

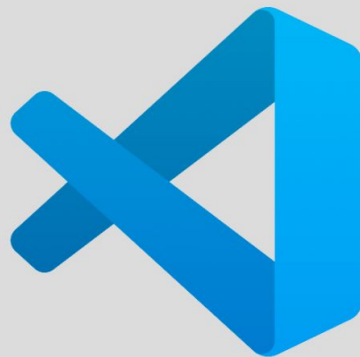
# 03. DISEÑO Y MAQUETACIÓN WEB CON HTML5 Y CSS3



**Las Fuentezuelas**  
INSTITUTO DE EDUCACIÓN SECUNDARIA

**DISEÑO DE  
INTERFACES WEB**  
2º DAW

- 01:** Configuración del entorno de desarrollo
- 02:** Estilos avanzados
- 03:** Preprocesadores CSS



Visual Studio Code

**01:**

**Configuración del entorno de desarrollo**

**El contenido de este apartado se encuentra desarrollado en la plataforma Moodle Centros.**

- Google Chrome y sus extensiones.
- Visual Studio Code.
- Git.
- GitHub.
- Trabajar con repositorios Git en Visual Studio Code.





**02:**

**Estilos avanzados**

## 01.01. Buenas prácticas CSS

### 1. Organiza la estructura de arriba hacia abajo.

```
/****** generic classes*****/  
styles goes here...  
/****** header *****/  
styles goes here...  
/****** nav menu *****/  
styles goes here...  
/****** main content *****/  
styles goes here...  
/****** footer *****/
```

### 2. Nombra correctamente los selectores.

Evitar: #1div, .=div, DivContent

Mejor utilizar: #div1, .div, divContent

## 01.01. Buenas prácticas CSS

### 3. Separa las palabras mediante guiones o mayúsculas.

```
/* Opción 1: Palabras separadas por mayúsculas */
navMenu { padding: 2em; border: 2px solid green; }

/* Opción 2: Palabras separadas por guiones */
nav-menu { padding: 2em; border: 2px solid green; }
```

### 4. Haz legible el CSS.

```
/* Opción 1: Estilos en una línea */
.nav-menu { padding: 2em; border: 2px solid green; }

/* Opción 2: Cada estilo en una línea */
.nav-menu {
    padding: 2em;
    border: 2px solid green;
}
```

## 01.01. Buenas prácticas CSS

### 5. Combina elementos.

- Si varios elementos comparten propiedades:
  - Compartirlas.
  - No repetir código.

```
h1, h2, h3 { font-family: Arial; font-weight: 700; }
```

### 6. Utiliza selectores descendientes.

- En vez de crear un selector de clase o de identificador.
  - Mejora la comprensión.

```
selector1 selector2 selectorN{  
    propiedad: valor;  
}
```



## 01.01. Buenas prácticas CSS

### 7. Utiliza las propiedades abreviadas.

```
/* Propiedades margin-left, margin-right y margin-top */  
.nav-menu {margin-left: 5px; margin-right: 5px; margin-top: 5px;}  
/* Propiedad abreviada margin */  
.nav-menu {margin: 5px 5px 0px 5px;}
```

### 8. Utiliza nombres descriptivos en los selectores.

```
.nav-button { background: blue; } /* Estilo del botón de la navbar*/
```

### 9. No utilices como nombre de selector una característica visual.

- Si se modifica la característica (color, tamaño, posición, etc.), se debería de cambiar el nombre del selector.

```
/* Selector con nombre que define la característica visual del color */  
.menu-red { background: red; }  
/* Utiliza mejor: */  
.nav-menu { background: red; }
```

## 01.01. Buenas prácticas CSS

### 10. Prueba el diseño en diferentes navegadores.

- Chrome, Firefox, Opera, Safari, IExplorer (Edge).
- Aplicación [browserling](#).

### 11. Validar el CSS.

- Detecta errores de CSS.
  - Extensión de Visual Studio Code.
  - [Herramienta de validación de W3C](#).

### 12. Agrega los prefijos de los navegadores en propiedades no estables.

- Extensión Autoprefixer de Visual Studio Code.

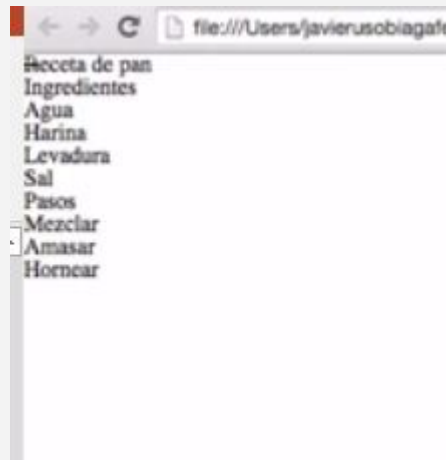
## 01.02. Reset CSS

- Los navegadores tienen una serie de estilos por defecto.
- Permiten mostrar HTML cuando no hay CSS definido
  - h1, h2, h3..., Márgenes, Listas, Etc.
- Se conoce con el nombre de **user agent stylesheet**.



## 01.02. Reset CSS

- Los estilos del navegador se pueden sobrescribir:
  - Corrigiendo línea por línea los elementos definidos por defecto
  - Hacer un **reset CSS**
    - Listado de código al inicio del documento
    - Resetea los estilos por defecto del navegador
      - Margin, padding, border
      - Tipografía
      - Etc.
    - Empieza desde cero a aplicar estilos.
    - [Web con Reset CSS](#)
    - [HTML5 Reset CSS](#)



## 01.03. Selectores avanzados

### Pseudoclasas

- Hacen referencia al cambio de estado de un elemento.
  - Ej.: Al pasar el ratón sobre un enlace

```
a{  
  color: blue;  
  text-decoration: none;  
}
```

- Las pseudoclasas de un elemento se indican con ‘:’
  - Para la etiqueta <a>: hover, active, visited, focus

```
a:hover{  
  text-decoration: underline;  
}
```

```
7 a,  
8 a:visited{  
9   color: blue;  
10  text-decoration: none;  
11 }  
12  
13 a:hover{  
14   text-decoration: underline;  
15 }  
16  
17 a:active{  
18   color: orange;  
19 }  
20  
21 /*  
22 a:focus{  
23   background-color: lightgrey;  
24 }  
25 */
```

## 01.03. Selectores avanzados

### Pseudoelementos

**E::first-line**{...}

**E::first-letter**{...}

Combinados con la propiedad content

**E::before**{...}

**E::after**{...}

Texto seleccionado por el usuario

**E::selection**{...}

Selecciona el hijo enésimo del elemento (sea del tipo que sea)

**E:nth-child(numero)**{...}

**E:nth-last-child(numero)**{...}

**E:first-child(numero)**{...}

**E:last-child(numero)**{...}

Selecciona el elemento sin hijos

**E:empty**{...}

Selecciona los hijos enésimo de un mismo tipo

**E:nth-of-type(numero)**{...}

**E:nth-last-of-type(numero)**{...}

```
<body>
```

```
  <h1>Titular de la Noticia</h1>
```

```
  <h2>Primer encabezado</h2>
```

```
  <p>TEXT0</p>
```

```
  <h2>Segundo encabezado</h2>
```

```
  <p>MAS TEXTO</p>
```

```
  <div>Fotito
```

```
    <h2>Tercer encabezado</h2>
```

```
  </div>
```

```
</body>
```

## 01.03. Selectores avanzados

### Otros selectores

```
<body>

  <h1>Titular de la Noticia</h1>
  <h2>Primer encabezado</h2>
  <p>TEXT0</p>
  <h2>Segundo encabezado</h2>
  <p>MAS TEXTO</p>
  <div>Fotito
    <h2>Tercer encabezado</h2>
  </div>
</body>
```

- **Adyacentes**  
h1 + h2 {  
border: 1px solid blue;  
}
- **Hermanos**  
h1 ~ h2 {  
border: 1px solid green;  
}

## 01.03. Selectores avanzados

```
<ul>
  <li><a href="google.co.uk" lang="en-GB" rel="Search Engine"> Google UK</a></li>
  <li><a href="google.es" lang="es-ES" rel="Buscador"> Google ES</a></li>
  <li><a href="google.fr" lang="fr-FR" rel="Un Francés"> Google FR</a></li>
</ul>
```

- Matching exacto:

```
a[rel='Un Francés']{
  color:blue;
}
```

- Matching contains:

```
a[rel~='Buscador']{
  color:blue;
}
```

- Comienza por:

```
a[href^="goo"]{
  font-family: 'Open Sans', sans-serif;
}
```

- Finaliza por:

```
a[href$="co.uk"]{
  font-family: Verdana, sans-serif;
}
```

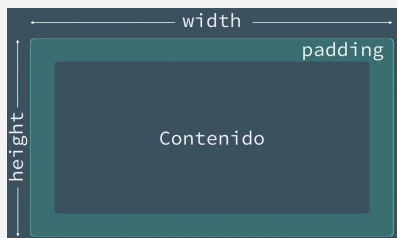
- Contiene:

```
a[href*="google"]{
  font-family: Impact, sans-serif;
}
```



## 01.04. Modelo de caja: problema

- Propiedades de la caja
  - Anchura (width)
  - Altura (height)
  - Espacio entre el contenido y el borde (padding)
  - Espacio fuera de la caja y alrededor (margin)



- En el modelo de caja, la anchura total del elemento es:
  - $\text{Anchura total} = \text{width} + \text{padding} + \text{border}$

## 01.04. Modelo de caja: problema

- ¡¡PROBLEMA!!
  - Diseños responsive
  - Uso de anchuras porcentuales: columna al 50% con padding
    - ¿Qué ocurrirá?
- SOLUCIÓN
  - Resetear el modelo de caja al principio del CSS
  - Anchura total de la caja = propiedad width definida

```
*{  
  box-sizing: border-box;  
}  
  
anchura total = width
```

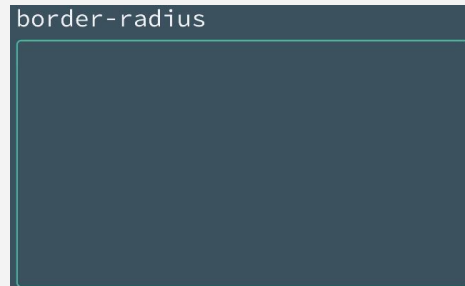
## 01.05. Bordes redondeados

- Para establecer esquinas redondeadas se emplea la propiedad **border-radius**
- Acepta valores por cada esquina:

```
.caja{  
  border-radius: 10px 15px 20px 50px;  
}
```

- Se puede dividir en varias propiedades:

```
.caja{  
  border-top-left-radius: 10px;  
  border-top-right-radius: 15px;  
  border-bottom-right-radius: 20px;  
  border-bottom-left-radius: 50px;  
}
```



## 01.05. Bordes redondeados

- La propiedad border-radius lo que hace en realidad es:
  - Aplicar redondeo a las esquinas
  - Tiene como base un círculo
  - El círculo tiene como radio el valor de la propiedad
    - CSS permite asignar dos valores distintos □ Elipse



```
.example{  
  border-radius: 10px/30px;  
}
```

## 01.05. Bordes redondeados

- Propiedad **border-radius**
  - Aplicación a las 4 esquinas del elemento

```
.example{  
  border-radius: 10px;  
}
```

- Un valor distinto a cada esquina
  - Desde la esquina superior izquierda □ En sentido horario

```
.example{  
  border-radius: 5px 10px 5px 20px;  
}
```

- Un valor a:
  - Esquina superior izquierda e inferior derecha.
  - Esquina superior derecha e inferior izquierda.

```
.example{  
  border-radius: 10px 20px;  
}
```

## 01.05. Bordes redondeados

- Trucos de border-radius:
  - Convertir cuadrado en círculo.

```
.example{  
  width: 100px;  
  height: 100px;  
  border-radius: 50%;  
}
```

## 01.06. Colores y opacidad

- Se añade el **canal alfa** a la misma definición del color en formato **RGB**.

```
.example{  
  color: rgba(0, 0, 0, .5);  
}
```

- Otra forma de definir colores:

- **hsl**

- Tono: 0-360 (rueda de color)
- Saturación (%):
  - 0%: gris medio
  - 100%: color completamente saturado
- Luminosidad (%):
  - 0%: negro
  - 50%: color puro
  - 100%: blanco

- **hsla**

- Permite indicar canal alfa

- Conversión de colores en distintos formatos:

- [www.hslpicker.com](http://www.hslpicker.com)



```
.example{  
  color: hsla(125, 100%, 50%, .5);  
}
```

## 01.07. Sombras

- Valores:
  - Desplazamiento de sombra en eje X (horizontal)
  - Desplazamiento de sombra en eje Y (vertical)
  - Difuminado de la sombra
  - Color de sombra: en todas sus variantes

```
.example{  
  box-shadow: 5px 5px 5px black;  
}
```

```
.example{  
  box-shadow: 5px 5px 5px  
              rgba(0,0,0,.5);  
}
```



## 01.07. Sombras

- Propiedad **inset** de box-shadow
  - Renderiza la sombra por dentro del elemento
  - Permite definir un efecto de resaltado
    - Con color blanco
    - Sin desplazamiento horizontal
- Valor **spread** (espaciado)
  - Antes del valor de color
  - Espacio de la sombra antes de difuminar
  - Comienza desde el centro del elemento
- Efecto de falso borde
  - Con desplazamientos 0

```
.caja{  
  box-shadow: inset 5px 5px 10px 8px  
             black;  
}
```



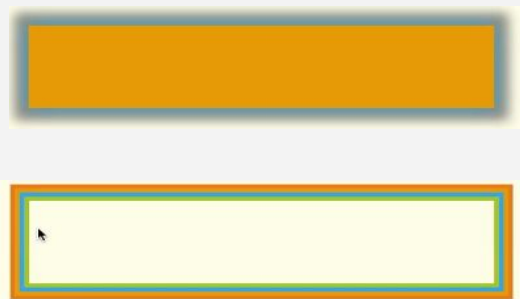
```
.caja{  
  box-shadow: X Y difuminado  
             espaciado color;  
}
```



## 01.07. Sombras

- Aplicación de varias sombras a un elemento
  - Se separa la definición de las sombras con una ‘,’
  - Cuidado con los difuminados definidos.

```
.caja{  
  box-shadow: 5px 5px 10px black,  
             -5px -5px 10px red;  
}
```



## 01.07. Sombras

- Se puede aplicar sombra a tipografías de texto (**text-shadow**)
  - Similar a box-shadow
  - No admite inset ni propiedad spread
- Se pueden combinar varias sombras a la vez
  - Para efectos 3D.

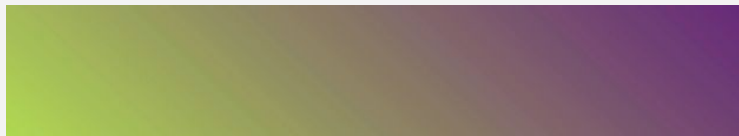
```
.titulo{  
  text-shadow: 5px 5px 0  
              rgba(0,0,0,.5);  
}
```



## 01.08. Degradados y fondos

- El degradado se trata como una imagen de fondo.
- Se debe incluir siempre en la propiedad background-image o background.
- Valores:
  - Dirección del degradado: en grados
    - Si no se define, degrada de arriba hacia abajo
  - Color inicial.
  - Color final.

```
.example{  
  background-image:  
    linear-gradient(45deg,red,black);  
}
```



## 01.08. Degradados y fondos

- Combinando el color de fondo de la propiedad background con degradado
  - Efecto de relieve



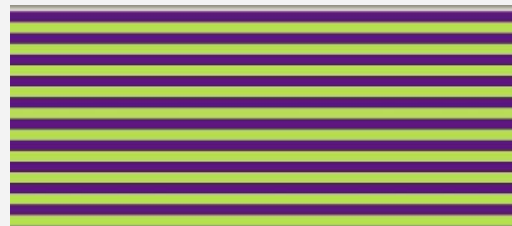
- Se permiten degradados de más de un color
  - Se puede indicar en qué porcentaje se cambia de color.
  - Ej.: background: linear-gradient(red, green 40%, green 60%, red)



## 01.08. Degradados y fondos

- Se pueden combinar con la propiedad background-size
  - Controla el tamaño de las imágenes de fondo
  - Valores:
    - Ancho de imagen
    - Alto de imagen
  - Con el valor auto se consigue la relación proporcional de aspecto de la imagen.
  - Ej.: background: linear-gradient(red, green 40%, green 60%, red)
  - background-size: 100px 20px;

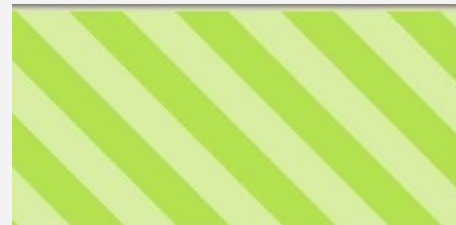
```
.example{  
  background-size:100px auto;  
}
```



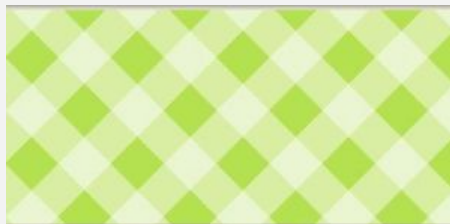
## 01.08. Degradados y fondos

- Se puede combinar con la propiedad **background-image**.
  - Definiendo la propiedad **repeating-linear-gradient**
  - Se puede controlar la dirección del degradado como un degradado normal

```
.example{  
  background-image:  
    repeating-linear-  
      gradient(red 20px,black 40px);  
}
```



- En CSS3 se pueden declarar más de una imagen de fondo en el mismo elemento separado por ','
  - Se puede hacer lo mismo con los degradados.



## 01.08. Degradados y fondos

- Se pueden definir degradados circulares
  - Con la función **radial-gradient**
  - Valores
    - Forma
      - Por defecto se adapta a la forma del elemento contenedor
        - Elemento rectangular □ Degradado elipse
    - Color inicial
    - Color final
  - Se puede controlar la posición de comienzo del degradado
    - Indicando coordenadas (x,y)
    - O posiciones (top, bottom, left, right)



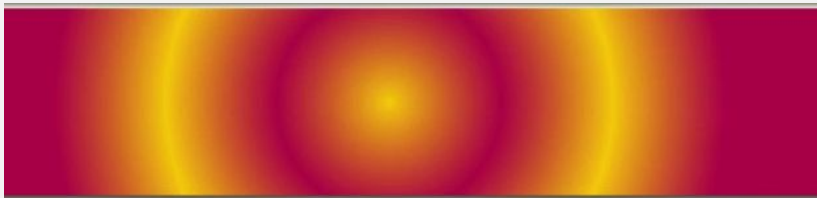
```
.example{  
  background-image:  
    radial-gradient(circle at 10%  
      20%, red, black);  
}
```





## 01.08. Degradados y fondos

- Se puede usar varios degradados en los degradados circulares
  - Hay que indicar los cortes del color en porcentaje (%)



- Como en los degradados lineales se puede combinar con otras propiedades y funciones:
  - background-size
  - repeating-radial-gradient

## 01.09. Webfonts

- Las tipografías propias (no las del sistema) se conocen como webfonts.
- Para incluirlas: declaración **@font-face**
  - Se permite definir varios formatos para navegadores distintos
    - woff, truetype

```
@font-face{  
  font-family: Lobster;  
  src: url(fonts/lobster.woff)  
       format("woff"),  
       url(fonts/lobster.ttf)  
       format("truetype");  
  font-weight: bold;  
  font-style: normal;  
}
```

- A partir de ese momento, ya se puede utilizar en los elementos CSS

```
h1{  
  font-family: Lobster, sans-serif;  
}
```

## 1.10. Transiciones en CSS

- Las **transiciones** permiten aplicar un cambio de estilo gradual a los elementos.
- Manera más sencilla de animar elementos en CSS.
- Ofrecen la ventaja de especificar el tiempo para que se produzca la transformación entre estilos
  - Efecto de animación.
- **Parámetros** para aplicar las transiciones se pueden establecer
  - En una sola línea
  - Propiedades por separado

```
transition:[propiedad a modificar] [Duración] [Tipo de animación] [Retardo];
```

```
transition: width 2s ease-out 0.5s;
```

## 1.10. Transiciones en CSS

### Propiedades

- transition-property → propiedad que se transiciona
  - Cuando al elemento con la clase transicionable se le aplique la clase transicionado, el color de fondo cambiará.

```
.transicionable{
  transition-property: background-color;
}

.transicionado{
  background-color: #FABADA;
}
```

- transition-duration → segundos que dura la transición.

```
.transicionable{
  transition-property: background-color;
  transition-duration: 2s;
}

.transicionado{
  background-color: #FABADA;
}
```

## 1.10. Transiciones en CSS

### Propiedades

- transition-timing-function
  - linear: velocidad uniforme.
  - ease: acelerada al inicio, luego se retarda y se acelera al final.
  - ease-in: empieza lentamente y va aumentando.
  - ease-out: empieza muy rápida y va descendiendo.
  - ease-in-out: empieza y acaba lentamente, en la parte central la velocidad es más rápida.

```
.transicionable{  
  
    background-color: blue;  
  
    transition-property: background-color;  
    transition-duration: 2s;  
    transition-timing-function: linear;  
}  
  
.transicionable:hover{  
    background-color: #FABADA;  
}
```

## 1.10. Transiciones en CSS

### Propiedades

- transition-delay
  - Tiempo (en segundos) que el navegador espera antes de poner en marcha la animación.

```
.transicionable{  
    transition-delay: 1s;  
}
```

## 1.10. Transiciones en CSS

- Propiedades modificables con transiciones ([enlace](#))

<input type="checkbox"/> all	<input type="checkbox"/> left	<input type="checkbox"/> word-spacing
<input type="checkbox"/> background-	<input type="checkbox"/> right	<input type="checkbox"/> letter-spacing
<input type="checkbox"/> color	<input type="checkbox"/> box-shadow	<input type="checkbox"/> fill
<input type="checkbox"/> border	<input type="checkbox"/> width	<input type="checkbox"/> padding
<input type="checkbox"/> border-radius	<input type="checkbox"/> height	<input type="checkbox"/> stroke
<input type="checkbox"/> color	<input type="checkbox"/> line-height	<input type="checkbox"/> text-shadow
<input type="checkbox"/> top	<input type="checkbox"/> margin	<input type="checkbox"/> vertical-align
<input type="checkbox"/> bottom	<input type="checkbox"/> opacity	<input type="checkbox"/> visibility
		<input type="checkbox"/> z-index

## 1.10. Transiciones en CSS

Se permite transicionar 2 o más propiedades CSS separándolas con coma, mediante la propiedad *transition* o *transition-property*.

```
.selector {  
  transition: background-color 1s ease-out,  
             color 1s ease-out;  
  
  /* O tambien */  
  transition-property: background, color;  
  transition-duration: 1s;  
  transition-timing-function: ease-out;  
}
```



## 1.11. Transformaciones en CSS

Las **transformaciones** permiten rotar, torcer, escalar o desplazar los elementos de la página web.

- Interesantes para realizar todo tipo de efectos visuales.
- Propiedades:
  - `transform` → la posición de origen para la transformación es el eje central del elemento.
  - `transform-origin` → la posición de origen para la transformación es el lado superior izquierdo del elemento por defecto.

## 1.11. Transformaciones en CSS

### Tipos

- scale → modifica el tamaño
- translate → cambia la posición (izquierda, derecha, arriba o abajo)
- rotate → gira el elemento en grados
- skew → distorsiona
- matrix → mueve con precisión de píxel

Lo ideal es aplicar transformaciones junto a las transiciones.

## 1.12. Paralaje

El **efecto de paralaje** (parallax) es la simulación de 3D.

- Dependiendo de la lejanía del objeto respecto al observador, éste se desplaza a diferente velocidad.
- Es el efecto de mirar a través de la ventana de un tren, los objetos que se encuentran próximos a nosotros parecen desplazarse a una velocidad mayor, mientras que los más lejanos apenas se mueven.

### Ejemplos

- [Teapot-creation](#)
- [Boy-Coy](#)

## 1.12. Paralaje

### Maneras de implementarlo

- Empleando JS
  - Búsqueda en Internet de efecto de paralaje JS.
- Empleando CSS puro ([vídeo](#))

## 1.13. Herramientas útiles y recursos

### **Caniuuse**

- Para ver los navegadores que soportan una determinada propiedad CSS o un elemento HTML5.

### **Autoprefixer**

- Para ahorrar tiempo y facilitar la inclusión de prefijos de las propiedades CSS que todavía no son estables.

### **Repositorio de fuentes**

- Google Fonts
- Font Squirrel
- Da Font

### **Repositorio de iconos de Font Awesome**

## 1.13. Herramientas útiles y recursos

### Generador de sombras online

- Textos: [css3gen.com/text-shadow](https://css3gen.com/text-shadow)
- Cajas o contenedores: [cssmatic.com/box-shadow](https://cssmatic.com/box-shadow)

### Generador de degradados

- [cssgradient.io](https://cssgradient.io)
- [css3gen.com/gradient-generator](https://css3gen.com/gradient-generator)

### Patrones de degradados

- Diseños muy trabajados sin necesidad de cargar imágenes con mucho peso.
- [leaverou.github.io/css3patterns](https://leaverou.github.io/css3patterns)



03:

Preprocesadores CSS

### 03.01. Preprocesadores CSS.

- Herramientas que permiten traducir un código de hojas de estilo no estándar, específico del preprocesador en cuestión, a un código CSS estándar, entendible por los navegadores.
- Generan automáticamente hojas de estilo, añadiendo características de los lenguajes de programación: variables, funciones, selectores anidados, herencia, etc.





## 03.01. Preprocesadores CSS.

### Ventajas

- Ofrecen **utilidades** que no se encuentran en el lenguaje CSS, o bien no son compatibles con todos los navegadores.
- Se consigue un **ahorro de tiempo**.
  - Se escribe menos código para hacer las cosas.
- Mayor facilidad de **mantenimiento** del código:
  - Mejor organización.
  - Se edita un valor y afecta a diversos lugares del código CSS generado.

## 03.01. Preprocesadores CSS.

### Desventajas

- Aprender un nuevo lenguaje y escribir código no estándar.
- El código no estándar debe **compilarse** en CSS:
  - El código del preprocesador no lo entiende el navegador.
  - Existen alternativas para optimización esa traducción.

## 03.02. Preprocesador Sass.

- Cualquier preprocesador es válido.
- **Sass** se ha convertido en el preprocesador más usado y el más demandado:
  - Mayor apoyo de la comunidad.
  - Más ofertas de trabajo con Sass.
  - Más proyectos desarrollados que usan Sass.
- Muchos frameworks como Bootstrap usan Sass.

## 03.02. Preprocesador Sass.

### Ventajas

- Reduce el tiempo para crear y mantener el CSS.
- Permite una organización modular de los estilos:
  - Vital en proyectos grandes.
- Estructuras avanzadas: variables, listas, funciones y estructuras de control.
- Genera distintos tipos de salida, comprimida, normal o minimizada.
- Permite vigilar los ficheros (modo watch)
  - Si se modifica la hoja de estilos, se regenera automáticamente.
- Pocas dependencias (versión actual).

## 03.02. Preprocesador Sass.

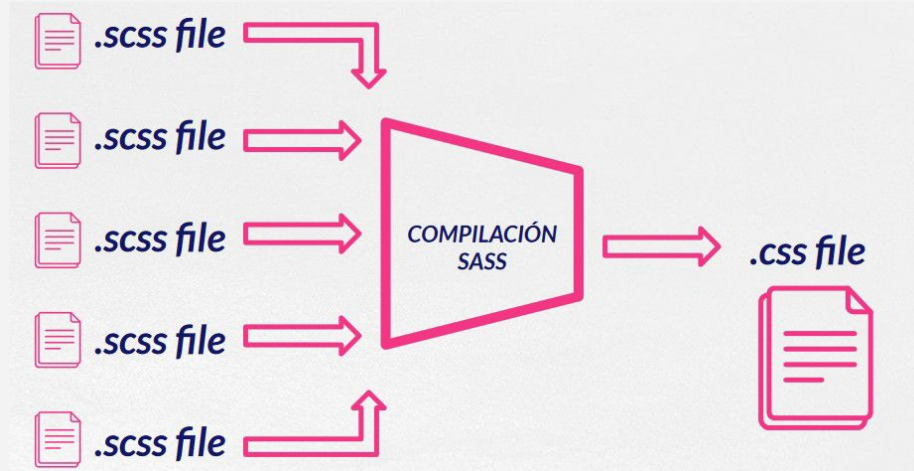
### Desventajas

- Hay que aprender a utilizar una nueva herramienta, lo que para algunos supone una desventaja.
- Hay un tiempo de consumo en el proceso de generación o compilación del CSS, sobre todo si es el archivo SCSS es muy grande. No es mucho tiempo, pero sí que existe esa pequeña demora.
- Tiene una sintaxis más compleja que CSS.

## 03.02. Preprocesador Sass.

### Funcionamiento

- Los navegadores no pueden compilar los ficheros sass.
- La compilación es un proceso aparte para obtener el fichero CSS.



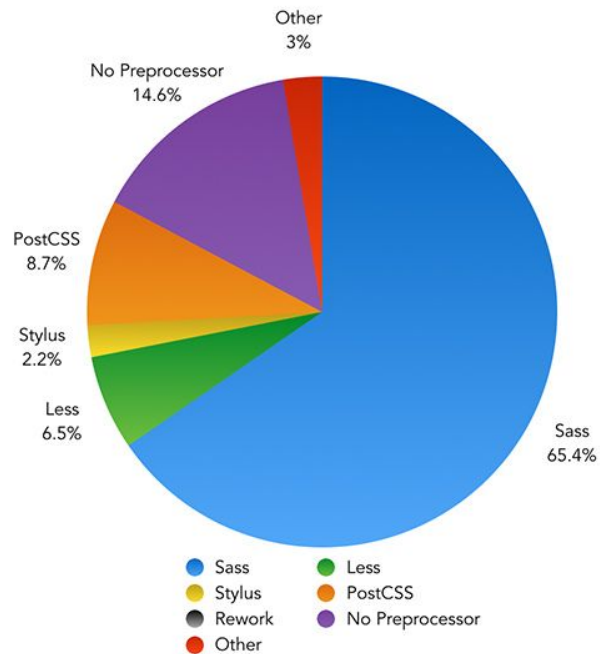
## 03.02. Preprocesador Sass.

### SASS vs SCSS

- **Sintaxis Sass (indentada)**
  - Archivos con extensión .sass
  - No usa llaves {}
    - Usa indentación.
  - No usa punto y coma ;
- **Sintaxis de Scss**
  - Archivos con extensión .scss
  - La más utilizada:
    - Muy parecida a la sintaxis de CSS.

SASS	SCSS	CSS
<pre>\$color: red \$color2: lime  a   color: \$color   &amp;:hover     color: \$color2</pre>	<pre>\$color: #f00; \$color2: #0f0;  a {   color: \$color;   &amp;:hover {     color: \$color2;   } }</pre>	<pre>a {   color: red; }  a:hover {   color: lime; }</pre>

## 03.02. Preprocesador Sass.





### 03.03. Instalación y configuración del preprocesador en VSC

- Necesario tener instalado **Node.js**
  - Se usará su gestor de paquetes (NPM - Node Package Manager) para instalar Sass.
    - [Descarga e instala la versión LTS \(Long Time Support\).](#)
- Abrir Visual Studio Code y verificar la instalación en el terminal.
  - node -v
  - npm -v
- **Instalar Sass** desde el terminal de VSC
  - npm install -g sass

### 03.03. Instalación y configuración del preprocesador en VSC

- Verificar la instalación en el terminal.
  - `sass --version`
  - Opcional: Activar la ejecución de scripts del sistema
    - Abrir Windows Powershell como administrador.
    - Ejecutar `Set-ExecutionPolicy Unrestricted`.
- Instalar la extensión **Live Sass Compiler** en VSC.
- Proyecto de ejemplo en VSC.

## 03.04. Sintaxis Scss

### Variables

- Colores, especificaciones de tamaño y otros.
- Símbolo dólar \$.

Tipo	Ejemplo de uso
Booleanas	<code>\$booleana: false;</code>
Numéricas	<code>\$line-height: 1.5; \$font-size: 1.2em;...</code>
Colores	<code>\$color: purple; \$borde: #333...</code>
Strings	<code>\$header: "Helvetica Neue"; \$str: Arial...</code>
Listas	<code>\$usuarios: Pepe, Juan, Alicia, Luisa; \$margin: 40px 5px 10px 20px;</code>
Mapas	<code>\$btn-colors: ("error" : \$color-error, "warning" : \$color-warning, "accepted" : \$color-accepted, "normal" : \$color-normal); - \$breakpoint: ( 'pequeño' : 576px, 'medio' : 768px, 'grande' : 992px );</code>
Null	<code>\$nulo: null;</code>

```
$bg-color: #df0174;  
$size: 1em;
```

```
$bg-color: #df0174;  
$size: 1em;  
body {  
  background-color: $bg-color;  
  margin: $size * 2;  
}
```

```
body {  
  background-color: #df0174;  
  margin: 2em;  
}
```

## 03.04. Sintaxis Scss

### Anidamiento de variables y de estilos

- Anida propiedades unas dentro de otras.
- Ahorro de código.

```
nav {
  ul {
    margin: 0;
    padding: 0;
    list-style: none;
  }
  a {
    display: block;
    padding: 6px 12px;
    text-decoration: none;
  }
}
```

```
nav ul {
  margin: 0;
  padding: 0;
  list-style: none;
}
nav a {
  display: block;
  padding: 6px 12px;
  text-decoration: none;
}
```

- Permite el uso de clases (.) y de propiedades del selector (&)

```
nav {
  .container {
    margin: 0;
    padding: 0;
    list-style: none;
    ul {
      display: block;
      padding: 6px 12px;
      text-decoration: none;
    }
  }
}
```

```
a {
  font-weight: bold;
  text-decoration: none;
  &:hover {
    text-decoration: underline;
  }
}
```

```
a {
  font-weight: bold;
  text-decoration: none;
}
a:hover {
  text-decoration: underline;
}
```

## 03.04. Sintaxis Scss

### Importación de estilos (@import)

- Incluir ficheros css unos dentro de otros.
- No genera nuevas peticiones HTTP.
  - Sass coge el contenido del fichero y lo combina.

```
@import 'base';  
  
body {  
  font: 100% Helvetica, sans-serif;  
  background-color: #efefef;  
}
```

## 03.04. Sintaxis Scss

### @Mixins

- Reglas fijas que se pueden invocar sin necesidad de escribir el código completo.
  - Similar a funciones: reutilización de código.
  - Parametrización.

```
@mixin transform($property) {  
  -webkit-transform: $property;  
  -ms-transform: $property;  
  transform: $property;  
}  
  
.box { @include transform(rotate(30deg)); }
```

- Mixin recibe un parámetro (property) que indica la cantidad de rotación.
- Al mixin se le llama con **@include** para indicar el nombre del mixin y pasándole el parámetro.

## 03.04. Sintaxis Scss

### @Extend

- Garantiza que todas las propiedades de una clase se transfieran a otra.

```
.button-scope {  
  margin: 5px;  
  border-radius: 2px;  
}  
.home-button {  
  @extend .button-scope;  
  background-color: $black;  
}  
.back-button {  
  @extend .home-button;  
}
```

```
.button-scope, .home-button, .back-button {  
  margin: 5px;  
  border-radius: 2px;  
}  
.home-button, .back-button {  
  background-color: #000000;  
}
```

## 03.04. Sintaxis Scss

### Funciones

- Sass permite utilizar funciones predefinidas:
  - Colores.
  - Listas.
  - Cadenas.
  - Selectores.
  - Números.
  - Mapas.
  - Introspección.
  - Miscellaneous.

```
***CODE***
$color-1: #ffff00;
$color-2: #0000ff;
body {
  background-color: mix($color-1, $color-2, 30%);
}
```

```
body {
  background-color: #4d4db3;
}
```



## 03.04. Sintaxis Scss

### Bucles y estructuras de control (@for, @while, @each, @if, @else if, @else)

- Por ejemplo, para añadir propiedades css a objetos con nombre parecido.

```
@for $i from 1 through 3 {  
  .item-#{$i} { width: 2em * $i; }  
}
```

```
.item-1 {  
  width: 2em; }  
.item-2 {  
  width: 4em; }  
.item-3 {  
  width: 6em; }
```

## 03.04. Sintaxis Scss

### Comentarios

- `/* */`
- `//`

### Error (@error), warn (@warn) y debug (@debug)

- Para la depuración del scss.

```
$test: false;
body {
  @if $test {
    @error "Mensaje de error";
    @debug "Test tiene el siguiente valor: #{ $test}";
  }
  else {
    @warn "Mensaje de warning";
  }
}
```

# 03. DISEÑO Y MAQUETACIÓN WEB CON HTML5 Y CSS3



**Las Fuentezuelas**  
INSTITUTO DE EDUCACIÓN SECUNDARIA

**DISEÑO DE  
INTERFACES WEB**  
2º DAW