal.



UNED

URJC online



EQUIPOS DE TRABAJO

Mar Moreno Alvarez



(IFCDO110) CONFECCIÓN Y PUBLICACIÓN DE PÁGINAS WEB

USUARIO EQUIPOS

USER: tarde

PASSWORD: Venom

CERTIFICADO DE PROFESIONALIDAD

Fecha Curso: 17/04/2024 – 05/10/2024



Familia profesional: **INFORMÁTICA Y COMUNICACIONES**

Área profesional: **Desarrollo**



FICHA DE CERTIFICADO DE PROFESIONALIDAD				
(IFCDO110) CONFECCIÓN Y PUBLICACIÓN DE PÁGINAS WEB (RD 1531/2011, de 31 de octubre modificado por el RD 628/2013, de 2 de agosto)				
COMPETENCIA GENERAL: Crear y publicar páginas web que integren textos, imágenes y otros elementos, utilizando lenguajes de marcas y editores apropiados, según especificaciones y condiciones de "usabilidad" dadas y realizar los procedimientos de instalación y verificación de las mismas en el servidor correspondiente.				
NIV.	Cualificación profesional de referencia	Unidades de competencia		Ocupaciones o puestos de trabajo relacionados:
2	IFC297_2 CONFECCIÓN Y PUBLICACIÓN DE PÁGINAS WEB (RD 1201/2007, de 14 de septiembre)	UC0950_2	Construir páginas web	<ul style="list-style-type: none">Desarrollador de páginas webMantenedor de páginas web
		UC0951_2	Integrar componentes software en páginas web	
		UC0952_2	Publicar páginas web	

Correspondencia con el Catálogo Modular de Formación Profesional				
H. Q	Módulos certificado	H. CP	Unidades formativas	Horas
210	MF0950_2: Construcción de páginas web	210	UF1302: Creación de páginas web con el lenguaje de marcas	80
			UF1303: Elaboración de hojas de estilo	70
			UF1304: Elaboración de plantillas y formularios	60
210	MF0951_2: Integración de componentes software en páginas web	180	UF1305: Programación con lenguajes de guión en páginas web	90
			UF1306: Pruebas de funcionalidades y optimización de páginas web	90
90	MF0952_2: Publicación de páginas web	90		90
	MP0278: Módulo de prácticas profesionales no laborales	80		
510	Duración horas totales certificado de profesionalidad	560	Duración horas módulos formativos	480

MF0950_2 Construcción de páginas Web



- UF1302 Construcción de páginas web con el lenguaje de marcas
- UF1303 Elaboración de hojas de estilo
- UF1304 Elaboración de plantillas y formularios

CSS (Cascading Style Sheets) - Introducción

¿Qué es?

CSS (Cascading Style Sheets) es un lenguaje de diseño utilizado para describir la presentación de los elementos HTML.

Es utilizado para establecer el color, el tamaño, la posición, la fuente, entre otras propiedades de los elementos en una página web. Es esencial para darle estilo y diseño a una página web.

Con CSS, los desarrolladores web pueden separar la estructura de una página web (definida con HTML) de su presentación visual. Esto permite que los diseños sean reutilizados y modificados fácilmente en varias páginas web, lo que agiliza el proceso de desarrollo y facilita el mantenimiento.

CSS también permite la creación de diseños responsivos, adaptando el diseño de una página web a diferentes tamaños de pantalla, lo que permite una mejor experiencia de usuario en dispositivos móviles.

CSS (Cascading Style Sheets) - Introducción

¿Qué es CSS?

Hojas de Estilo en Cascada

**CSS son las siglas en inglés para
«hojas de estilo en cascada»
(Cascading Style Sheets).**

Básicamente, es un lenguaje que maneja el diseño y presentación de las páginas web, es decir, cómo lucen cuando un usuario las visita. Funciona junto con el lenguaje HTML que se encarga del contenido básico de los sitios.



CSS (Cascading Style Sheets) - Introducción

¿Para qué se usa?

Tiene varios usos importantes en el desarrollo web:

Separar el contenido y el diseño: CSS permite separar el contenido de una página web (definido con HTML) de su presentación visual, permitiendo que los diseños sean reutilizados y modificados fácilmente en varias páginas web.

Crear diseños adaptativos: CSS permite crear diseños adaptativos, adaptando el diseño de una página web a diferentes tamaños de pantalla, lo que permite una mejor experiencia de usuario en dispositivos móviles y diferentes tamaños de pantalla.

Mejorar la accesibilidad: CSS permite mejorar la accesibilidad de una página web para personas con discapacidades, especificando por ejemplo tamaños de letra más grande, colores de fondo y letra de contraste, entre otros.

CSS (Cascading Style Sheets) - Introducción

Para qué sirve CSS

Hojas de Estilo en Cascada

Con CSS, puedes crear reglas para decirle a tu sitio web cómo quieres mostrar la información y guardar los comandos para elementos de estilo (como fuentes, colores, tamaños, etc.) separados de los que configuran el contenido.

Además, puedes crear formatos específicos útiles para comunicar tus ideas y producir experiencias más agradables, en el aspecto visual, para los usuarios del sitio web.

CSS (Cascading Style Sheets) - Introducción

¿Para qué se usa? (2)

Mejorar el SEO: CSS permite utilizar etiquetas y atributos específicos para mejorar el SEO de una página y hacer que sea más fácilmente encontrada por los motores de búsqueda.

Mejorar la velocidad de carga: Al utilizar CSS es posible reducir el uso de imágenes y reducir el tamaño del código HTML, lo que ayuda a mejorar la velocidad de carga de una página web.

En general, CSS es una herramienta esencial para darle estilo y diseño a una página web, permitiendo separar la estructura de su presentación visual y crear diseños adaptativos y responsivos

CSS (Cascading Style Sheets) - Introducción



...y la cascada? ¿Qué significa?

Comprender el algoritmo en cascada te ayuda a comprender en cómo el navegador resuelve conflictos como este.

El algoritmo en cascada se divide en 4 etapas distintas.

- 1. Posición y orden de aparición**
- 2. Especificidad**
- 3. Origen**
- 4. Importancia**

CSS (Cascading Style Sheets) - Introducción

...y la cascada? ¿Qué significa?

Posición y orden de aparición:

El orden en el que aparecen tus reglas CSS.

Especificidad: Un algoritmo que determina qué selector de CSS tiene la coincidencia más fuerte.

Origen: El orden en el que aparece CSS y de dónde proviene, ya sea un estilo de navegador, CSS de una extensión de navegador o tu propio CSS creado.

Importancia: Algunas reglas CSS tienen más peso que otras, especialmente con el tipo de regla !important.

CSS (Cascading Style Sheets) - Introducción

Historia

CSS fue desarrollado por el **World Wide Web Consortium (W3C)** en 1996 como una forma de separar la presentación de un documento de su estructura de contenido. **El objetivo era permitir que los diseñadores pudieran crear estilos visuales para sus páginas web sin tener que modificar el código HTML de la misma.**

La primera versión de **CSS**, conocida como **CSS1**, fue publicada en 1996 y **soportaba solo un número limitado de propiedades y selectores**. Sin embargo, ya permitía hacer cosas como definir el color de fondo de un elemento, el tamaño de la fuente, el margen y el padding.

En 1998 se publicó la versión **CSS2**, que añadía nuevas propiedades y mejoraba algunas de las ya existentes. También se incluyó soporte para media queries, lo que permitía adaptar la presentación de una página web a diferentes dispositivos y resoluciones de pantalla.

En el año 2012, se lanzó **CSS3**, la cual se divide en varios módulos y se desarrolla de forma continua, esta versión incluyó nuevas características como la posibilidad de crear animaciones y transiciones mediante la modificación de las propiedades de los elementos, soporte para Grid y Flexbox, entre otras.

CSS (Cascading Style Sheets) - Introducción

Historia

En la actualidad, se está trabajando en la versión CSS4, la cual se espera incluir mejoras en la capacidad de seleccionar elementos, mejoras en el soporte para grids, mejoras en la capacidad de adaptar la presentación a dispositivos y resoluciones, entre otras características.

Cómo funciona CSS

CSS funciona mediante la aplicación de reglas de estilo a los elementos de una página web.

Estas reglas se escriben en un archivo CSS y se vinculan a una página HTML mediante un enlace en la cabecera de la página.

CSS (Cascading Style Sheets) - Introducción

Cómo funciona CSS (2)

Cada regla CSS consta de un **selector**, que especifica a qué elementos de la página se aplicará la regla, y un conjunto de declaraciones, que especifican cómo se deben presentar esos elementos.

Por ejemplo, una regla CSS podría tener un selector “h1” y una declaración “color: rojo;”, lo que haría que todos los encabezados h1 en la página sean rojos.

CSS también tiene un sistema de cascada, lo que significa que varias reglas pueden aplicarse a un mismo elemento y pueden entrar en conflicto entre sí.

En estos casos, las reglas más específicas tienen prioridad sobre las reglas más generales.

CSS (Cascading Style Sheets) - Introducción

Cómo funciona CSS (3)

Ejemplo: si una regla global establece que todos los elementos de tipo “p” deben ser azules, pero una regla específica establece que un párrafo específico debe ser rojo, el párrafo será rojo.

CSS también tiene un sistema de heredabilidad, lo que significa que los estilos de los elementos padres se heredan automáticamente a sus elementos hijos.

Esto permite a los desarrolladores ahorrar tiempo y esfuerzo al aplicar estilos a elementos similares en toda la página.

```
1 | p {  
2 |     background-color: blue;  
3 | }
```

ejemplo de código CSS que cambia el color de fondo de todos los párrafos en una página web a azul:

CSS (Cascading Style Sheets) - Introducción

Cómo funciona CSS (4)

Este código se puede incluir en un archivo externo con extensión .css y vincularlo en el HTML de la siguiente manera:

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

También se puede incluir en el HTML directamente dentro de una etiqueta «style»:

```
1 | <head>
2 |   <style>
3 |     p {
4 |       background-color: blue;
5 |     }
6 |   </style>
7 | </head>
8 |
```

CSS (Cascading Style Sheets) - Introducción

¿Qué características y actuación tiene CSS en una página Web?

CSS juega un papel fundamental en la presentación y diseño de un sitio web.

Su principal función es separar la estructura de contenido de una página (definida en HTML) de su presentación visual, lo que permite a los diseñadores y desarrolladores trabajar de manera independiente en cada aspecto del sitio web.

A través de CSS se pueden definir estilos visuales para los elementos de una página web, como el color de fondo, el tamaño de la fuente, la disposición de los elementos, entre otros.

Esto permite dar una apariencia consistente y profesional a todas las páginas del sitio web, y hacer que sea fácil de navegar y de leer para los usuarios.

CSS (Cascading Style Sheets) - Introducción

¿Qué características y actuación tiene CSS en una página Web? (2)

CSS también permite adaptar la presentación de una página web a diferentes dispositivos y resoluciones de pantalla a través de **media queries**

(permiten aplicar estilos CSS según el tipo general de un dispositivo (como impresión o pantalla) u otras características como la resolución de la pantalla o el ancho del viewport del navegador.),

lo que es esencial en la era actual en donde el acceso a la web se da desde diferentes dispositivos y pantallas de diferentes tamaños

CSS (Cascading Style Sheets) - Componentes

Componentes principales

Selectores: son los elementos HTML a los que se aplicarán las reglas CSS. Los selectores pueden ser basados en el nombre de la etiqueta, la clase, el ID, entre otros.

Propiedades: son las características visuales que se modificarán en los elementos seleccionados. Algunas propiedades comunes son color, tamaño de fuente, margen, padding, borde, entre otras.

Valores: son los valores asignados a las propiedades. Pueden ser valores numéricos, unidades de medida, nombres de colores, entre otros.

Declaraciones: son las líneas que componen una regla CSS y consisten en una propiedad y un valor separados por dos puntos.

Reglas: son conjuntos de declaraciones que se aplican a los elementos seleccionados.

CSS (Cascading Style Sheets) - Componentes

Componentes principales (2)

Archivos: son los archivos donde se guardan las reglas CSS y se vinculan en el HTML.

Media Queries: es un componente que permite adaptar la presentación de una página web a diferentes dispositivos y resoluciones de pantalla.

Grid y Flexbox: son sistemas de diseño que ayudan a crear diseños más flexibles y adaptables.

Lenguajes de preprocesamiento: (**LESS, SASS, STYLUS**) son lenguajes que proporcionan una sintaxis adicional a CSS para facilitar el desarrollo y mejorar la productividad de los desarrolladores.

Hay algunos otros componentes menores que también son importantes en CSS:

Comentarios: son líneas de código que no son interpretadas por el navegador y sirven para dejar notas o documentación dentro de un archivo CSS.

Unidades de medida: son los valores numéricos que se asignan a las propiedades y pueden ser expresadas en diferentes unidades como pixels, ems, porcentajes, entre otras.

CSS (Cascading Style Sheets) - Componentes

Componentes principales (3)

Pseudo-clases y pseudo-elementos: son selectores que permiten seleccionar elementos HTML basados en su estado o posición dentro del documento.

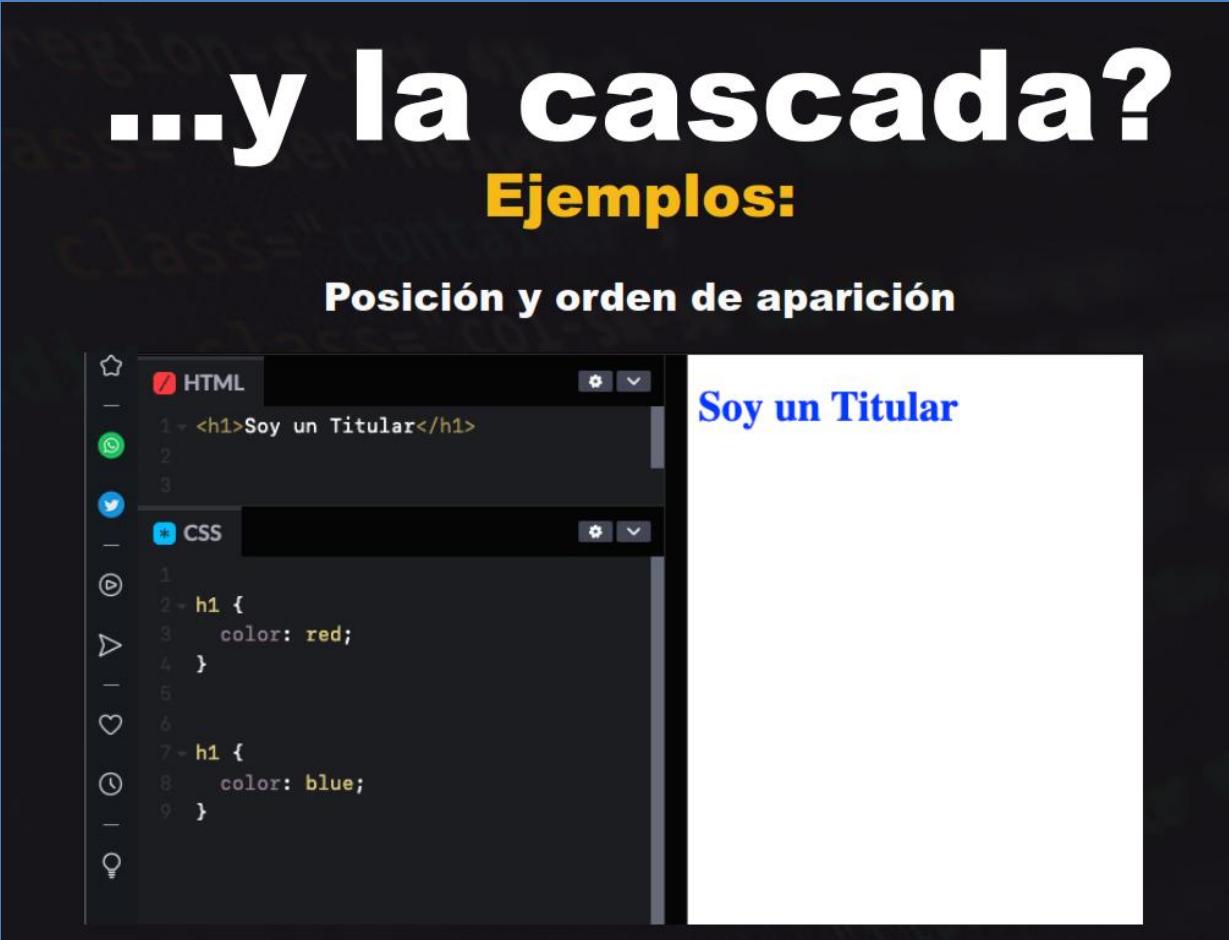
Box-model: es el sistema que utiliza CSS para calcular el tamaño y la posición de los elementos en una página web.

Herramientas de depuración: son herramientas que permiten identificar y solucionar problemas en el código CSS, tales como el inspector de elementos en los navegadores web.

CSS tiene muchos componentes que trabajan juntos para permitir que los diseñadores y desarrolladores creen estilos visuales atractivos y adaptables para los sitios web, y hacer que sea fácil de navegar y leer para los usuarios.

CSS (Cascading Style Sheets) - Componentes

Ejemplos:



...y la cascada?

Ejemplos:

Posición y orden de aparición

HTML

```
<h1>Soy un Titular</h1>
```

CSS

```
h1 {  
    color: red;  
}  
  
h1 {  
    color: blue;  
}
```

Soy un Titular

The image shows a dark-themed web page with large white text. Below it is a screenshot of a code editor. The code editor has two tabs: 'HTML' and 'CSS'. The 'HTML' tab contains the code: <h1>Soy un Titular</h1>. The 'CSS' tab contains two declarations for 'h1': one with 'color: red;' and another with 'color: blue;'. To the right of the code editor is a preview window showing the text 'Soy un Titular' in blue.

CSS (Cascading Style Sheets) - Componentes

Ejemplos

...y la cascada?

Ejemplos

Especificidad

```
HTML
1 <h1 class="my-element">Soy el
  Mismo titular</h1>
2
```

```
CSS
1 .my-element {
2   color: red;
3 }
4
5 h1 {
6   color: blue;
7 }
```

Soy el Mismo titular

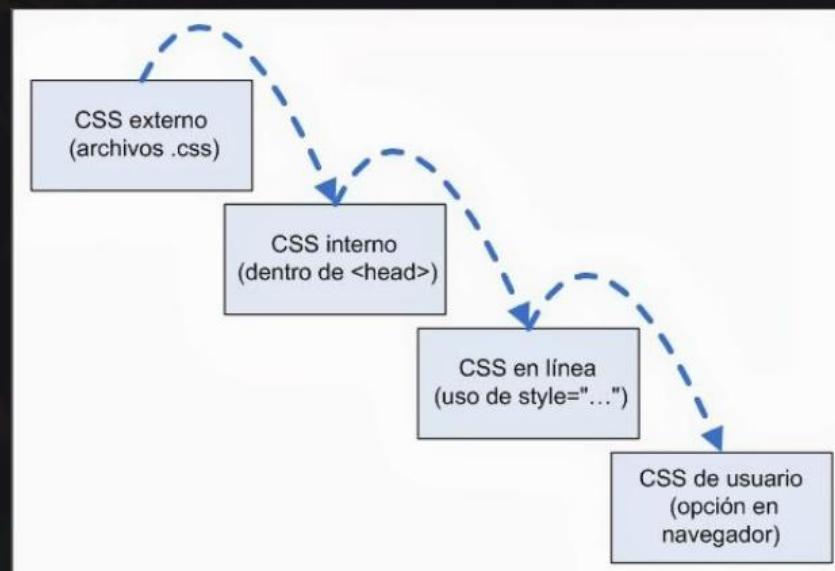
CSS (Cascading Style Sheets) - Componentes

Ejemplos

...y la cascada?

Ejemplos

Origen



CSS (Cascading Style Sheets) - Componentes

Ejemplos



The image shows a code editor interface with two tabs: 'HTML' and 'CSS'. In the 'HTML' tab, there is a single line of code:

```
3 <h1 id="my-element">Soy el Mismo  
titular</h1>
```

In the 'CSS' tab, there are three style definitions:

```
1 .my-element {  
2   color: red;  
3 }  
  
5 h1 {  
6   color: blue !important;  
7 }  
  
9 h1{  
10  color: red;  
11 }
```

To the right of the code editor, the resulting output is displayed in a large yellow font: **!Important**. Below this, the text "Soy el Mismo titular" is shown in blue, indicating the final rendered color.

CSS (Cascading Style Sheets) - Componentes



!Important “ojo con su uso”

important es una mala práctica y debería evitarse porque hace que el código sea más difícil de depurar al romper la cascada natural de las hojas de estilo.

Cuando dos declaraciones en conflicto con el ! important son aplicadas al mismo elemento, se aplicará la declaración con mayor especificidad.

CSS (Cascading Style Sheets) - Componentes

Declaración



CSS (Cascading Style Sheets) - Componentes

Cómo se declaran

INLINE

(Dentro del propio selector como un atributo)

ESTILOS INTERNOS

(Dentro del propio documento HTML)

ESTILO EXTERNO

(En un documento externo vinculado (*link*) al HTML)

CSS (Cascading Style Sheets) - Componentes

Ejemplo de 3 formas

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Ejemplo de Estilos </title>
    <link rel="stylesheet" href="css/estilos.css">

    <!-- Estilo Interno Tiene que estar dentro del head-->
    <style type="text/css">
        p{
            color: blue;
        }
    </style>

</head>

<body>
    <!-- Estilo Inline -->
    <p style="color: red">Lorem ipsum dolor sit amet consectetur adipisicing elit. Natus cumque ipsa accusamus quis, error eos ducimus nesciunt sint sed vitae rem ea maxime, quae magnam fugiat assumenda ullam unde eius?</p>
```

CSS (Cascading Style Sheets) - Componentes

**Entonces
¿Cuál es el mejor?**

SIN DUDA

ESTILO EXTERNO

**La forma recomendada de aplicar estilos CSS
Se puede aplicar a un conjunto de documentos HTML
Se importa, como hemos visto mediante la etiqueta:
(link) en el HTML**

CSS (Cascading Style Sheets) - Componentes



**El resto se considera
ANTIPATRON**

**Tenemos que tener en cuenta,
siempre que podamos la buena praxis
en nuestros proyectos.**

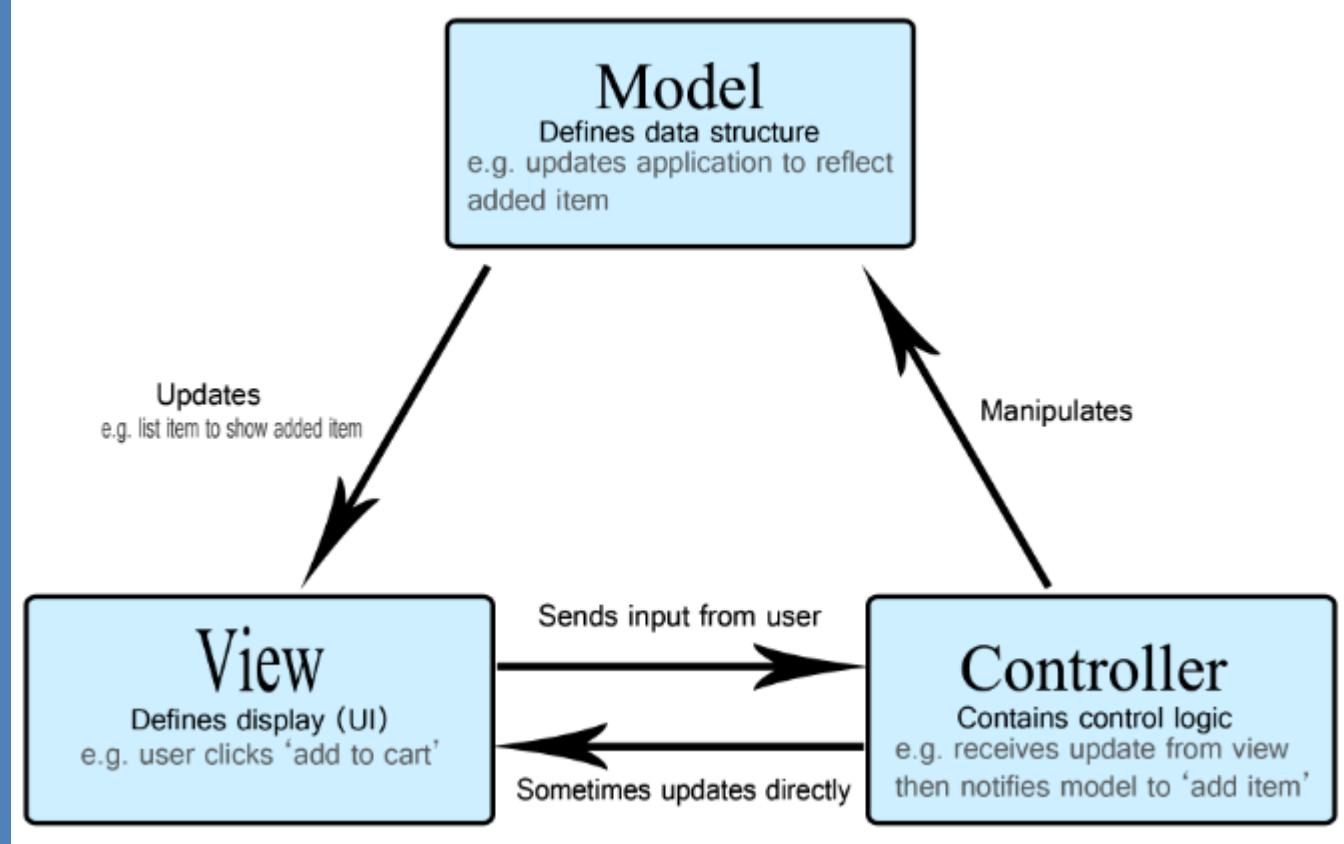
Quedaos con este concepto

Patrón de Diseño

MVC

UF1303 Elaboración de hojas de estilo

CSS (Cascading Style Sheets) - Componentes



CSS (Cascading Style Sheets) - Componentes

Selectores CSS Básicos

Los selectores definen sobre qué elementos se aplicará un conjunto de reglas CSS.

Selector de tipo

Selecciona todos los elementos que coinciden con el nombre del elemento especificado.

```
<p>Soy un párrafo</p>
```

```
p{  
    color: green;  
}
```

Todos los párrafos que hay en el documento me los colores en verde

CSS (Cascading Style Sheets) - Componentes

Selectores CSS Básicos

Selector de clase

Selecciona todos los elementos que tienen el atributo de la clase especificada.

Importante: Concepto>Clase>Reutilización

```
<p class="parrafoPepe">Soy otro párrafo Pepe</p>
```

Todos los párrafos con el nombre **parrafoPepe**, colorear en azul

```
8
9 .parrafoPepe{
10
11   color: blue;
12 }
```

CSS (Cascading Style Sheets) - Componentes

Selectores CSS Básicos

Selector de ID

Selecciona un elemento basándose en el valor de su atributo id.

Solo puede haber un elemento con un determinado ID dentro de un documento.

```
<p id="parrafoMaria"> Soy el párrafo Maria</p>
```

Todos los párrafos con el ID parrafoMaria, colorear en NARANJA

```
#parrafoMaria{  
    color: orange;  
}
```

CSS (Cascading Style Sheets) - Componentes

Selectores CSS Básicos

Pero además se pueden combinar
Se puede utilizar diferentes clases en un mismo
selector y el único ID permitible.

```
<p class="claseUno claseDos" id="idUno">Hola SOY NUEVO AQUÍ</p>
```

Aquí, dos Clases
y el (Solo UNO) ID pueden
convivir perfectamente

```
24  
25 .claseUno{  
26   background-color:orange;  
27 }  
28 .claseDos{  
29   margin-top: 30px;  
30   border: 10px solid tomato;  
31 }  
32 #idUno{  
33   font-size 22px;  
34   border: 20px dotted gold;  
35 }
```

CSS (Cascading Style Sheets) - Componentes

Analizar ejem. anterior

```
<p class="claseUno claseDos" id="idUno">Hola SOY NUEVO AQUÍ</p>

24
25 .claseUno{
26   background-color:orange;
27 }
28 .claseDos{
29   margin-top: 30px;
30   border: 10px solid tomato;
31 }
32 #idUno{
33   font-size 22px;
34   border: 20px dotted gold;
35 }
```

Colapso?

Hola SOY NUEVO AQUÍ

CSS (Cascading Style Sheets) - Componentes

...entonces

```
<p class="claseUno claseDos" id="idUno">Hola SOY NUEVO AQUÍ</p>
```

```
24  
25 .claseUno{  
26     background-color:orange;  
27 }  
28 .claseDos{  
29     margin-top: 30px;  
30     border: 10px solid tomato;  
31 }  
32 #idUno{  
33     font-size 22px;  
34     border: 20px dotted gold;  
35 }
```

Cálculo de Especificidad ID > Class

Hola SOY NUEVO AQUÍ

CSS (Cascading Style Sheets) - Componentes

Selectores CSS Básicos

Selector universal

Selecciona todos los elementos.

Opcionalmente, puede estar restringido a un espacio de nombre específico o a todos los espacios de nombres.

```
<p>Soy un párrafo</p>
<h1>Soy un H1</h1>
<h2>Y yo un H2</h2>
<h3>...pues yo un H3, más pequeño</h3>
```

Soy un párrafo

Soy un H1

Y yo un H2

...pues yo un H3, más pequeño

```
1 *{
2   color: blue;
3   background-color: lightgrey;
4   margin: 0;
5   padding: 0;
6   font-family: 'Open Sans', sans-serif;
7 }
8
9 h1{
10  font-size: 10px;
11 }
12 p{
13  font-size: 50px;
14  background-color: red;
15 }
16 }
```

El "*" es el selector Universal en CSS

Soy un párrafo

Soy un H1

Y yo un H2

...pues yo un H3, más pequeño

CSS (Cascading Style Sheets) - Componentes

Selectores CSS Básicos

Selector de atributo

Selecciona elementos basándose en el valor de un determinado atributo.

Sintaxis: [attr] [attr=value] [attr~=value] [attr|=value]
[attr^=value] [attr\$=value] [attr*=value]

**... de momento vale
:-)**

CSS (Cascading Style Sheets) - Componentes

+ Selectores CSS

Tipo/Clases/ID



Selectores

Descendientes

Podemos, de forma aún más específica seleccionar elementos en base a su relación, como por ejemplo, unos que estén dentro de otros.

Y pueden estar formados por selector de tipo, de clase, ID o por una combinación de todos ello.

CSS (Cascading Style Sheets) - Componentes

Selectores Descendientes

```
<div>
    <p>Soy un parrafo de DIV</p>
</div>

<section>
    <p>Soy un parrafo de sección</p>
</section>
```

Soy un parrafo de DIV

Soy un parrafo de sección

```
div p{
    background-color: red;
}

p{
    color: blue;
}
```

CSS (Cascading Style Sheets) - Componentes

Selectores Descendientes

Fórmula de leer estos selectores

section espacio div espacio p{
}

Búscame todos los párrafos que esten dentro de un div que a su vez esten dentro de un secction

CSS (Cascading Style Sheets) - Componentes

Otro Ejemplo

Fórmula de leer estos selectores



.class **espacio** **div** **espacio** **p{**
}

Búscame todos los párrafos que esten dentro de un div y que el párrafo tenga clase(nombreClase)



CSS (Cascading Style Sheets) - Componentes

**Selectores CSS
Pseudoclases y
pseudoelementos**

Escenarios Especiales



CSS (Cascading Style Sheets) - Componentes

Pseudoclases

Ya hemos visto cómo seleccionar por **TIPO/CLASE/ID**.
¿Pero cómo seleccionamos por su
estado, posición y relación?

Por ejemplo.

Ponemos el cursor encima de un enlace (estado)
o tiene una posición específica (el primero, el tercero...)

Para eso están las pseudoclases, ya que se representan
como una **pseudoselección**
y no necesitan atributos HTML

CSS (Cascading Style Sheets) - Componentes

Pseudo x estado

**Buscamos un estado especial en un escenario CSS.
Cuando pase el ratón por encima del párrafo
se transformará en rojo**

```
<p>Lorem ipsum dolor sit amet  
consectetur adipisicing elit. Culpa  
rehrenderit exercitationem eaque  
totam autem, similique fugiat eos ea  
aperiam quasi molestiae asperiores  
error corporis ad debit is maxime  
explicabo neque libero?</p>
```

**No ha necesitado ningún tipo
de atributo en HTML**

```
1  p{  
2      color: blue;  
3  }  
4  
5  p:hover{  
6      color: red;  
7  }  
8  
9  }  
10 }  
11 |
```

CSS (Cascading Style Sheets) - Componentes

Otro ejemplo

**Cuando pinchamos encima del recuadro a rellenar
se transformará en blue**

```
<input type="text" placeholder="Escribe aquí">
```

```
10  }
11
12 input:focus{
13   background-color: blue;
14 }
```

**No ha necesitado ningún tipo
de atributo en HTML**



CSS (Cascading Style Sheets) - Componentes

Hay montones de Pseudoclases

:active
:checked
:default
:dir()
:disabled
:empty
:enabled
:first
:first-child
:first-of-type
:fullscreen
:hover
:indeterminate

:in-range
:invalid
:lang()
:last-child
:last-of-type
:focus
:left
:link
:not()
:nth-child()
:nth-last-child()
:nth-last-of-type()
:nth-of-type()

:only-child
:only-of-type
:optional
:out-of-range
:read-only
:read-write
:required
:right
:root
:scope (en-US)
:target
:valid
:visited

CSS (Cascading Style Sheets) - Componentes

Pseudo x posición

Buscamos un estado en base a su posición y seguimos sin tocar ningún atributo en HTML

```
<article>
  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Officiis ex
  molestias aspernatur, sit alias culpa itaque reiciendis, hic laudantium
  sequi ea repellat natus placeat dolore, neque quaerat, odio voluptates
  commodi!
  </p>
  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Officiis ex
  molestias aspernatur, sit alias culpa itaque reiciendis, hic laudantium
  sequi ea repellat natus placeat dolore, neque quaerat, odio voluptates
  commodi!
  </p>
</article>
```

```
article p:first-child{
  color: orange;
}
```

Lorem ipsum dolor sit amet consectetur adipisicing elit. Officiis ex molestias aspernatur, sit alias culpa itaque reiciendis, hic laudantium sequi ea repellat natus placeat dolore, neque quaerat, odio voluptates commodi!

Lorem ipsum dolor sit amet consectetur adipisicing elit. Officiis ex molestias aspernatur, sit alias culpa itaque reiciendis, hic laudantium sequi ea repellat natus placeat dolore, neque quaerat, odio voluptates commodi!

CSS (Cascading Style Sheets) - Componentes

Pseudo x relación

Buscamos un estado en base a su relación específica.

Sigue sin necesitar atributo

```
<!-- Ejemplo por relación -->  
  
<hr>  
<h1>Ejemplo de relación</h1>  
<div>  
  <p class="especifica">Lorem, ipsum dolor sit amet consectetur adipisicing elit.  
  Ducimus corporis nisi tenetur quas consequatur quos quasi facere consectetur quae  
  sequi ut ullam officiis molestias nulla veniam, odio suscipit. Minus, beatae.  
  </p>  
  <br>  
  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Quisquam ab recusandae,  
  voluptate nostrum fugit veniam sunt quidem, accusamus  
  sit eveniet perferendis labore. Sit tempora dolorem,  
  </p>  
  <br>  
  <p>Lorem ipsum dolor sit amet consectetur adipisicing  
  reprehenderit, harum veniam numquam explicabo quis qui  
  perspiciatis sequi esse voluptas at fugiat, blanditiis  
  </p>  
  <br>  
</div>
```

/*Ejemplo por relación*/

p:not(.especifica){
 color: darkcyan;
 text-decoration: underline;
}

CSS (Cascading Style Sheets) - Componentes

Pseudo Elemento

Tipografía

**Consigue mayor control y precisión a la hora
crear estilos CSS y maquetaciones HTML**

**Se usan para definir partes concretas
de un elemento HTML**

**A diferencia de las PseudoClases se representan
con dos veces, dos puntos ::**

```
51
52 h1::first-letter{
53   color: red;
54 }
55
56 p::first-line{
57   color: orange;
58   text-decoration: underline;
59 }
```

**No modifica a todo el elemento,
sino a un subElemento,
a un PseudoElemento**



CSS (Cascading Style Sheets) - Componentes

PseudoElemento Contenido

```
li{  
    list-style-type: none;  
}  
  
li::before{  
    content: "> ";  
}  
  
li::after{  
    content: " <";  
}
```

Establece el tipo de viñeta para una lista

Agrega contenido antes de un elemento

Agrega contenido después de un elemento

Vamos a verlo todo con ejemplos

CSS (Cascading Style Sheets) - Componentes

**Unidades de Medida
Unidades de Color
Modelos de Cajas
y tipos de elementos**



CSS (Cascading Style Sheets) - Componentes

Tipografía `<font-.../>`

UF1303 Elaboración de hojas de estilo

CSS (Cascading Style Sheets) - Componentes



The image shows a screenshot of the Google Fonts website. At the top, there is a large yellow title "Texto y Tipos". Below it, a red subtitle asks "¿Cómo dar estilos tipográficos?". A white text statement says "Dar estilos tipográficos es esencial en nuestro diseño WEB,". The Google Fonts logo, featuring a stylized 'G' composed of colored triangles (red, yellow, green), is prominently displayed. The main content area shows three font families: "Roboto" by Christian Robertson, "Bricolage Grotesque" by Mathieu Triry, and "Ruwudu" by SIL International. Each font family has a preview text sample and a link to its page. The interface includes a search bar, filter buttons for categories, language, and number of styles, and a color palette. The overall theme is dark with light-colored text and icons.

CSS (Cascading Style Sheets) - Componentes

Texto y Tipos

Referenciarlas y manipularlas

```
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Open+Sans&family=Pacifico" ...>
```

Realiza conexiones tempranas para mejorar el rendimiento de la página

```
1  *{
2    margin: 0;
3    padding: 0;
4  }
5
6  div{
7    width: 300px;
8    margin: auto;
9  }
10
11
12  h1{
13    font-family: 'Pacifico', cursive;
14    text-align: left;
15  }
16
17  p{
18    font-family: 'Open Sans', sans-serif;
19    font-size: 20px;
20    font-style: italic;
21    color: darkgray;
22    text-align: justify;
23    text-decoration: underline;
24    letter-spacing: 3px;
25    line-height: 30px;
26
27  }
```

Título

Lorem ipsum dolor, sit, amet consectetur adipisicing elit. Sit, consectetur aut a expedita. Mollitia, temporibus? Voluptatum asperiores sapiente explicabo sed molestias, ex officia quibusdam nihil eos debitis sit aliquam sunt.

CSS (Cascading Style Sheets) - Componentes

Unidades de Medida



CSS (Cascading Style Sheets) - Componentes

Longitud Absoluta

Unidad	Nombre	Equivale a
cm	Centímetros	1cm = 96px/2,54
mm	Milímetros	1mm = 1/10 de 1cm
Q	Cuartos de milímetros	1Q = 1/40 de 1cm
in	Pulgadas	1in = 2,54cm = 96px
pc	Picas	1pc = 1/6 de 1in
pt	Puntos	1pt = 1/72 de 1in
px	Píxeles	1px = 1/96 de 1in

CSS (Cascading Style Sheets) - Componentes

Longitud Relativa

Unidad	Relativa a
em	Tamaño de letra del elemento padre, en el caso de propiedades tipográficas como <code>font-size</code> , y tamaño de la fuente del propio elemento en el caso de otras propiedades, como <code>width</code> .
ex	Altura x de la fuente del elemento.
ch	La medida de avance (ancho) del glifo "0" de la letra del elemento.
rem	Tamaño de la letra del elemento raíz.
lh	Altura de la línea del elemento.
vw	1% del ancho de la ventana gráfica.
vh	1% de la altura de la ventana gráfica.
vmin	1% de la dimensión más pequeña de la ventana gráfica.
vmax	1% de la dimensión más grande de la ventana gráfica.

%
Relativa al
elemento padre

CSS (Cascading Style Sheets) - Componentes

Unidades de Color



CSS (Cascading Style Sheets) - Componentes

Medidas de Color

Sistema de color en CSS

El color en CSS puede expresarse de diferentes formas; podemos insertarlos con su **nombre**, con su **RGB**, su **hexadecimal** o su **HSL**.

Veamos cómo funciona cada uno...

RGB

La sigla **RGB** representa colores a partir de la intensidad de los colores (**red, green, blue**).

La intensidad de cada color se determina por un número del 0 al 255, siendo el 0 la ausencia del color y el 255 su forma más intensa. En el modelo RGB, el color se representa usando tres números del 0 al 255, cada uno define la intensidad de un color en el orden R-G-B.

CSS (Cascading Style Sheets) - Componentes

Medidas de Color

Hexadecimal

En su código hexadecimal estos números se representan con parejas de dígitos que usan números del **0** al **9** y letras de la **a** a **F**.

La intensidad empieza por el **0** y termina en la **F**.
De este modo, el mínimo de un color viene siendo **00 (RGB 0)** y su máximo **FF (RGB 255)**.

En medio de estos valores habrá una combinación de números y letras que representan un número **RGB** según su orden.

CSS (Cascading Style Sheets) - Componentes

Medidas de Color

HSL

El modelo **HSL** es una sigla para los conceptos de **hue** (tono), **saturation** (saturación) y **lightness** (brillo).

Al igual que el **RGB**, se escribe en nuestra hoja de estilo **CSS** con su sigla **HSL** y su valor se escribe entre paréntesis.

Este modelo nos permite crear conjuntos de colores de manera más intuitiva, ya que podemos simplemente cambiar la saturación y el brillo para tener variaciones de un mismo color.

CSS (Cascading Style Sheets) - Componentes

Medidas de Color

Canal ALFA

En las representaciones RGB y HSL existe la posibilidad de agregar un cuarto valor que representa el canal alfa.

El **canal alfa** se refiere a la transparencia u opacidad de un color y se define con un número entre el 0 y el 1, siendo el 0 el valor más transparente y el 1 el valor más opaco.

Esto se escribe en nuestra hoja de estilo CSS como **RGBA** o **HSLA** y es una propiedad que se hereda del elemento padre al elemento hijo.

Si quisiéramos un verde con un 30% de opacidad podríamos escribir **RGBA (0, 255, 0, 0.3)**. Para escribir el mismo valor en HSL lo escribiríamos como **HSLA (120, 100, 50, 0.3)**.

OJO, NO FUNCIONA CON NOMBRES PREDEFINIDOS (Rojo, 0.5)

CSS (Cascading Style Sheets) - Componentes

Medidas de Color

Ejemplo de Colores por Nombre, RGB y HLA



Color	Name	Hex Code	RGB Code
	IndianRed	#CD5C5C	rgb(205, 92, 92)
	LightCoral	#F08080	rgb(240, 128, 128)
	Salmon	#FA8072	rgb(250, 128, 114)
	DarkSalmon	#E9967A	rgb(233, 150, 122)

<https://htmlcolorcodes.com/es/nombres-de-los-colores/>

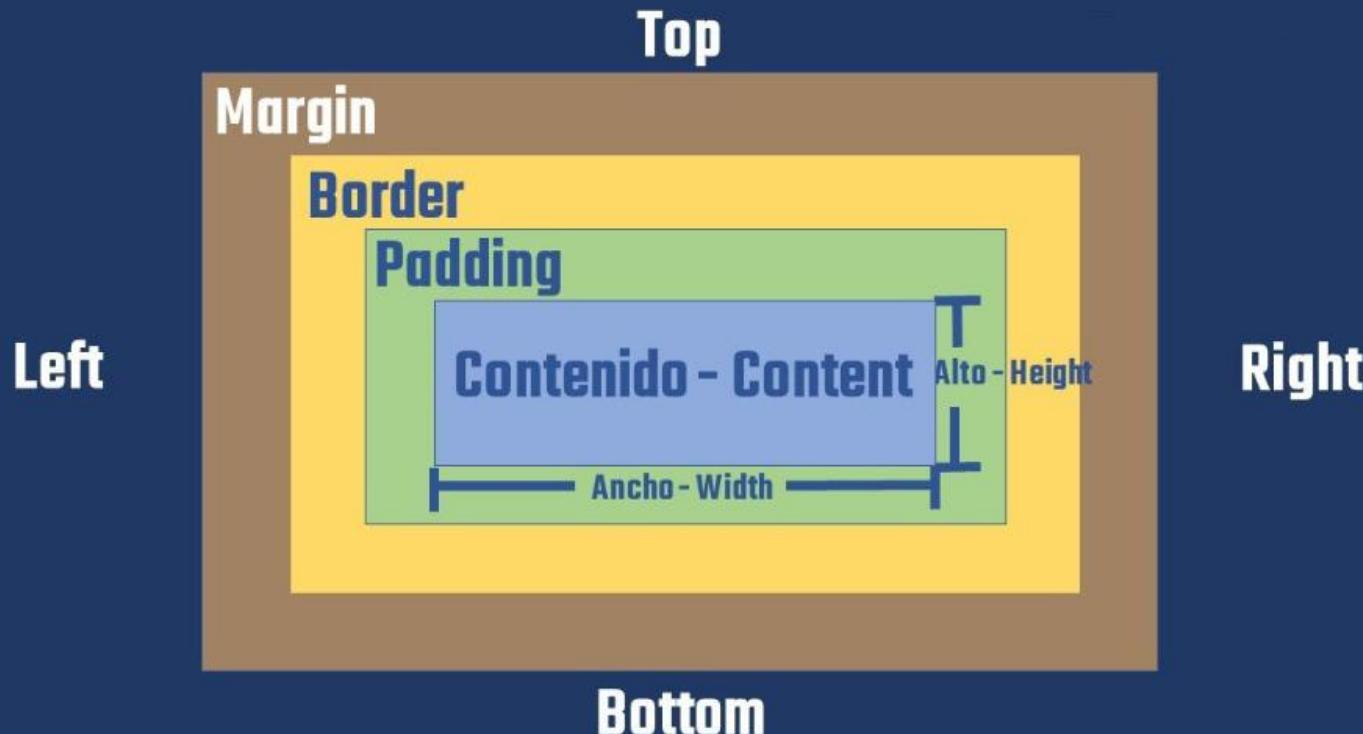
CSS (Cascading Style Sheets) - Componentes

Modelos de Caja



CSS (Cascading Style Sheets) - Componentes

Model Box



CSS (Cascading Style Sheets) - Componentes

Model Box

```
<div class="cajaA"> Caja A
    <div class="cajaB">Caja B
        </div>
</div>
```

Caja A
Caja B

```
*{
    margin: 0;
    padding: 0;
    font-family: sans-serif;
}

.cajaA{
    background-color:red;
    width: 50%;
    margin: 50px auto;
    font-size: 50px;
    padding: 30px 0px 0px 30px;
    color: white;
}

.cajaB{
    background-color:yellow;
    color: blue;
    font-size: 100px;
    width: 100%;
    padding: 30px 30px 30px 30px ;
    border: 30px solid magenta;
}
```

CSS (Cascading Style Sheets) - Componentes

Fondos

<https://unsplash.com/es>

<https://tinypng.com>

CSS (Cascading Style Sheets) - Componentes

**floats &
positions**



CSS (Cascading Style Sheets) - Componentes

FLOATS

Hemos visto cómo se comparte una CAJA utilizando medidas importantes como el margin y el padding, ademas de darle ancho y alto con las propiedades width y height

Ahora vamos a ver cómo se relacionan las cajas entre ellas como se pone una caja al lado de otra, o una encima de otra... esto es divertido :-))

CSS (Cascading Style Sheets) - Componentes

FLOATS

```
<div class="caja1">Soy la caja 1</div>
<div class="caja2">Soy la caja 2</div>
<div class="caja3">Soy la caja 3</div>
```



CSS (Cascading Style Sheets) - Componentes

FLOATS

```
<div class="caja1">Soy la caja 1</div>
<div class="caja2">Soy la caja 2</div>
<div class="caja3">Soy la caja 3</div>
```

```
.caja1, .caja2, .caja3{
    color: black;
    width: 200px;
    height: 200px;
    float: left;
}

.caja1{
    background-color: red;
}

.caja2{
    background-color: yellowgreen;
}

.caja3{
    background-color: cyan;
}
```



CSS (Cascading Style Sheets) - Componentes

**positions:
absolute &
relative**



CSS (Cascading Style Sheets) - Componentes

FlexBox

Es un sistema de elementos flexibles.
Los elementos HTML se adaptan y colocan automáticamente siendo mucho más fácil personalizar los diseños.

Está pensado para diseños en una sola dimensión, es decir, o filas o columnas.

Antes de que existiese el Flexbox había otros modos de hacer layouts:

Block
Inline
Table
Position

CSS (Cascading Style Sheets) - Componentes

FlexBox

CONTAINER

ITEM1

ITEM2

ITEM3

ITEM4

ITEMS

por defecto: flex-direction: row

El **FlexBox** hace que se más fácil crear estructuras de layout **responsive**, sin usar posicionamiento o flotantes.

Los layouts en FlexBox trabajan en una sola dimensión, en **filas (row)** como en el ejemplo o en **columnas (column)**

CSS (Cascading Style Sheets) - Componentes

FlexBox

```
<div class="flexContainer">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
  <div>5</div>
</div>
```

```
7 .flexContainer div{
8   background-color: ■white;
9   color: ■red;
10  width: 150px;
11  height: 150px;
12  margin-bottom: 10px;
13  display: grid;
14  place-items: center;
15  font-size: 30px;
16  font-weight: 900;
17  border-radius: 50%;
18  margin: 20px;
19
20 }
21
22 .flexContainer{
23   background-color: ■rgb(253,198,18);
24   /* ----- */
25 }
```



CSS (Cascading Style Sheets) - Componentes

FlexBox

Flex-Container

```
21  
22 .flexContainer{  
23   background-color: #rgb(253,198,18);  
24   /* ----- */  
25   display: flex;  
26 }
```

1

2

3

4

5

CSS (Cascading Style Sheets) - Componentes

FlexBox

Flex-Container

```
21  
22 .flexContainer{  
23     background-color: #rgb(253,198,18);  
24     /* ----- */  
25     display: flex;  
26     flex-direction: column;  
27 }
```

1

2

3

4

5

CSS (Cascading Style Sheets) - Componentes

FlexBox

Flex-Container

```
21  
22 .flexContainer{  
23     background-color: #rgb(253, 198, 18);  
24     /* ----- */  
25     display: flex;  
26     flex-direction: column-reverse;  
27 }
```

5

4

3

2

1

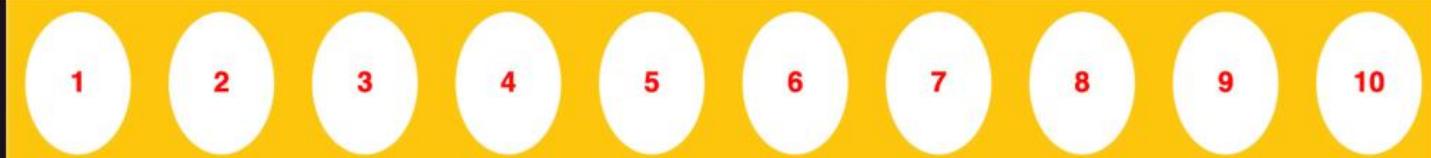
CSS (Cascading Style Sheets) - Componentes

FlexBox

Flex-Container

```
<div class="flexContainer">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
  <div>5</div>
  <div>6</div>
  <div>7</div>
  <div>8</div>
  <div>9</div>
  <div>10</div>
</div>
```

```
21
22 .flexContainer{
23   background-color: #rgb(253,198,18);
24   /* ----- */
25   display: flex;
26 }
```

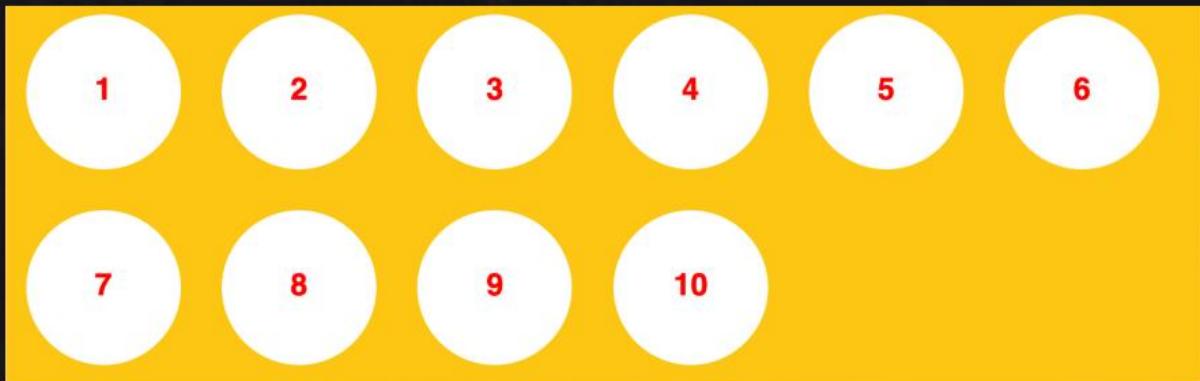


CSS (Cascading Style Sheets) - Componentes

FlexBox

Flex-Container

```
21  
22 .flexContainer{  
23     background-color: #rgb(253, 198, 18);  
24     display: flex;  
25     /* Default:nowrap */  
26     flex-wrap: wrap;  
27 }
```

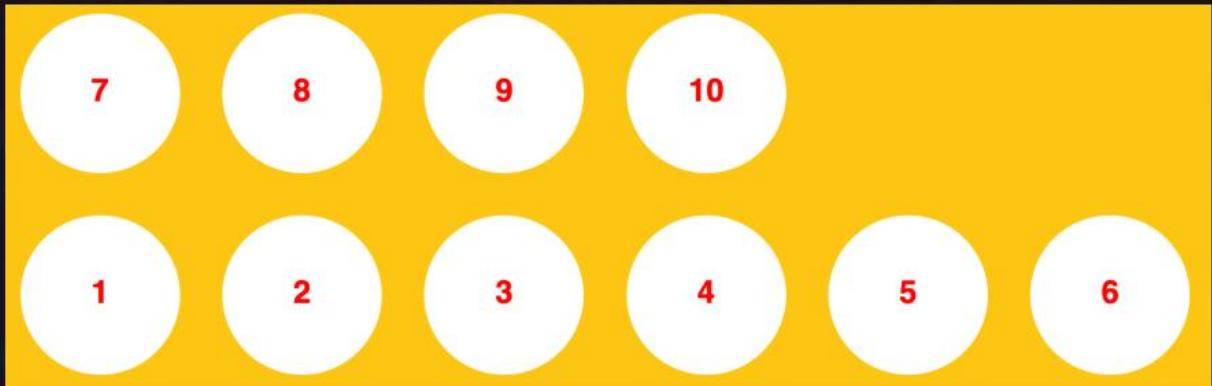


CSS (Cascading Style Sheets) - Componentes

FlexBox

Flex-Container

```
21  
22 .flexContainer{  
23     background-color: #rgb(253, 198, 18);  
24     display: flex;  
25     /* Default: nowrap */  
26     flex-wrap: wrap-reverse;  
27 }
```

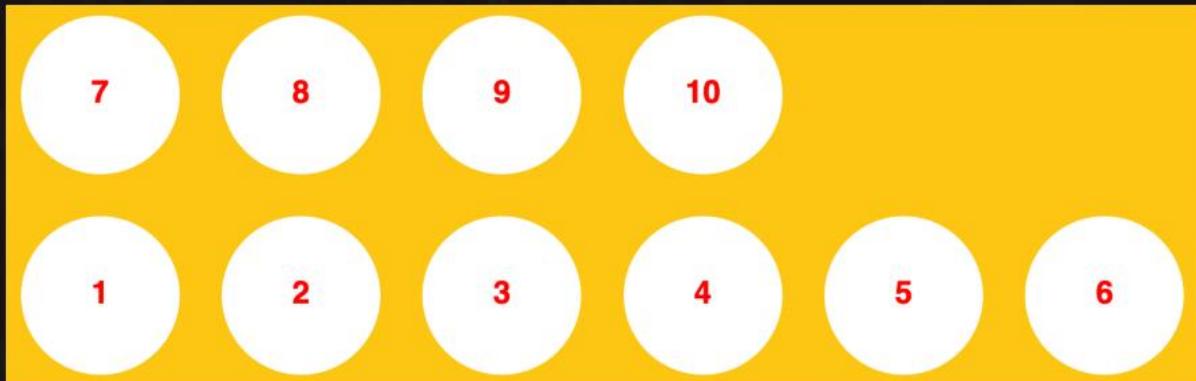


CSS (Cascading Style Sheets) - Componentes

FlexBox

Flex-Container

```
.flexContainer{  
    background-color: #rgb(253,198,18);  
    display: flex;  
    /* Unifican Dirección y Espacio */  
    flex-flow: row wrap-reverse;
```



CSS (Cascading Style Sheets) - Componentes

FlexBox

Flex-Container

```
21  
22 .flexContainer{  
23     background-color: #rgb(253,198,18);  
24     display: flex;  
25     /* ----- */  
26     justify-content: center ;  
27 }
```



CSS (Cascading Style Sheets) - Componentes

FlexBox

Flex-Container

```
.flexContainer{  
    background-color: #rgb(253,198,18);  
    display: flex;  
    /* ----- */  
    justify-content: flex-start ;  
  
}
```



```
.flexContainer{  
    background-color: #rgb(253,198,18);  
    display: flex;  
    /* ----- */  
    justify-content: flex-end ;  
  
}
```



CSS (Cascading Style Sheets) - Componentes

FlexBox

Flex-Container

```
.flexContainer{  
    background-color: #rgb(253, 198, 18);  
    height: 400px;  
    display: flex;  
    /* ----- */  
    align-items: center ;  
}
```

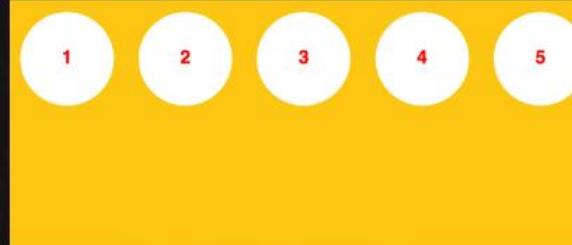


CSS (Cascading Style Sheets) - Componentes

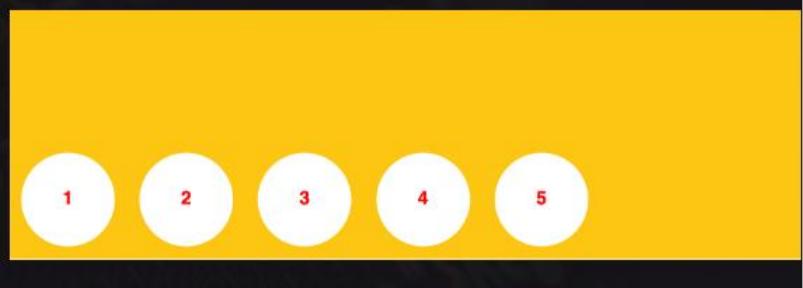
FlexBox

Flex-Container

```
.flexContainer{  
    background-color: #rgb(253,198,18);  
    height: 400px;  
    display: flex;  
    /* ----- */  
    align-items: flex-start;
```



```
.flexContainer{  
    background-color: #rgb(253,198,18);  
    height: 400px;  
    display: flex;  
    /* ----- */  
    align-items: flex-end;
```



CSS (Cascading Style Sheets) - Componentes

FlexBox

Flex-Container

```
<div class="flexContainer">
  <div style="font-size: 110px;">1</div>
  <div style="font-size: 60px;">2</div>
  <div style="font-size: 100px;">3</div>
</div>
```

```
.flexContainer{
  background-color: #rgb(253, 198, 18);
  height: 400px;
  display: flex;
  /* ----- */
  align-items: baseline;
```



CSS (Cascading Style Sheets) - Componentes

FlexBox

Flex-Items

propiedades de
los elementos hijos
de container



UF1303 Elaboración de hojas de estilo

CSS (Cascading Style Sheets) - Componentes

FlexBox

Flex-Items

```
.flexContainer{  
    background-color: #rgb(253,198,18);  
    height: 400px;  
    display: flex;  
    /* ----- */  
    align-items: stretch ;  
}
```

Stretch, hace que se ajuste a todo el espacio

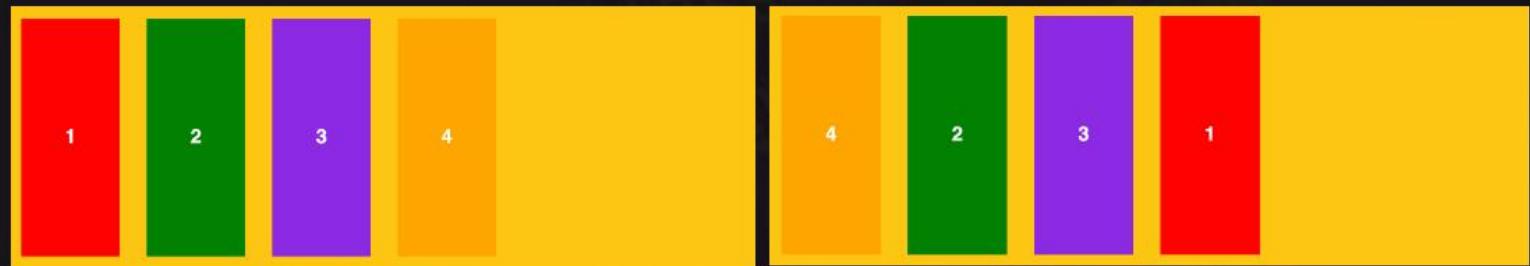
```
<div class="flexContainer">  
    <div style="order:1; background-color: red;">1</div>  
    <div style="order:2; background-color: green;">2</div>  
    <div style="order:3; background-color: blueviolet;">3</div>  
    <div style="order:4; background-color: orange;">4</div>  
</div>
```

CSS (Cascading Style Sheets) - Componentes

FlexBox

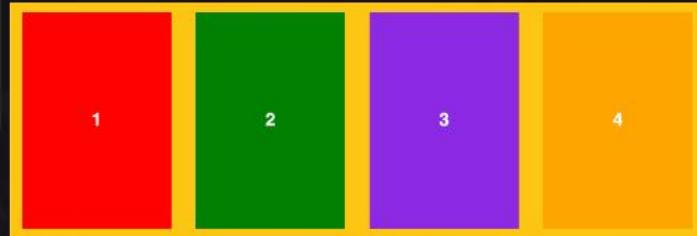
Flex-Items

```
<div class="flexContainer">
    <div style="order:4; background-color: red;">1</div>
    <div style="order:2; background-color: green;">2</div>
    <div style="order:3; background-color: blueviolet;">3</div>
    <div style="order:1; background-color: orange;">4</div>
</div>
```

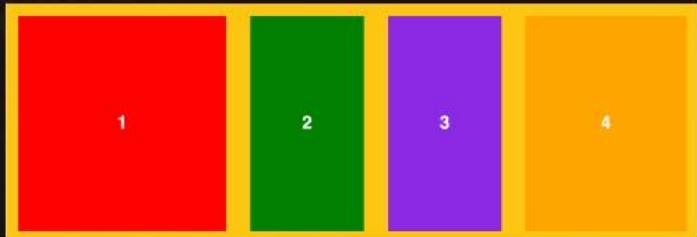


CSS (Cascading Style Sheets) - Componentes

```
<div class="flexContainer">
  <div style="flex-grow: 1; background-color: red;">1</div>
  <div style="flex-grow: 1; background-color: green;">2</div>
  <div style="flex-grow: 1; background-color: blueviolet;">3</div>
  <div style="flex-grow: 1; background-color: orange;">4</div>
</div>
```



```
<div class="flexContainer">
  <div style="flex-grow: 5; background-color: red;">1</div>
  <div style="flex-grow: 1; background-color: green;">2</div>
  <div style="flex-grow: 1; background-color: blueviolet;">3</div>
  <div style="flex-grow: 3; background-color: orange;">4</div>
</div>
```



CSS (Cascading Style Sheets) - Componentes

FlexBox

Flex-Items

Espacios más pequeño NO!!!

```
<div class="flexContainer">
  <div style="flex-shrink: 0; background-color: red;">1</div>
  <div style="background-color: green;">2</div>
  <div style="background-color: blueviolet;">3</div>
  <div style="background-color: orange;">5</div>
  <div style="background-color: violet;">6</div>
  <div style="background-color: magenta;">7</div>
  <div style="background-color: cyan;">8</div>
  <div style="background-color: darkblue;">9</div>
</div>
```

The diagram illustrates two scenarios of FlexBox item distribution. On the left, items 1 through 9 are distributed evenly across the container, each occupying approximately equal width. On the right, item 1 is given a large `flex-grow: 1` value, causing it to expand and fill most of the available space, while the other items are compressed into smaller gaps.

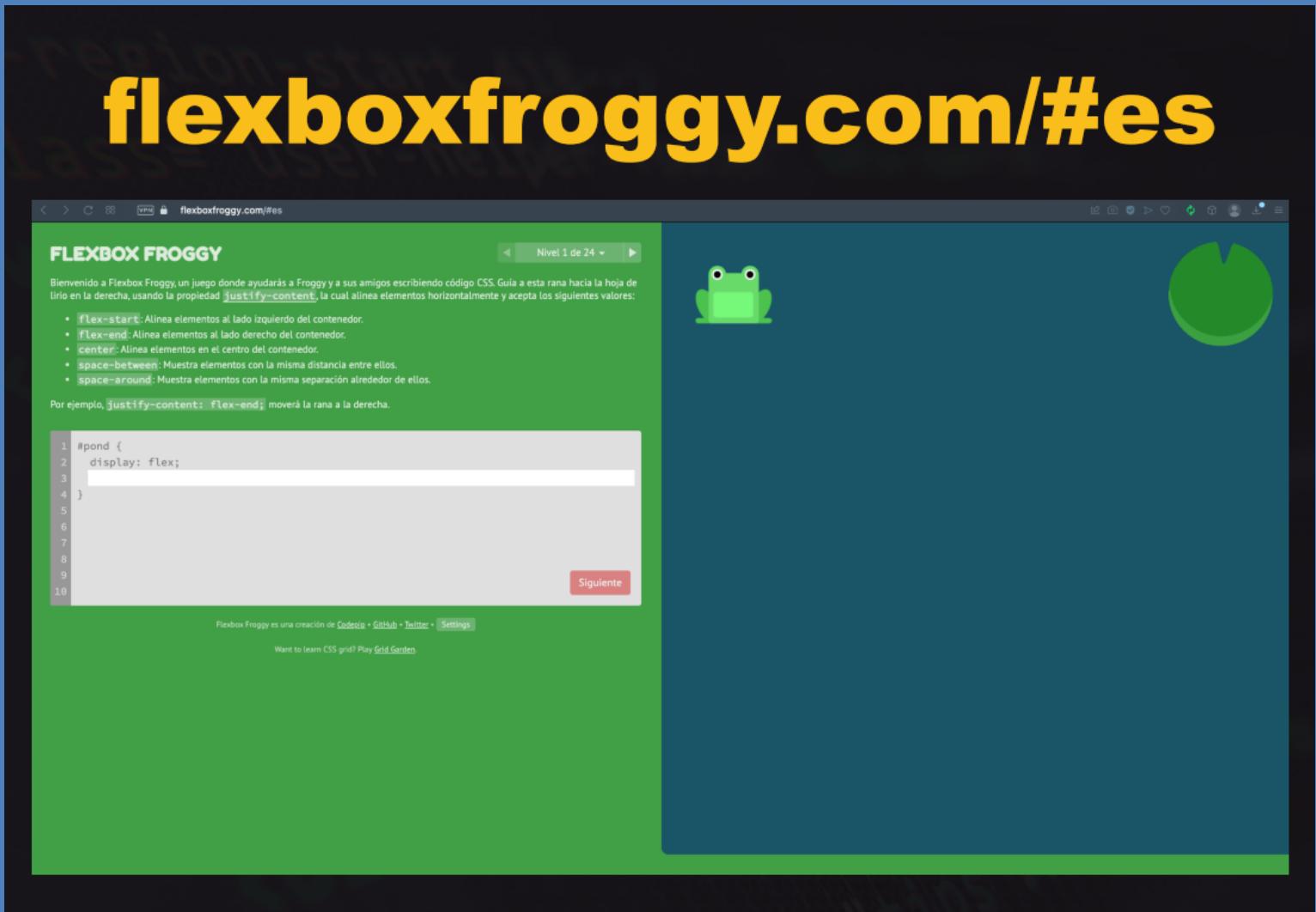
UF1303 Elaboración de hojas de estilo

CSS (Cascading Style Sheets) - Componentes



UF1303 Elaboración de hojas de estilo

CSS (Cascading Style Sheets) - Componentes



CSS (Cascading Style Sheets) - Componentes

**@mediaQueries | CSS
Responsive Web Design**



CSS (Cascading Style Sheets) - Componentes

Mobile First

Como su propio nombre indica mobile first o por su traducción en inglés, «los móviles primero», resume perfectamente esta filosofía de desarrollo.



Mobile first: ¿Cómo mejora el posicionamiento?

Google considerará siempre, en primera estancia, la versión móvil de tu sitio, para posicionarla.

CSS (Cascading Style Sheets) - Componentes

¿Mobile Friendly?

Responder a esta pregunta, nos dará las claves a la hora de mejorar nuestro sitio web.



La optimización del contenido para móviles.
Nos referimos tanto a los tiempos de carga, la experiencia de la navegación, el tamaño, etc.

CSS (Cascading Style Sheets) - Componentes

¿Mobile Friendly?

**Google premia que una web tenga
un tiempo de carga reducido.**

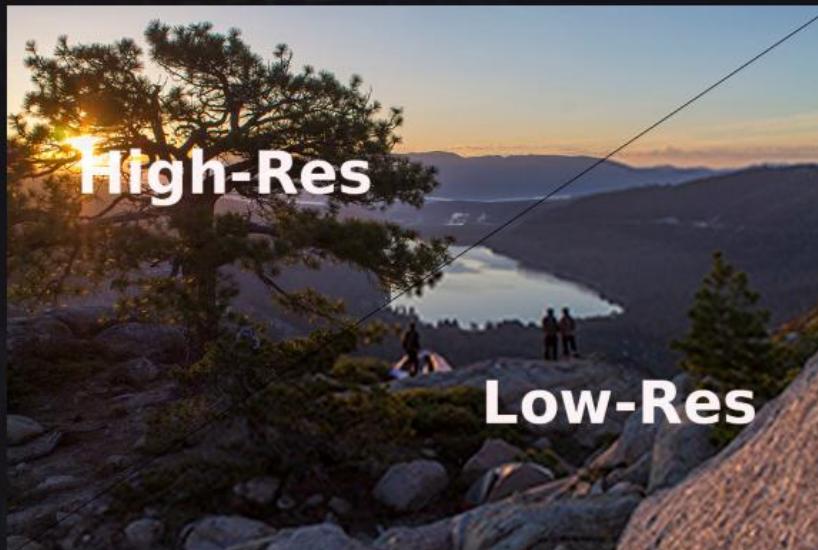
**Todas las páginas de nuestra web tienen que funcionar a
pleno rendimiento para garantizar que esto sea así.**



CSS (Cascading Style Sheets) - Componentes

¿Mobile Friendly?

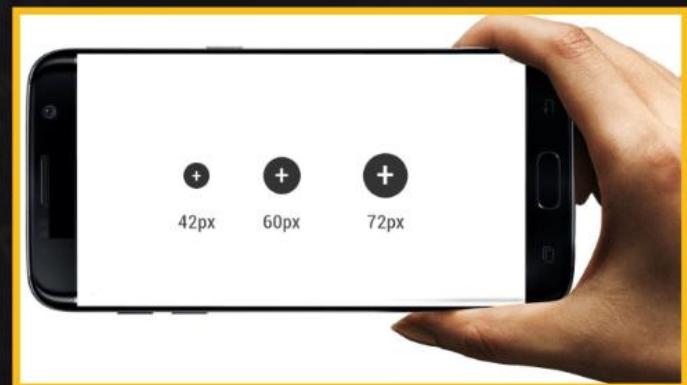
**El uso de imágenes de alta calidad
Esto es muy importante, ya que muchos móviles poseen
resoluciones muy altas, e imágenes que no sean de alta
definición, se verían borrosas.**



CSS (Cascading Style Sheets) - Componentes

¿Mobile Friendly?

El tamaño de la fuente y de los botones tienen que ser lo suficientemente grandes, como para que el usuario no pierda la lectura sin hacer zoom, y que no presione un botón por error..



CSS (Cascading Style Sheets) - Componentes

**Es una metodología
Mobile > Desktop
¿Tú eliges?**



CSS (Cascading Style Sheets) - Componentes

**¿Qué
necesitamos
para ello?**



**¿Primero?
Lo llevamos viendo
desde el principio**

CSS (Cascading Style Sheets) - Componentes

<meta>

```
3 <!DOCTYPE html>
4 <html lang="es">
5 <head>
6   <meta charset="UTF-8">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>Titulo</title>
9 </head>
```

Con el **width=device-width** conseguimos que el **viewport** sea igual a la anchura real de la pantalla del dispositivo, que no emule una pantalla mayor.

Con **initial-scale=1** conseguimos que no se haga zoom sobre el documento. Es bien simple, el contenido de la web no se transformará, ni se agrandará, ni se hará menor.



CSS (Cascading Style Sheets) - Componentes



CSS (Cascading Style Sheets) - Componentes

@media(min/max-width)

¿Cómo usarlas?

MÓVILES VERTICALES



Estilos normales

MÓVILES HORIZONTAL
TABLET VERTICAL



@media (min-width: 768px){}

TABLET HORIZONTAL
ESCRITORIOS NORMALES



@media (min-width: 1024px){}

ESCRITORIOS MUY ANCHOS



@media (min-width: 1200px){}

min-width

max-width

```
25 @media(min-width: 768px){  
26 }  
27  
28 @media(min-width: 1024px){  
29 }  
30  
31 @media(min-width: 1080px){  
32 }
```

```
25 @media(max-width: 1080px){  
26 }  
27  
28 @media(max-width: 1024px){  
29 }  
30  
31 @media(max-width: 768px){  
32 }
```



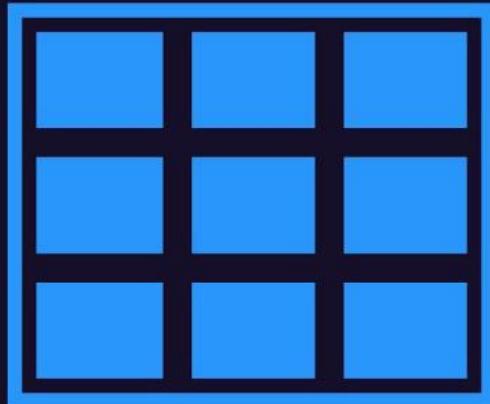
CSS (Cascading Style Sheets) - Componentes



CSS (Cascading Style Sheets) - Componentes

...porque hay plan G

CSS Grid vs Flexbox



CSS (Cascading Style Sheets) - Componentes

...a practicar

`<header>`

`<nav>`

`<section>`

`<article>`

`<aside>`

`<footer>`

UF1303 Elaboración de hojas de estilo

CSS (Cascading Style Sheets) - Componentes



(IFCDO110) CONFECCIÓN Y PUBLICACIÓN DE PÁGINAS WEB

UF1303 Elaboración de hojas de estilo

CSS (Cascading Style Sheets) - Componentes



(IFCDO110) CONFECCIÓN Y PUBLICACIÓN DE PÁGINAS WEB

UF1303 Elaboración de hojas de estilo

CSS (Cascading Style Sheets) - Componentes



(IFCDO110) CONFECCIÓN Y PUBLICACIÓN DE PÁGINAS WEB

UF1303 Elaboración de hojas de estilo



(IFCDO110) CONFECCIÓN Y PUBLICACIÓN DE PÁGINAS WEB



Otras Etiquetas

Etiqueta <style>

La etiqueta **<style>** es un elemento HTML5 que se utiliza para incluir estilos CSS (Cascading Style Sheets) directamente dentro de un documento HTML.

Estos **estilos**, también **conocidos como estilos internos o embebidos**, se aplican a los elementos HTML dentro de la página, permitiendo personalizar su apariencia y comportamiento visual.

La etiqueta **<style>** **debe colocarse dentro del elemento <head>** del documento HTML, aunque algunos navegadores también admiten su uso en el cuerpo del documento.

Sin embargo, para garantizar la compatibilidad y seguir las mejores prácticas, es recomendable colocarla en la sección **<head>**. Un documento **HTML** puede contener varias etiquetas **<style>**, aunque es común agrupar todos los estilos en una sola.

Otras Etiquetas

Etiqueta <style>

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Mi sitio web</title>
7
8      <style>
9          body {
10              font-family: Arial, sans-serif;
11              background-color: #f0f0f0;
12          }
13          h1 {
14              color: #333;
15          }
16      </style>
17  </head>
18  <body>
19      <h1>¡Bienvenido a mi sitio web!</h1>
20      <p>Este es un ejemplo de cómo utilizar la etiqueta &lt;style&gt; en HTML5.</p>
21  </body>
22  </html>
```

Otras Etiquetas

Etiqueta <style>

En este ejemplo, la etiqueta <**style**> contiene estilos CSS que modifican el tipo de fuente, el color de fondo y el color del texto para los elementos de la página. Estos estilos se aplicarán a todos los elementos HTML correspondientes en el documento.

GRACIAS POR TU ATENCIÓN



(IFCDO110) CONFECCIÓN Y PUBLICACIÓN DE PÁGINAS WEB