

# 1T -HITO GRUPAL

## Diseño de Interfaces Web

### - Partes del guión:

- Presentación.
- Información (de que vamos a hablar). (Presentación del tema)
- Información de CSS. (Nociones básicas de css)
- Donde podemos compilar CSS (Visual Studio Code, Sublime, ....).
- El tema a abordar.
- Buenas prácticas (Formas de mejorar el código)
- Algún ejercicio que lo podemos subir al Github.
- Y despedirnos.

- **Presentación**

Buenas a todos, bienvenidos al podcast para la asignatura de Desarrollo de Interfaces Web, somos Carlos Parro y Oscar Izquierdo alumnos de 2º de DAW en Campus FP Getafe y el tema a tratar de hoy nos adentra en el mundo de la programación web y más bien nos sitúa dentro del asombroso mundo del diseño de páginas web.

Empezaremos explicando qué es el concepto css y su importancia dentro de la programación web, a continuación avanzaremos comentando y adentrándonos en la historia del mismo comentando cómo surgió y cómo ha evolucionado hasta el día de hoy, trataremos de explicar los diferentes estilos de implementación de código css, y dentro de ello cómo podemos acceder a los elementos dentro del lenguaje, así como los diferentes pseudoselectores que existen para realizarlo.

Por otro lado, nos adentraremos en los diferentes tipos de preprocesadores que se hallan disponibles, como formas diferentes de compilar código css.

Y por último, hablaremos de los diferentes usos que podemos darle al css mediante la utilización de animaciones y el importante concepto del RWD o Web Responsive Design dentro del diseño web.

- **Explicación de CSS a fondo, Importancia e Historia**

### **Información general de CSS**

Las hojas de estilos aparecieron poco después que el lenguaje de etiquetas SGML, alrededor del año 1970. Desde la creación de SGML, se observó la necesidad de definir un mecanismo que permitiera aplicar de forma consistente diferentes estilos a los documentos electrónicos.

El gran impulso de los lenguajes de hojas de estilos se produjo con el boom de Internet y el crecimiento exponencial del lenguaje HTML para la creación de documentos electrónicos. La guerra de navegadores y la falta de un estándar para la definición de los estilos dificultaron la creación de documentos con la misma apariencia en diferentes navegadores.

## **¿Pero qué es exactamente CSS en la actualidad?**

CSS es el lenguaje para describir la presentación de las páginas web, incluidos los colores, el diseño y las fuentes. Permite adaptar la presentación a diferentes tipos de dispositivos, como pantallas grandes, pantallas pequeñas o impresoras. CSS es independiente de HTML y se puede utilizar con cualquier lenguaje de marcado basado en XML. La separación de HTML de CSS hace que sea más fácil mantener los sitios, compartir hojas de estilo entre páginas y adaptar las páginas a diferentes entornos. Esto se conoce como la separación de la estructura (o: contenido) de la presentación.

## **Editores de código**

### **Visual Studio Code**

Visual Studio Code es uno de los editores multiplataforma de código abierto más potentes. En lugar del resaltado de sintaxis estándar y el autocompletado, VS Code ofrece IntelliSense, que proporciona terminaciones inteligentes basadas en tipos de variables, definiciones de funciones y módulos importados. Esto lo convierte en una solución poderosa para depurar su código.

### **Sublime Text**

Sublime Text es compatible con múltiples lenguajes de programación como Notepad ++. Ofrece características estándar que ofrece Notepad ++, incluyendo soporte multiplataforma, edición dividida y edición de selección múltiple.

### **Brackets**

Brackets ofrece dos características importantes: editores en línea y vista previa en vivo. En cambio, puede abrir una ventana y poner el código en el que desea enfocarse simultáneamente.

## **¿Cuál debo elegir?**

Los editores de código son programas muy sencillos que requieren poco espacio de almacenamiento y rendimiento del sistema. El dispositivo y el sistema operativo donde se utilicen dependen básicamente de los gustos del usuario. No obstante, es evidente que no todos los editores de código funcionan en todos los sistemas operativos, por lo que la elección puede verse limitada, por ejemplo, a uno que sea compatible con Linux. Asimismo, al igual que otros tipos de software, estos programas se clasifican en software propietario, de código abierto y gratis, por lo que, cuando tenemos que escoger un editor de código web, nuestra decisión final a menudo dependerá también del precio.

Cómo podemos acceder a los elementos de una web a través de css.

Para que el html se comuniquen con la página de css hay que seguir unos pasos que os contaré ahora.

1ºPaso: Abre tu archivo `index.html` y pega la siguiente línea en algún lugar dentro del cabeza de la página, es decir, entre las etiquetas `<head>` y `</head>`:

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

2ºPaso: Crearas un archivo css con el nombre que la hayamos puesto en la ruta que hemos puesto anteriormente en este caso se llamará `style.css`. Dentro de ese archivo pondremos esto:

```
p {  
    color: red;  
}
```

3º Guarda el archivo `index.html` y cárgalo en tu navegador. Debes ver que las etiquetas que tengas con una `p` se hayan puesto de color rojo.

## Anatomía de una regla CSS

### Selector

El elemento HTML en el que comienza la regla. Esta selecciona el(los) elemento(s) a dar estilo (en este caso, los elementos `<p>`). Para dar estilo a un elemento diferente, solo cambia el selector.

### Declaración

Una sola regla como `color: red;` especifica a cuál de las propiedades del elemento quieres dar estilo.

### Propiedades

Maneras en las cuales puedes dar estilo a un elemento HTML. (En este caso, `color` es una propiedad del elemento `<p>`). En CSS, seleccionas qué propiedad quieres afectar en tu regla.

### Valor de la propiedad

A la derecha de la propiedad, después de los dos puntos (:), tienes el valor de la propiedad, para elegir una de las muchas posibles apariencias para una propiedad determinada (hay muchos valores para `color` además de `red`).

### Nota las otras partes importantes de la sintaxis:

Cada una de las reglas (aparte del selector) deben estar encapsuladas entre llaves (`{}`).

Dentro de cada declaración, debes usar los dos puntos (:) para separar la propiedad de su valor.

Dentro de cada regla, debes usar el punto y coma (;) para separar una declaración de la siguiente.

(También podemos coger elementos de html con elementos de tipo `id` y `class`)

- **Tipos de Selectores**

Como hemos visto anteriormente, los selectores nos ayudan a indicar el elemento sobre el que se van a aplicar los estilos. Existen muchos tipos de selectores y algunos de los más destacados son los que se detallan a continuación.

**Selector universal**

Sintaxis: `* { atributo:valor; }`

Ejemplo: `* { color: grey; }` /\* El estilo se aplicará a todos los elementos de la página \*/

**Selector etiqueta**

Sintaxis: `etiqueta { atributo:valor }`

Ejemplo: `p {color: green;}` /\* El estilo se aplicará a todos los elementos `<p>`. \*/

**Selector clase**

Sintaxis: `.clase { atributo:valor }`

Ejemplo: `.blend{color: red;}` /\* El estilo se aplicará a cualquier elemento que tenga la clase `.blend` \*/

**Selector identificador**

El selector identificador utiliza el atributo `id` para seleccionar un elemento. Solo puede haber un elemento con un `id` dado en un documento.

Sintaxis: `#id { atributo:valor }`

Ejemplo: `#cent {color: blue;}` /\* El estilo se aplicará al elemento que tenga el `id` `#cent` \*/

**Selector descendiente**

Un elemento es descendiente de otro cuando se encuentra entre las etiquetas de apertura y de cierre del elemento padre.

Sintaxis: `selector1 selector2 selectorN {atributo: valor;}` /\* El estilo se aplica sobre el selector `N` \*/

Ejemplo: `div p { color: black;}` /\* El estilo se aplica a todos los párrafos que se encuentren dentro de una etiqueta `div` \*/

## Combinación de selectores

La combinación de selectores nos permite dar un estilo a todos los selectores indicados.

Sintaxis: selector1, selector2, selector3 { atributo: valor; } /\* El estilo se aplica sobre los selectores indicados \*/

Ejemplo: div, p { color: orange; } /\* El estilo se aplica a todos los divs y párrafos \*/

## Selector de hijos

Se usa para seleccionar un elemento que es hijo de otro elemento.

Sintaxis: selector1 > selector2 { atributo: valor; } /\* El estilo se aplica sobre el selector 2 \*/

Ejemplo: div > p { color: white; } /\* El estilo se aplica a todos los párrafos que sean hijos de un div \*/

## Selector adyacente

Se usa para seleccionar elementos que son hermanos, es decir, su elemento padre es el mismo y están seguidos en el código HTML.

Sintaxis: selector1 + selector2 { atributo: valor; } /\* El estilo se aplica al selector 2 \*/

Ejemplo: div + p { color: black; } /\* El estilo se aplica a todos los párrafos que sean hermanos de un div \*/

- **Preprocesadores, SAAS y LESS**

Un preprocesador de CSS se puede definir como una herramienta que nos permite escribir pseudocódigo CSS que luego será compilado de convertir en CSS tal y como lo conocemos de forma habitual. Este pseudocódigo está formado por variables, condicionales, bucles o funciones, elementos habituales de cualquier lenguaje de programación. Por este motivo, podríamos decir que tenemos un lenguaje de programación cuya misión es la de generar el código CSS.

El uso de preprocesadores en nuestros proyectos aporta una serie de beneficios entre los que podemos destacar.

Posibilidad de añadir funcionalidades adicionales que no son posible de utilizar en archivos CSS tradicionales, como son el uso de variables y lógica condicional.

Dispondremos de una hoja de estilos más limpia, consiguiendo a la vez que sea más eficiente y fácil de mantener.

Ofrece la posibilidad de reutilizar mucho código, lo que se traduce en un ahorro de tiempo y trabajo.

Los cambios serán mucho más rápidos, ya que únicamente deberemos cambiar el valor de alguna variable.

Ayuda a crear código compatible con todos los navegadores, lo que supone una gran ayuda para los diseñadores de sitios web.

Se crea una capa de abstracción, donde no trabajaremos directamente sobre nuestro archivo CSS, ofreciendo una mayor seguridad.

Posibilidad de separación absoluta del proceso de desarrollo y producción.

### **LESS vs. SASS: comparamos preprocesadores**

LESS, junto a SASS, se ha establecido también en los círculos de desarrollo como el lenguaje más utilizado para el diseño de páginas web. Este lenguaje de hojas de estilo está más orientado a CSS y por lo tanto es más similar a la sintaxis de SCSS. Los dos preprocesadores comparten algunas características: LESS y SASS contienen el uso de mixins y variables. Una de sus principales diferencias es que SASS está basado en Ruby, mientras que LESS utiliza JavaScript. Sin embargo, esta particularidad no proporciona a ninguno de los dos ventajas sobre el otro.

- **Animaciones y RWD (Diseño Adaptativo vs Diseño Responsive, Media Queries)**

Las animaciones: CSS3 permite animar la transición entre un estilo CSS y otro. Las animaciones constan de dos componentes: un estilo que describe la animación CSS y un conjunto de fotogramas que indican su estado inicial y final, así como posibles puntos intermedios en la misma.

Las animaciones CSS tienen tres ventajas principales sobre las técnicas tradicionales de animación basada en scripts:

Son muy fáciles de usar para animaciones sencillas, puedes hacerlo incluso sin tener conocimientos de Javascript.

La animación se muestra correctamente, incluso en equipos poco potentes. Animaciones simples realizadas en Javascript pueden verse mal (a menos que estén muy bien hechas). El motor de renderizado puede usar técnicas de optimización como el "frame-skipping" u otras técnicas para que la ejecución de la animación se vea tan suave como sea posible.

Al ser el navegador quien controle la secuencia de la animación, permitimos que optimice el rendimiento y eficiencia de la misma, por ejemplo, reduciendo la frecuencia de actualización de la animación ejecutándola en pestañas que no estén visibles.

## Configurando la animación

Para crear una secuencia de animación CSS, tú utilizarás el elemento que quieras animar con la propiedad `animation` y sus sub-propiedades. Con ellas podemos no solo configurar el ritmo y la duración de la animación sino otros detalles sobre la secuencia de la animación. Con ellas no configuramos la apariencia actual de la animación, para ello disponemos de `@keyframes` como describiremos más adelante .

Las subpropiedades de `animation` son:

### **animation-delay**

Tiempo de retardo entre el momento en que el elemento se carga y el comienzo de la secuencia de la animación.

### **animation-direction**

Indica si la animación debe retroceder hasta el fotograma de inicio al finalizar la secuencia o si debe comenzar desde el principio al llegar al final.

### **animation-duration**

Indica la cantidad de tiempo que la animación consume en completar su ciclo (duración).

### **animation-iteration-count**

El número de veces que se repite. Podemos indicar `infinite` para repetir la animación indefinidamente.

### **animation-name**

Especifica el nombre de la regla `@keyframes` que describe los fotogramas de la animación.

### **animation-play-state**

Permite pausar y reanudar la secuencia de la animación

### **animation-timing-function**

Indica el ritmo de la animación, es decir, como se muestran los fotogramas de la animación, estableciendo curvas de aceleración

### **animation-fill-mode**

Especifica qué valores tendrán las propiedades después de finalizar la animación (los de antes de ejecutarla, los del último fotograma de la animación o ambos).

## **RWD (Responsive Web Design)**

Cada vez es más frecuente acceder a Internet con diferentes tipos de dispositivos, que a su vez tienen diferentes pantallas y resoluciones, con distintos tamaños y formas, que hacen que se consuman las páginas webs de formas diferentes, apareciendo por el camino también diferentes necesidades, problemas y soluciones.

Hoy en día se le denomina Responsive Web Design (o RWD) a los diseños web que tienen la capacidad de adaptarse al tamaño y formato de la pantalla en la que se visualiza el contenido, respecto a los diseños tradicionales en los que las páginas web estaban diseñadas sólo para un tamaño o formato específico, y no tenían esa capacidad de adaptación.

Aunque en principio el concepto de web adaptativa es muy sencillo de comprender, aplicarlo puede ser todo un quebradero de cabeza si no se conocen bien las bases y se adquiere experiencia. En MediaQuerí.es puedes encontrar algunos ejemplos de páginas que utilizan Responsive Web Design para tener clara la idea.

### **Diferencias entre diseño responsive y diseño adaptativo**

El diseño web responsive y el diseño web adaptativo son conceptos muy parecidos, pero efectivamente no son lo mismo. Ambos son métodos de programación flexible que adaptan la web en función del dispositivo para asegurar su visualización. Entonces, ¿cuál es la diferencia?

La principal diferencia es que el diseño responsive se adapta literalmente. En lo que respecta al diseño adaptativo, se hace un diseño para cada dispositivo (varias resoluciones preestablecidas como ordenador de escritorio, tableta y móvil y todas sus casuísticas).

### **Diseño web responsive**

1. Reestructura todos los elementos de la web para optimizar todo el espacio y asegurar una buena experiencia de usuario, un excelente aspecto visual y la funcionalidad. Todo ello atendiendo a aspectos como la accesibilidad de los contenidos.
2. Un solo diseño que se adapta según la pantalla.
3. Utiliza tamaños proporcionales, en lugar de valores fijos en píxeles. Establece medidas en porcentajes.
4. Emplea media queries y diferentes hojas de estilo (CSS) para cada medida de pantalla.
5. Es muy flexible.
6. Puede requerir mayor tiempo de carga.



**Diseño web adaptativo**

1. Utiliza tamaños de pantalla fijos y preestablecidos para cada uno de los dispositivos.
2. Diferentes diseños de forma independiente en los que podemos variar lo que mostramos, aumentando la calidad de la experiencia de navegación del usuario.
3. Utiliza valores fijos en píxeles.
4. No necesita tanto código. Sencillez.
5. No es tan flexible. No se ajusta exactamente a cualquier resolución.
6. Menor tiempo de carga. El dispositivo recibe sólo lo necesario para su visualización.

## BIBLIOGRAFÍA

### 1. PÁGINAS WEB

Autor	Fecha	Título	Fuente	
No presenta	2016	HTML & CSS - W3C.	<b>Sitio.</b> W3.org	Consultado el 12 de Diciembre de 2021 <a href="https://www.w3.org/standards/webdesign/htmlcss">https://www.w3.org/standards/webdesign/htmlcss</a>
Páginas web	Viernes, Nov 19 2021	Editores de código: las mejores aplicaciones para sistemas como Windows o Mac	<b>Sitio.</b> IONOS Digitalguide	Consultado el 12 de Diciembre de 2021 <a href="https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/editor-de-codigo/">https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/editor-de-codigo/</a>
No presenta	Lunes, Dic 12 2021	Modelo–vista–controlador - Wikipedia, la enciclopedia libre	<b>Sitio.</b> Es.wikipedia.org	Consultado el 12 de Diciembre de 2021 <a href="https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/CSS_basics">https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/CSS_basics</a>
No presenta	Miércoles Jul 27 2016	Preprocesadores CSS, una forma diferente de crear nuestras hojas de estilos	<b>Sitio.</b> acens	Consultado el 12 de Diciembre de 2021 <a href="https://www.acens.com/comunicacion/white-papers/preprocesadores-css-una-forma-diferente-de-crear-nuestras-hojas-de-estilos/">https://www.acens.com/comunicacion/white-papers/preprocesadores-css-una-forma-diferente-de-crear-nuestras-hojas-de-estilos/</a>
Páginas web	Lunes Oct 12 2020	SASS: ¿CSS en el siguiente nivel?	<b>Sitio.</b> IONOS Digitalguide	Consultado el 12 de Diciembre de 2021 <a href="https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/sass/">https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/sass/</a>

No presenta	Viernes Dic 10 2021	Usando animaciones CSS - CSS   MDN	<b>Sitio.</b>  Developer.mozilla.org	Consultado el 11 de Diciembre de 2021 <a href="https://developer.mozilla.org/es/docs/Web/CSS/CSS_Animations/Using_CSS_animations">https://developer.mozilla.org/es/docs/Web/CSS/CSS_Animations/Using_CSS_animations</a>
Martínez, P	Martes En 24 2017	Diferencias entre diseño web responsive y adaptativo	<b>Sitio.</b>  woko   Agencia creativa de Estrategia digital	Consultado el 11 de Diciembre de 2021 <a href="https://woko.agency/blog/diseño-web-responsive-y-adaptativo-diferencias/">https://woko.agency/blog/diseño-web-responsive-y-adaptativo-diferencias/</a>