



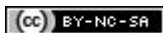
# Diseño de Interfaces Web

## 2º DAW

### Unidad de Trabajo 2

# Creación de interfaces web utilizando estilos

Francisco Jesús Gómez Romero, 2022/23



# Metodología

- **Cada tema:**

- Introducción
- Desarrollo → Ejercicios de clase
- Consolidación → Evaluación

- **Cada sesión:**

- Recordatorio de la clase anterior y objetivos de hoy
- 5 min de descanso entre horas

- **Cada punto del tema:** Explicaciones - Ejemplos - Ejercicios/Actividades

- **Trabajo de los alumnos:**

- Autónomo y colaborativo
- Práctico más que teórico



# Evaluación



Actividades de aula

AP



Proyecto

50%



Examen

50%

$\geq 5$

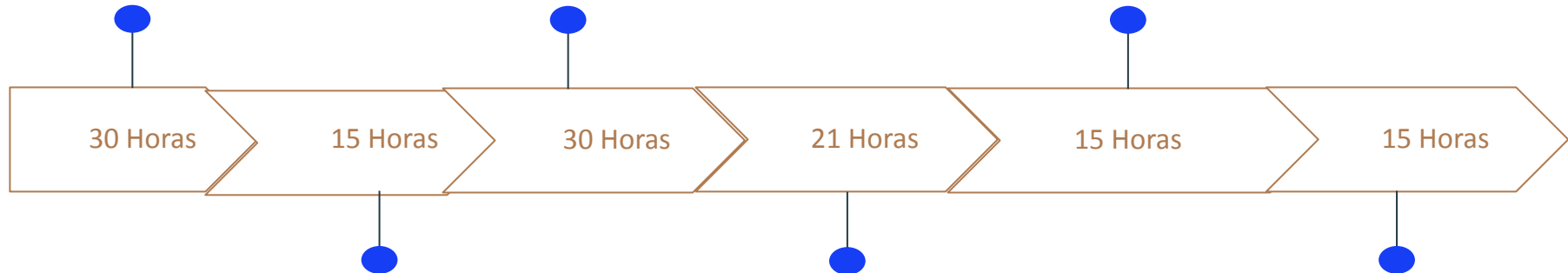
# Secuenciación

Total: 126 horas

UT1: Planificación de Interfaces Gráficas

UT3: Implantación de contenido multimedia

UT5: Desarrollo de Webs Accesibles



**UT2: Creación de interfaces web utilizando estilos**

UT4: Integración de contenido interactivo

UT6: Desarrollo de interfaces Web amigables

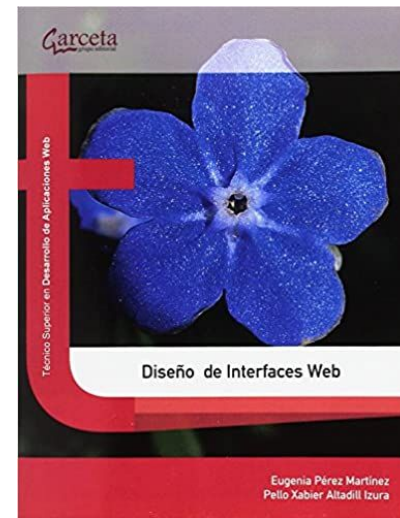
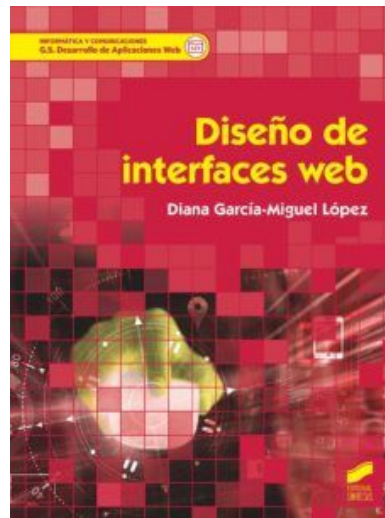
# Resources

## Bibliografía

- Ed. SINTESIS
- Ed. Garceta
- Ed. Ra-Ma

## Material de clase

- Diapositivas
- Documentación suministrada por el profesor
- Referencias de ampliación y refuerzo
- Actividades y **apuntes**
- Glosario



# Índice de contenidos

## Introducción a CSS

- Concepto y evolución de CSS

- Formas de aplicar estilo a un documento HTML

## Lenguaje CSS

- Sintaxis

- Selectores

- Propiedades

## Modelo de cajas

## Herencia

## Pseudoclasas y pseudoelementos

## CSS Layout

- Display

- Max-width

- Position

- Superposición (z-index)

- Overflow

- Flotación

- Alineación

- Box-sizing y resize

## Medios

- Herramientas CSS

- Buenas prácticas

- CSS 3

- Bootstrap

- CSS Avanzado

- YUI

- SASS

# Intereses y expectativas

¿Qué espero aprender en esta unidad?



## RA2

**Crea interfaces Web homogéneas definiendo y aplicando estilos.**

- a) Se han reconocido las posibilidades de modificar las etiquetas HTML.
- b) Se han definido estilos de forma directa.
- c) Se han definido y asociado estilos globales en hojas externas.
- d) Se han definido hojas de estilos alternativas.
- e) Se han redefinido estilos.
- f) Se han identificado las distintas propiedades de cada elemento.
- g) Se han creado clases de estilos.
- h) Se han utilizado herramientas de validación de hojas de estilos.
- i) Se ha utilizado y actualizado la guía de estilo.



¿Cómo podemos aplicar  
**estilos** a nuestras webs?

## Evaluación Conocimientos Previos

Quiz w3school CSS

[https://www.w3schools.com/css/css\\_quiz.asp](https://www.w3schools.com/css/css_quiz.asp)

## Actividad 2.1

Busca ejemplos de  
contenidos  
desarrollados con  
CSS y compártelos  
con la clase



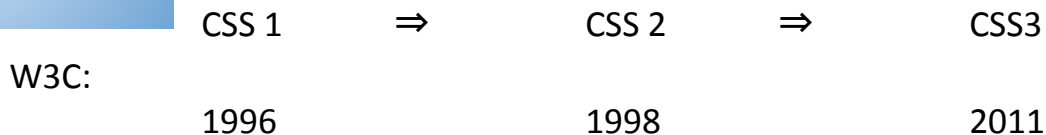
- 🔍 mortgage comparison canada
- 🔍 mortgage comparison canada - Google Search
- 🔍 mortgage comparison calculator canada
- 🔍 mortgage comparison tool canada
- 🔍 mortgage rate comparison ontario canada
- 🔍 bank mortgage rates comparison canada
- 🔍 commercial mortgage rates comparison canada
- 🔍 how to compare mortgage
- 🔍 which is the best mortgage comparison site
- 🔍 how to compare mortgage deals
- 🔍 canada mortgage rate comparison chart

MacBook Pro

## ¿Qué es CSS?

- Cascading Style Sheets (Hojas de estilo en cascada)
- Sirve para dar estilo al documento HTML ⇒ Cómo se va a ver el HTML
- Permite separar los estilos del contenido
  - Reutilización ⇒ Aplicar el mismo estilo a todas las páginas
  - Mantenimiento ⇒ Si tengo que modificar algo solo lo hago en un sitio
  - Simplificación ⇒ Archivos menos cargados de código
- Aplicar diferentes estilos a un mismo documento: [ejemplo](#)
- [Estándar W3C](#)
- También aporta:
  - Acelerar el desarrollo
  - Diseño para varios dispositivos: media queries

## Historia



¿Qué diferencias encuentras entre cada versión?

Nivel	Año	Descripción
CSS1	1996	Se crean propiedades para tipografías, colores, alineación, etc...
CSS2	1998	Propiedades de posicionamiento, tipos de medios, etc...
CSS2.1	2005	Corrige errores de CSS2 y modifica ciertas propiedades
CSS3	2011	Se comienzan a crear características de CSS como módulos independientes

## ¿Cómo aplicamos CSS en nuestra página?

- En línea:

```
<p style="text-align:left">This is some text in a paragraph.</p>
```

- Interna:

```
<head>
...
<style>
  body {
    background-color: aquamarine;
  }
</style>
</head>
```

- Hoja de estilo externa (con link o con @import):

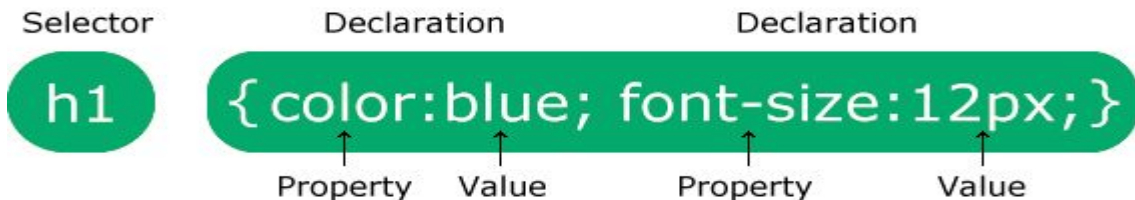
```
<head><link rel="stylesheet" href="nombre.css" ></head>
```

[HTML link tag](#)

- ```
<style>@import url ("estilo.css")screen</style>
```

## ¿Cómo indicamos los estilos de los elementos HTML?

- Selector ⇒ Partes del documento a las que afecta la regla
- Regla ⇒ Cómo se verán los elementos seleccionados por el selector
  - Propiedad : Valor;



### Notas:

- En general No sensible a mayúsculas/minúsculas
- El “;” de la última propiedad se puede omitir
- Comentarios: /\* This is a CSS comment \*/



## Ejercicio 1

Crea un documento Ejercicio1.html con un encabezado de nivel 1 y un párrafo. Dale un contenido a cada uno y luego aplícales un estilo utilizando CSS “en línea” para representarlos como se indica:

- El encabezado en verde
- El párrafo en rojo.

## Ejercicio 2

Haz una copia del fichero del ejercicio 1 con nombre Ejercicio2.html en el que cambiarás la forma de incluir código CSS para hacerlo con las etiquetas `<style>` de manera interna.

¿Qué ventaja encuentras en hacerlo de esta manera? Coméntalo dentro de las etiquetas `<style>` utilizando comentarios CSS.

## Ejercicio 3

Haz una copia del fichero del ejercicio 2 con nombre Ejercicio3.html en el que cambiarás la forma de incluir código CSS para hacerlo usando un fichero CSS externo.

¿Qué ventaja encuentras en hacerlo de esta manera con respecto a la del ejercicio 2? Coméntalo dentro de las etiquetas `<style>` utilizando comentarios CSS.

## Tipos de Selectores

Selector de:

- Elemento  $\Rightarrow$  p, h1, ...
- Clase  $\Rightarrow$  .clase
- Identificador  $\Rightarrow$  #id
- Universal  $\Rightarrow$  \*
- Atributo  $\Rightarrow$  [atributo]

[atributo = "valor"]

```
p{  
  
    propiedad:valor;  
  
}
```

```
#identificador{  
  
    propiedad:valor;  
  
}  
  
<p id="identificador"></p>  
  
<p id="identificador2"></p>
```

```
.clase{  
  
    propiedad:valor;  
  
}  
  
<p class="clase"></p>  
  
<p class="clase2"></p>
```

## Ejercicio 4

1. Crea un archivo **HTML** llamado `index.html`. → <https://codeshare.io/MkZXde>
2. Crea un archivo **CSS** llamado `styles.css`.
3. Conecta el archivo `styles.css` a `index.html` usando la etiqueta `<link>` en el `<head>`.

En el archivo `styles.css`, agrega el código para aplicar estilos usando los diferentes selectores:

1. **Selector de Elemento:** Cambia el color del texto de todos los párrafos a azul.
2. **Selector de Clase:** Agrega un fondo amarillo al párrafo con la clase `importante`.
3. **Selector de ID:** Cambia el color del texto y el tamaño de fuente (*font-size*) del título con `id="titulo"` [atributo = "valor"]
4. **Selector Universal:** Elimina los márgenes (*margin*) y el relleno (*padding*) de todos los elementos.
5. **Selector de Atributo:** Aplica un borde rojo al campo de entrada de texto (`input`) que tenga `type="text"`. (*border: 2px solid red;*)

Elemento ⇒ p, h1, ...

Clase ⇒ .clase

Identificador ⇒ #id

Universal ⇒ \*

Atributo ⇒ [atributo]

### Ejemplo de Selectores Básicos en CSS

Este párrafo está estilizado con el **selector de elemento**, cambiando su color a azul.

Este párrafo usa el **selector de clase** para resaltar el fondo en amarillo.

Nombre:

Email:



# Ejercicio 5

Usando el código proporcionado por el profesor: <https://codeshare.io/ZLrj3o> completa el CSS con los selectores adecuados

No olvides enlazar el css con el html.

---

Subtítulo de la Página

Este párrafo tiene que destacarse visualmente.

Elemento de lista 1

Elemento de lista 2

Elemento de lista 3

Usuario:

Contraseña:

# Ejercicio 6



1. Crear un documento HTML que contenga 4 encabezados, todos serán de tipo h1.
2. A cada encabezado habrá que darle un color distinto. Los que ocupen una posición par serán rojos y los impares verdes.
3. Finalmente, haz que el primero tenga color azul y el resto sigan igual que en el punto 2.

**Este encabezado h1 será de color azul**

**Este encabezado h1 será de color rojo**

**Este encabezado h1 será de color verde**

**Este encabezado h1 será de color rojo**

# Selectores Avanzados



Avanzados (ya los veremos):

- Pseudoclase  $\Rightarrow$  selector:pseudoclase
- ...

[https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_selectors](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_selectors)



# Selectores combinadores



## Según la relación entre elementos

- Multiples  $\Rightarrow$  Separados por comas (Ej: p, h1, h2 {...})
- Descendentes  $\Rightarrow$  Separado por espacios (Ej: div p {...})
- Hijos directos  $\Rightarrow$  selector1 > selector2 {...} /\* No aplica a los nietos, etc.. \*/
- Dependientes  $\Rightarrow$  Seguidos sin espacios (Ej: p.miClase {...})
- Adyacentes  $\Rightarrow$  selector1 + selector2 {...} /\* Aplica a los hermanos \*/
- Todos los hermanos  $\Rightarrow$  div ~ p

# Ejercicio 7: selectores combinadores

1. Crea un documento que contenga un body como el siguiente: <https://codeshare.io/Ek0yD8>

## Sección 1

Este es un párrafo de introducción.

Otro párrafo en la sección 1.

## Sección 2

Primer párrafo de la sección 2.

Segundo párrafo en la sección 2.

2. Usa el combinador de **descendiente** para que todos los párrafos dentro de `.container` tengan texto azul.
3. Aplica el combinador de **hijo directo** para que solo el `<h2>` dentro de `.container` sea rojo y no el de dentro de `<section>`.
4. Usa el combinador de **hermano adyacente** para que el primer `<p>` que sigue a un `<h2>` sea verde.
5. Aplica el combinador de **hermano general** para que todos los párrafos después de `.intro` tengan un fondo amarillo.
6. Usa un **selector múltiple** para aplicar el mismo estilo (margen inferior de 10px) a los elementos `<h2>` y `.intro`.
7. Aplica un **selector dependiente** para que cualquier `<p>` dentro de `.container` que también tenga la clase `.intro` tenga un color de fondo gris.

# Ejercicio 8: selectores



A partir del código HTML y CSS que se muestra, añadir los selectores CSS que faltan para aplicar los estilos deseados. Cada regla CSS incluye un comentario en el que se explica los elementos a los que debe aplicarse:

Código: <https://codeshare.io/zlnoMD>

## Unidades absolutas

px	Píxeles. Útil cuando queremos un tamaño fijo que no se redimensione con la página.
in	Pulgadas (1 pulgada = 2.54 cm)
cm	Centímetros
mm	Milímetros
pt	Puntos (1 pt = 1/72 pulgadas)
pc	Picas (1 pica = 12 puntos)

## Unidades relativas

em	Relativo al tamaño de la fuente del elemento ( 2 em significa 2 veces el tamaño de la fuente actual)
rem	Se basa en el tamaño de fuente del elemento raíz (HTML). Por defecto 16px
%	Porcentaje (relativo al elemento padre)
vh y vw	Medidas relativas de acuerdo al viewport 1vh = 1% de la altura del viewport 100vh = altura del viewport
fr	Flexible Grid Units (fr) Se utiliza en Grid Layout y representa una fracción del espacio disponible en un contenedor

- Centrar un texto según el tamaño del viewport (con line-height)
- vh y vw: Centrar cualquier elemento según el ancho del viewport (con propiedad transform)
- Diseño de footer fijo en la parte inferior del viewport solo si hay poco contenido

# Unidades de medida: Ejercicio 9



Completa los huecos (\_\_\_) en el CSS con las unidades que se indican en los comentarios:

- `.vh-box`: Completar con `50vw` y `30vh`.
- `.vw-box`: Completar con `80%` y `200px`.
- `.percent-box`: Completar con `60%`, `200px` y `500px`.




Abre el archivo HTML en el navegador y redimensiona la ventana para ver cómo responde cada bloque a los cambios de tamaño.

## Preguntas para Reflexionar:

- ¿Cómo afecta cada unidad (vh, vw, %) al diseño cuando se cambia el tamaño de la ventana?
- ¿Qué diferencias observas entre el comportamiento del `.vh-box` y el `.vw-box`?
- ¿Cómo limita el tamaño del bloque `.percent-box` la propiedad `max-width`?

<https://codeshare.io/3yNdPB>

## Colores básicos

Color	Nombre	HEX	RGB
	black	#000000	0,0,0
	white	#ffffff	255,255,255
	red	#ff0000	255,0,0
	blue	#0000ff	0,0,255
	yellow	#ffff00	255,255,0
	gray	#808080	128,128,128
	green	#008000	0,128,0

HTML soporta alrededor de 140 nombres de colores diferentes



# Propiedades: colores y fondo



- **Nombre predefinido:** red, yellow, blue, green...
- **Código hexadecimal:** #RRGGBB o #RRGGBBAA
- **RGB:** Red, Green, Blue
- **RGBA:** Red, Green, Blue, Alpha
- **HSL:** Hue, Saturation, Lightness
- **HSLA:** Hue, Saturation, Lightness, Alpha

# Propiedades: colores y fondo



Propiedad	Descripción	Valores
<code>color</code>	Color del texto	RGB   HSL   HEX   nombre del color   RGBA   HSLA
<code>background-color</code>	Color de fondo	RGB   HSL   HEX   nombre del color   RGBA   HSLA   transparent
<code>background-image</code>	Imagen de fondo	url(«...»)   none
<code>background-repeat</code>	Repetición de la imagen de fondo	repeat   repeat-x   repeat-y   no-repeat
<code>background-attachment</code>	Desplazamiento de la imagen de fondo	scroll   fixed
<code>background-position</code>	Posición de la imagen de fondo	percentage   length   left   center   right
<code>background-size</code>	Tamaño de la imagen de fondo	auto   cover   contain   valor
<code>Opacity</code>	Transparencia de un elemento	[ 0 – 1 ] (0 → totalmente transparente)

# Propiedades: colores y fondo: Ejercicio 10



En cada sección del CSS (`.color-box`, `.image-box`, `.gradient-box`, `.multiple-box`, `.text-box`), completa los huecos (\_\_\_) con los valores que se indican en los comentarios:

- En `.color-box`, añade un `background-color` con un color hexadecimal (por ejemplo, `#3498db`) o un nombre de color (como `blue`).
- En `.image-box`, aplica una imagen de fondo y ajusta las propiedades de repetición (`background-repeat`), tamaño (`background-size`), y posición (`background-position`).
- En `.gradient-box`, usa `linear-gradient` para aplicar un gradiente de color de un lado al otro.
- En `.multiple-box`, combina una imagen y un gradiente como fondo múltiple, ajustando cada capa individual.
- En `.text-box`, cambia el color del texto y usa `rgba()` en `background-color` para darle opacidad.
- Redimensiona la ventana para ver cómo se ajustan los fondos y los colores.
- Experimenta cambiando los valores para ver cómo afectan cada bloque de manera visual.

<https://codeshare.io/zlnmPk>