

UD 5

CSS Avanzado

DISEÑO DE INTERFACES WEB

Técnico de Grado Superior Desarrollo de Aplicaciones Web

2024-25

Resultados de aprendizaje

2. Crea interfaces Web homogéneos 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.

Contenidos

- Técnicas imprescindibles.
 - Propiedades shorthand.
 - Limpiar floats.
 - Elementos a la misma altura.
 - Sombras.
 - Transparencias.
 - Sustitución de texto por imágenes.
 - Esquinas redondeadas.
 - Texto. Efectos gráficos: Texto con sombra.
Texto con relleno gradiente o degradado.
 - Inicializar los estilos.
Depuración.
Buenas prácticas.
- Uso avanzado de selectores, pseudo-clases, pseudo-elementos y selectores CSS 3.
- Propiedades avanzadas.
 - Propiedad white-space.
 - Propiedad display.
 - Propiedad outline.
 - Propiedad quotes.

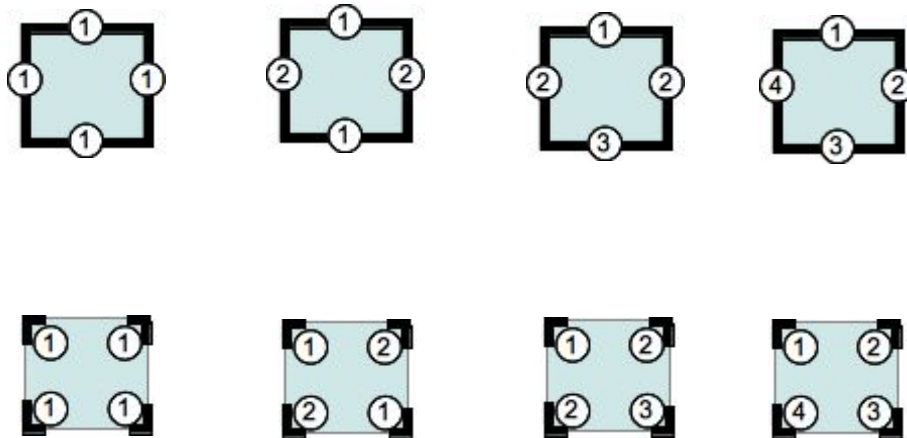
Técnicas imprescindibles

Propiedades *shorthand*

La especificación CSS define propiedades abreviadas para agrupar la definición de propiedades comunes que actúan sobre el mismo tema. Por ejemplo, la propiedad CSS `background` es una propiedad abreviada que puede definir los valores de `background-color`, `background-image`, `background-repeat` y `background-position`. Del mismo modo, las propiedades más comunes relacionadas con la tipografía se pueden definir usando la abreviatura `font`, y los diferentes márgenes alrededor de una caja se pueden definir usando la abreviatura `margin`.

```
margin-top: 10px;  
margin-right: 5px;  
margin-bottom: 10px;  
margin-left: 5px;
```

```
margin: 10px 5px 10px 5px;
```



Técnicas imprescindibles

Propiedades *shorthand*

```
background-color: #000;  
background-image: url(images/bg.gif);  
background-repeat: no-repeat;  
background-position: left top;
```

```
background: #000 url(images/bg.gif) no-repeat left top;
```

```
font-style: italic;  
font-weight: bold;  
font-size: 0.8em;  
line-height: 1.2;  
font-family: Arial, sans-serif;
```

```
font:  
    italic bold 0.8em/1.2 Arial,  
    sans-serif;
```

Casos especiales
¿Qué pasa cuando se omiten propiedades?

```
p {  
    background-color: red;  
    background: url(images/bg.gif) no-repeat left top;  
}
```

Técnicas imprescindibles

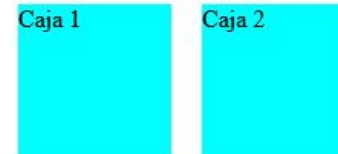
Limpiar *floats*

Float estaba destinado a mostrar, por ejemplo, imágenes con texto ajustado a su alrededor.
Ejemplo: [Drop Caps](#)

La propiedad `float` puede crear problemas de flujo en el diseño, dejando elementos "flotando" fuera del contenedor.

```
.container {  
  background-color: grey;  
  border: 1px solid black;  
}  
  
.box1,  
.box2 {  
  width: 100px;  
  height: 100px;  
  margin: 10px;  
  float: left;  
  background-color: cyan;  
}
```

```
<div class="container">  
  <div class="box1">Caja 1</div>  
  <div class="box2">Caja 2</div>  
</div>
```



Técnicas imprescindibles

Limpiar *floats*

Para resolverlo, se usa **clear**, que añade un bloque después de los elementos flotantes para "limpiar" el flujo.

```
.container {  
  background-color: grey;  
  border: 1px solid black;  
}  
  
.container::after {  
  content: "";  
  display: block;  
  clear: both;  
}  
  
.box1,  
.box2 {  
  width: 100px;  
  height: 100px;  
  margin: 10px;  
  float: left;  
  background-color: cyan;  
}
```

```
<div class="container">  
  <div class="box1">Caja 1</div>  
  <div class="box2">Caja 2</div>  
</div>
```



Técnicas imprescindibles

Elementos a la misma altura

Un par de técnicas:

Uso de flexbox con align-items: stretch

Uso de display: table y display: table-cell

Permite mantener varios elementos con la misma altura sin tener que especificarla manualmente.
Flexbox es más versátil.

```
.container {  
  display: table;  
  width: 100%;  
}  
  
.box {  
  display: table-cell;  
  width: 50%;  
  background-color: violet;  
  padding: 20px;  
}
```

```
<div class="container">  
  <div class="box">Caja 1 con contenido de  
  sobra para mostrar el efecto.</div>  
  <div class="box">Caja 2</div>  
</div>
```


Técnicas imprescindibles

Sombras

Las sombras `box-shadow` permiten manipular la profundidad visual.

`inset`:

- si no se especifica, la sombra asume una sombra difuminada (como si la caja fuera levantada por encima del contenido).
- si se especifica, cambia la sombra a una dentro del marco (como si el contenido estuviera deprimido dentro de la caja).

```
blockquote {  
  padding: 20px;  
  box-shadow:  
    inset 0 -3em 3em rgba(0, 0, 0, 0.1),  
    0 0 0 2px rgb(255, 255, 255),  
    0.3em 0.3em 1em rgba(0, 0, 0, 0.3);  
}
```

“Caminante, son tus huellas
el camino y nada más;
Caminante, no hay camino,
se hace camino al andar.”

— Antonio Machado

Técnicas imprescindibles

Sombras

Las sombras `text-shadow` permiten manipular la profundidad visual en textos.

```
.red-text-shadow {  
  text-shadow: red 0 -2px;  
}
```

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.

Técnicas imprescindibles

Transparencias

La transparencia **opacity** ajusta la opacidad de un elemento; esto es, en qué grado se superpone el fondo al elemento.

valor	Significado
0	El elemento es transparente (invisible).
Cualquier valor entre 0 y 1	El elemento es translúcido.
1	El elemento es opaco (sólido).

Técnicas imprescindibles

Sustitución de texto por imágenes

Útil en casos en los que se necesita mostrar logotipos (o lo que sea) que sea preferible como imagen en lugar de texto.

Esta técnica permite ocultar el texto original para que no interfiera visualmente

Mantiene la accesibilidad y el SEO al incluir el texto en el código HTML. Esto es crucial, ya que los motores de búsqueda y las tecnologías de asistencia aún pueden leer el texto original, incluso cuando se reemplaza visualmente por una imagen.

- `text-indent` negativo
- `display: none` en el texto y `background-image`
- `position: absolute` en el texto ubicándolo fuera y `background-image`

Técnicas imprescindibles

Sustitución de texto por imágenes

```
.logo1 {  
  background-image: url('logo1.png');  
  width: 128px;  
  height: 128px;  
  display: block;  
  text-indent: -9999px; /* Mueve el texto fuera */  
  overflow: hidden;  
}
```

```
.logo3 {  
  background-image: url('logo3.png');  
  width: 128px;  
  height: 128px;  
  position: relative;  
}  
.logo3-text {  
  position: absolute;  
  left: -9999px; /* Mueve el texto fuera */  
}
```

```
<div class="logo1">  
</div>  
  
<div class="logo2">  
  <span class="logo2-text">Texto que no se ve</span>  
</div>  
  
<div class="logo3">  
  <span class="logo3-text">Texto que tampoco se ve</span>  
</div>
```

```
.logo2 {  
  background-image: url('logo2.png');  
  width: 128px;  
  height: 128px;  
  display: block;  
}  
.logo2-text {  
  display: none;  
}
```

Técnicas imprescindibles

Esquinas redondeadas

border-radius permite dar a los elementos un aspecto suave y moderno con esquinas redondeadas.

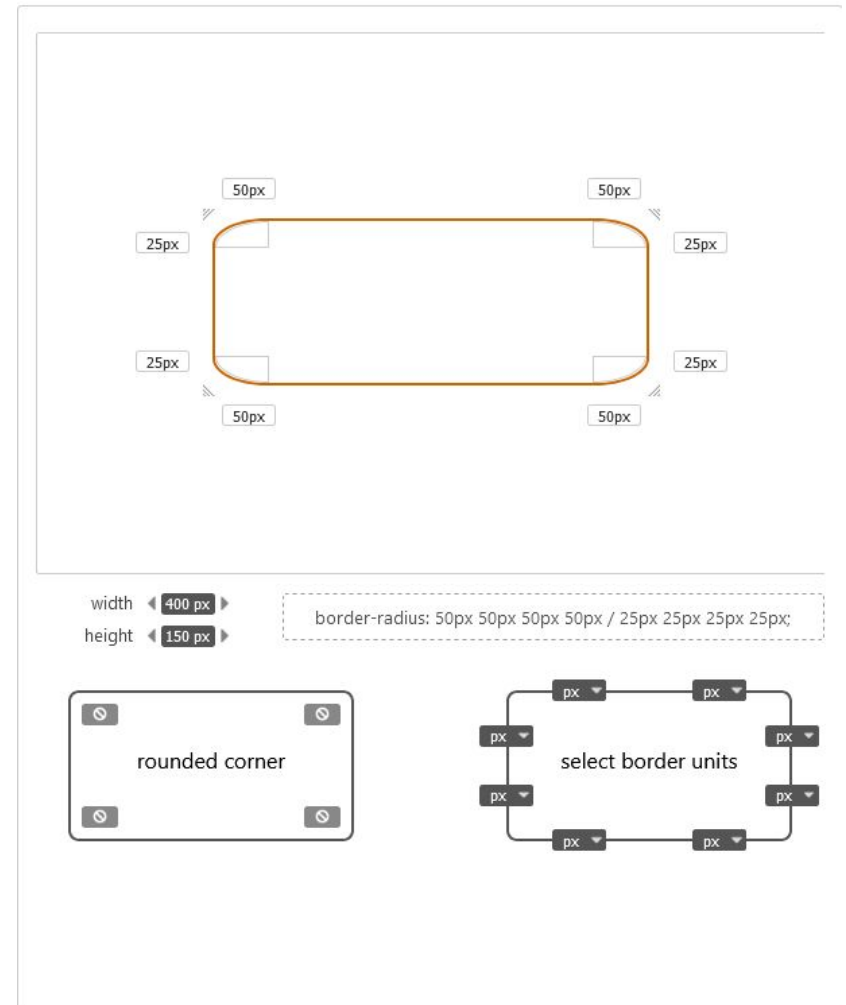
Puede usarse con valores específicos en cada esquina o como un valor global.

```
.rounded {  
  border-radius: 15px;  
}
```

Generador de border-radius

Esta herramienta puede ser usada para generar efectos de [border-radius](#) (inglés) de CSS3.

Herramienta



The screenshot shows the 'Herramienta' (Tool) interface for generating border-radius CSS code. It features a visual representation of a rounded rectangle with corner radius values (50px and 25px) and a CSS code output field.

width: 400 px
height: 150 px

border-radius: 50px 50px 50px 50px / 25px 25px 25px 25px;

rounded corner

select border units

Técnicas imprescindibles

Efectos en el texto: text-decoration

- text-decoration
- text-decoration-color
- text-decoration-line
- text-decoration-skip-ink
- text-decoration-style
- text-decoration-thickness
- text-emphasis
- text-emphasis-color
- text-emphasis-position
- text-emphasis-style
- text-shadow
- text-underline-offset
- text-underline-position

Técnicas imprescindibles

Efectos en el texto: text-decoration

```
.under {  
  text-decoration: underline red;  
}  
  
.over {  
  text-decoration: wavy overline lime;  
}  
  
.line {  
  text-decoration: line-through;  
}  
  
.plain {  
  text-decoration: none;  
}  
  
.underover {  
  text-decoration: dashed underline overline;  
}  
  
.thick {  
  text-decoration: solid underline purple 4px;  
}  
  
.blink {  
  text-decoration: blink;  
}
```

Subrayado por debajo

Suprarayado por arriba

~~Línea atravesando el texto~~

Este [link no está subrayado](#), aunque por defecto lo estaría.

Este texto tiene línea por encima y por debajo.

El subrayado de este texto es más gordo que lo habitual.

Este texto tiene el efecto de parpadeo, aunque puede estar inactivo en el navegador.

Técnicas imprescindibles

Efectos en el texto: text-shadow

CSS Demo: text-shadow

RESET

text-shadow: 1px 1px 2px pink;

text-shadow: #FC0 1px 0 10px;

text-shadow: 5px 5px #558ABB;

text-shadow: red 2px 5px;

text-shadow: 5px 10px;

text-shadow: 1px 1px 2px red, 0 0 1em blu

Far out in the uncharted
backwaters of the unfashionable
end of the western spiral arm of
the Galaxy...

Técnicas imprescindibles

Inicialización de estilos

¿Realmente es necesario?

El CSS reset elimina el estilo predeterminado de los navegadores. Proporciona un punto de partida “limpio” para el diseño.

```
* {  
  margin: 0;  
  padding: 0;  
  box-sizing: border-box;  
}
```



```
body {  
  margin: 0;  
}
```



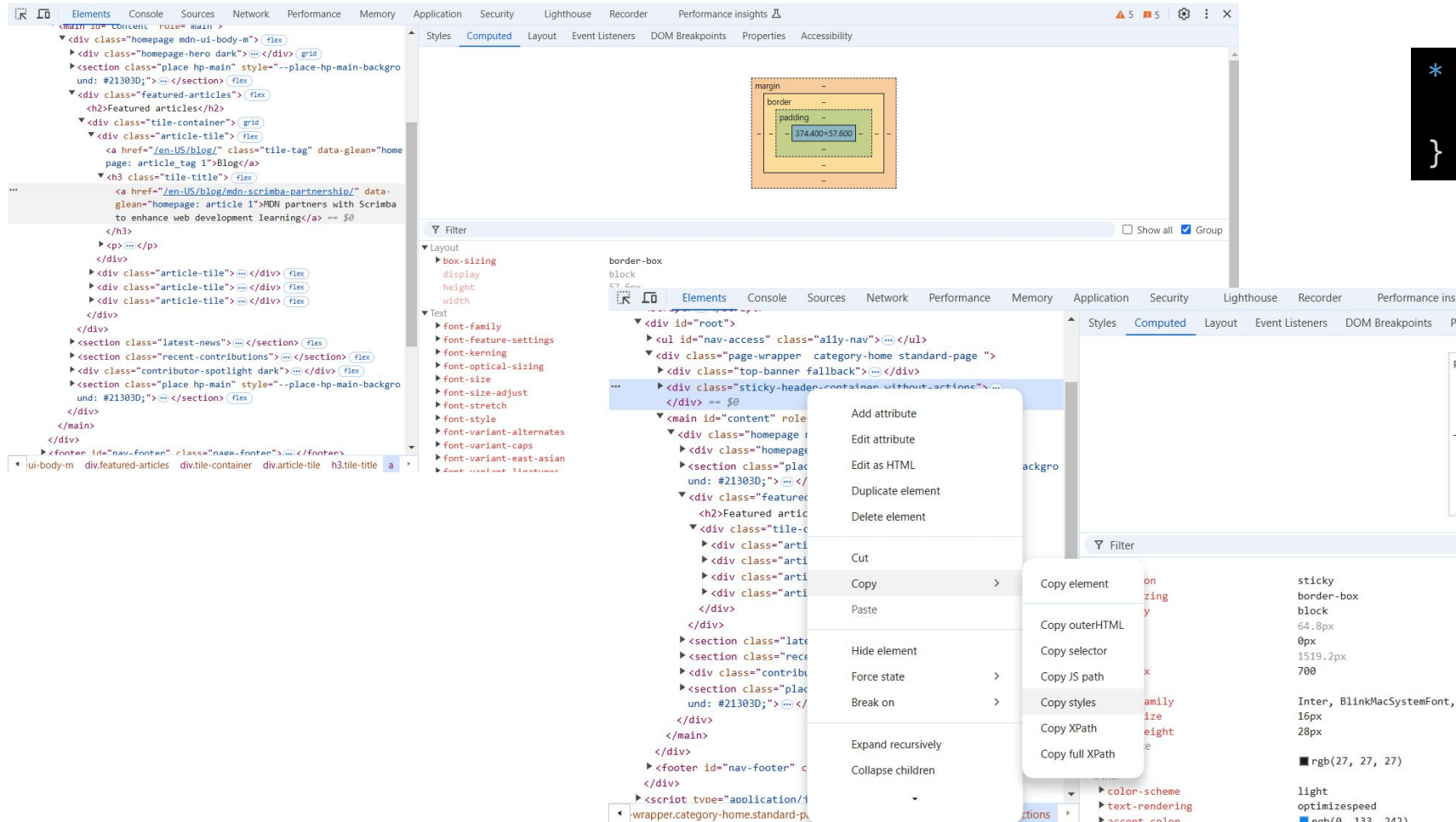
Normalize.css

A modern, HTML5-ready alternative to CSS resets

Técnicas imprescindibles

Depuración

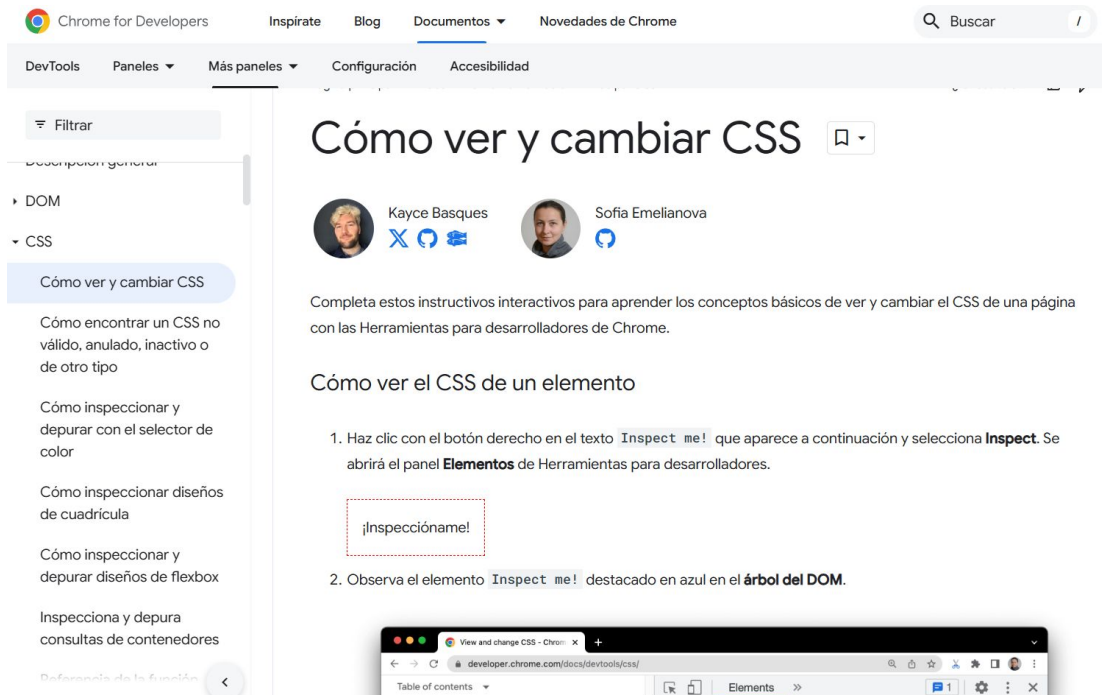
Puede ser útil mostrar provisionalmente un borde en todos los elementos



```
* {  
  outline: 1px solid red;  
}
```

Técnicas imprescindibles Depuración

<https://developer.chrome.com/docs/devtools/css?hl=es-419>



Identify potential CSS improvements

- 1 Capture an overview of your page's CSS
- 2 Identify potential CSS improvements (e.g. low contrast issues, unused declarations, color or font mismatches)
- 3 Locate the affected elements in the Elements panel

Capture overview

Preview feature

Our team is actively working on this feature and we would love to know what you think. [Send us your feedback.](#)



Video and documentation

[Quick start: get started with the new CSS overview panel](#)

<https://developer.chrome.com/docs/devtools/css-overview?hl=es-419>

Técnicas imprescindibles

Depuración

mdn web docs

ReferencesGuidesPlusCurriculumBlogTools

Theme

Aprende desarrollo web > CSS > Bloques de construcción CSS > Depurar el CSS

Esta página ha sido traducida del inglés por la comunidad. Aprende más y únete a la comunidad de MDN Web Docs.

Filter

Contenido desbordado

Valores y unidades CSS

Dimensionar elementos en CSS

Imágenes, medios y elementos de formulario

Estilizando tablas

Advanced styling effects

Depurar el CSS

Organizing your CSS

Comprensión de los fundamentos de CSS

Creating fancy letterheaded paper

A cool-looking box

Styling text

CSS layout

Depurar el CSS

Anterior

Overview: Bloques de construcción CSS

Siguiente

Al escribir CSS te puedes encontrar que, a veces, alguna parte de tu CSS no hace lo que esperas. Tal vez creas que cierto selector debería coincidir con un elemento, pero no sucede nada; o una caja tiene un tamaño diferente al que esperabas. Este artículo te orientará sobre cómo solucionar un problema de CSS y te mostrará cómo las DevTools incluidas en todos los navegadores modernos pueden ayudarte a descubrir qué sucede.

Prerrequisitos:	Conocimientos básicos de informática, tener el software básico instalado, conocimientos básicos de trabajar con archivos , HTML básico (véase Introducción a HTML) y nociones de cómo funciona el CSS (véase Primeros pasos con el CSS).
Objetivo:	Conocer los conceptos básicos de las DevTools que hay en los navegadores y de cómo inspeccionar y editar el CSS de un modo fácil.

https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/Debugging_CSS

Técnicas imprescindibles

Buenas prácticas...

Evitar conflictos de estilos

Mejorar el rendimiento y mantenimiento

... convenciones que ayudan a mantener el código CSS legible, modular y eficiente.

Esto incluye:

nomenclatura, estructura de selectores, uso de propiedades y organización de archivos.

- Usa **nombres de clases significativos**
- Evita la redundancia
 - Aprovechar las propiedades *shorthand*
- Organiza el código en secciones (colores, tipografías, layout...)
- Usa lo menos que puedas selectores anidados o muy específicos
- Usa unidades relativas (em, %) en lugar de valores absolutos (px)

Echa un vistazo a metodologías de CSS, como BEM, OOCSS

Uso avanzado de selectores

Selectores de tipo, de clase y de ID

También el
selector universal *

```
h1 {  
}
```

```
.box {  
}
```

```
#unique {  
}
```

Selectores de atributo

```
a[title] {  
}
```

```
a[href="https://example.com"]  
{  
}
```


Uso avanzado de selectores

Selectores de atributo de presencia y valor

Selector	Ejemplo	Descripción
<code>[attr]</code>	<code>a[title]</code>	Relaciona elementos con un mismo nombre de atributo, <i>attr</i> (el valor que se indica entre corchetes).
<code>[attr=value]</code>	<code>a[href="https://example.com"]</code>	Relaciona elementos con un mismo nombre de atributo, <i>attr</i> , cuyo valor es exactamente el mismo, <i>value</i> (la cadena de caracteres que se indica entre corchetes).
<code>[attr~=value]</code>	<code>p[class~="special"]</code>	<p>Relaciona los elementos con un mismo nombre de atributo, <i>attr</i>, cuyo valor es exactamente <i>value</i>, o los elementos con un mismo atributo <i>attr</i> que contiene uno o más valores de los cuales, al menos uno, coincide con <i>value</i>.</p> <p>Ten en cuenta que en una lista que incluya más de un valor, los distintos valores se separan con un espacio.</p>
<code>[attr =value]</code>	<code>div[lang="zh"]</code>	Relaciona los elementos con un mismo nombre de atributo, <i>attr</i> , cuyo valor puede ser exactamente <i>value</i> o puede comenzar con <i>value</i> seguido inmediatamente por un guion.

Uso avanzado de selectores

Selectores de atributo de presencia y valor

```
li[class] {  
  font-size: 200%;  
}  
  
li[class="a"] {  
  background-color: yellow;  
}  
  
li[class~="a"] {  
  color: red;  
}
```

```
<h1>Attribute presence and value selectors</h1>  
<ul>  
  <li>Item 1</li>  
  <li class="a">Item 2</li>  
  <li class="a b">Item 3</li>  
  <li class="ab">Item 4</li>  
</ul>
```

Attribute presence and value selectors

- Item 1
- **Item 2**
- **Item 3**
- Item 4

Uso avanzado de selectores

Selectores de atributo coincidentes con subcadenas

Selector	Ejemplo	Descripción
<code>[attr^=value]</code>	<code>li[class^="box-"]</code>	Relaciona elementos con un mismo nombre de atributo, <i>attr</i> , cuyo valor empieza exactamente con la subcadena de caracteres <i>value</i> .
<code>[attr\$=value]</code>	<code>li[class\$="-box"]</code>	Relaciona elementos con un mismo nombre de atributo, <i>attr</i> , cuyo valor termina exactamente con la subcadena de caracteres <i>value</i> .
<code>[attr*=]</code>	<code>li[class*="box"]</code>	Relaciona elementos con un mismo nombre de atributo, <i>attr</i> , cuyo valor incluye al menos una ocurrencia de la subcadena <i>value</i> en algún punto de la cadena.

Uso avanzado de selectores

¿Cómo ignorar
mayúsculas y
minúsculas?

Selectores de atributo
coincidentes con subcadenas

```
li[class^="a"] {  
  font-size: 200%;  
}  
  
li[class$="a"] {  
  background-color: yellow;  
}  
  
li[class*="a"] {  
  color: red;  
}
```

```
<h1>Attribute substring matching selectors</h1>  
<ul>  
  <li class="a">Item 1</li>  
  <li class="ab">Item 2</li>  
  <li class="bca">Item 3</li>  
  <li class="bcabc">Item 4</li>  
</ul>
```

Attribute substring matching selectors

- Item 1
- Item 2
- Item 3
- Item 4

Uso avanzado de selectores pseudo-clases

selector que marca los elementos que están en un estado específico, por ejemplo, los que son el primer elemento de su tipo, o aquellos por los que el cursor pasa por encima

comienzan con **dos puntos :**

```
article p:first-child {  
  font-size: 120%;  
  font-weight: bold;  
}
```

Hay muchísimas
(ver la Referencia)
Por ejemplo...

- `:hover` : solo interviene si el usuario pasa el cursor sobre un elemento, normalmente un enlace.
- `:focus` : solo interviene si el usuario selecciona el elemento con los controles del teclado.

Uso avanzado de selectores pseudo-elementos

actúan como si hubieras añadido un elemento totalmente nuevo en el HTML, en lugar de haber aplicado una clase nueva a los elementos presentes

comienzan con **dos puntos dobles ::**

```
article p::first-line {  
  font-size: 120%;  
  font-weight: bold;  
}
```

Ejemplo de uso:
CSS Arrow

::before y ::after

estos pseudo-elementos son especiales:
generan contenido (indicado en la propiedad **content**)

```
.box::before {  
  content: "Esto se mostrará antes del siguiente contenido. ";  
}
```

```
<p class="box">Contenido en mi página.</p>
```

Uso avanzado de selectores pseudo-clases y pseudo-elementos

Se pueden combinar
por ejemplo

```
article p:first-child::first-line {  
    font-size: 120%;  
    font-weight: bold;  
}
```

Selecciona la primera línea del primer elemento <p>
que esté dentro de un elemento <article>

Uso avanzado de selectores

selectores de combinación

Descendientes

```
body article p
```

Hijos

```
article > p
```

Adyacentes

```
p + img
```

Hermanos

```
p ~ img
```

Uso avanzado de selectores métodos de document

Document.querySelector()

Devuelve el primer elemento del documento (utilizando un [recorrido primero en profundidad pre ordenado](#) de los nodos del documento) que coincida con el grupo especificado de selectores.

Sintaxis

JS

```
element = document.querySelector(selectores);
```

Donde:

- `element` es un objeto de tipo [element](#).
- `selectores` es una cadena de caracteres que contiene uno o más [selectores CSS](#) ^(inglés) separados por coma.

Document.querySelectorAll()

El método `querySelectorAll()` de un [Element](#) devuelve una [NodeList](#) estática (no viva) que representa una lista de elementos del documento que coinciden con el grupo de selectores indicados.

Nota: Este método se implementa en base al método `querySelectorAll()` del mixin `ParentNode`.

Sintaxis

JS

```
elementList = parentNode.querySelectorAll(selectors);
```

Parámetros

`selectors`

Un [DOMString](#) que contiene uno o más selectores para buscar coincidencias. Esta cadena de texto debe ser una cadena [CSS selector](#) válida; si no lo es, se lanzará una excepción `SyntaxError`. Vea [Locating DOM elements using selectors](#) para más información acerca de utilizar selectores para identificar elementos. Se pueden especificar varios selectores separándolos utilizando comas.

Uso avanzado de selectores

Tabla de referencias

Selector	Ejemplo	Tutorial de aprendizaje de CSS
Selector de tipo	h1 { }	Los tipos de selectores
Selector universal	* { }	El selector universal
Selector de clase	.box { }	Los selectores de clase
Selector de ID	#unique { }	Los selectores de ID
Selector de atributo	a[title] { }	Los selectores de atributo
Pseudoclase	p:first-child { }	Las pseudoclases
Pseudoelemento	p::first-line { }	Los pseudoelementos
Operadores de combinación descendentes	article p	Operadores de combinación descendentes
Operador de combinación de elementos hijo	article > p	Operadores de combinación de elementos hijo
Operador de combinación de elementos hermanos adyacentes	h1 + p	Hermanos adyacentes
Operador de combinación general de elementos hermanos	h1 ~ p	Hermanos generales

Uso avanzado de selectores ... para practicar



CSS Diner

Share

Level 1 of 32 ✓

Select the plates

Help, I'm stuck!

CSS Editor

style.css

HTML Viewer

table.html

```
1 Type in a CSS selector
2 {
3 /* Styles would go here. */
4 }
5
6 /*
7 Type a number to skip to a level.
8 Ex → "5" for level 5
9 */
10
11
12
13
14
15
16
17
18
19
20
```

```
1 <div class="table">
2   <plate />
3   <plate />
4 </div>
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

Type Selector

Select elements by their type

A

Selects all elements of type `A`. Type refers to the type of tag, so `<div>`, `<p>` and `` are all different element types.

Examples

`div` selects all `<div>` elements.

`p` selects all `<p>` elements.

Uso avanzado de selectores ... para practicar

mdn web docs

References Guides Plus Curriculum NEW Blog Tools

Theme Log in

Sign up for free

CSS selectors > Test your skills: Selectors

English (US)

Filter

In this article

Task 1

Task 2

Task 3

Task 4

Task 5

Complete beginners start here!

▶ Getting started with the web

HTML — Structuring the web

▶ Introduction to HTML

▶ Multimedia and embedding

▶ HTML tables

Test your skills: Selectors

The aim of this skill test is to assess whether you understand [CSS selectors](#).

Note: You can try solutions in the interactive editors on this page or in an online editor such as [CodePen](#), [JSFiddle](#), or [Glitch](#).

If you get stuck, you can reach out to us in one of our [communication channels](#).

Task 1

In this task, use CSS to do the following things, without changing the HTML:

- Make `<h1>` headings blue.
- Give `<h2>` headings a blue background and white text.
- Cause text wrapped in a `` to have a font-size of 200%.

Your final result should look like the image below:

Propiedades avanzadas display

El valor **none** permite Desactivar la visualización del elemento; el documento se procesa como si el elemento no existiera en el DOM.

La propiedad CSS display especifica:

- si un elemento es tratado como **block** o **inline element**
 - el diseño usado por sus hijos, como **flow**, **grid** o **flex**
-
- **block**: El elemento ocupa toda la línea horizontal y crea un salto de línea (por defecto en <div>, <h1>, etc.)
Propiedades: Margen, padding, y tamaño (ancho, alto) afectan a todo el elemento.
 - **inline**: El elemento se coloca en línea con otros, solo ocupa el ancho de su contenido y no fuerza salto de línea (por defecto en , <a>, etc.)
Propiedades: Solo afecta padding, margen, y borde horizontal; el ancho y alto no cambian.
 - **inline-block**: Combinación de inline y block.
Permite elementos en línea con otros y puede tener propiedades de tamaño (ancho, alto).

Modos avanzados (display: flex y display: grid) para diseños de disposición complejos.

Propiedades avanzadas display

Reto 1:

Crea una lista de elementos `` dentro de un contenedor ``. Aplica `display: inline` a los `` para que aparezcan en línea, como si fueran elementos de un menú horizontal.

Reto 2:

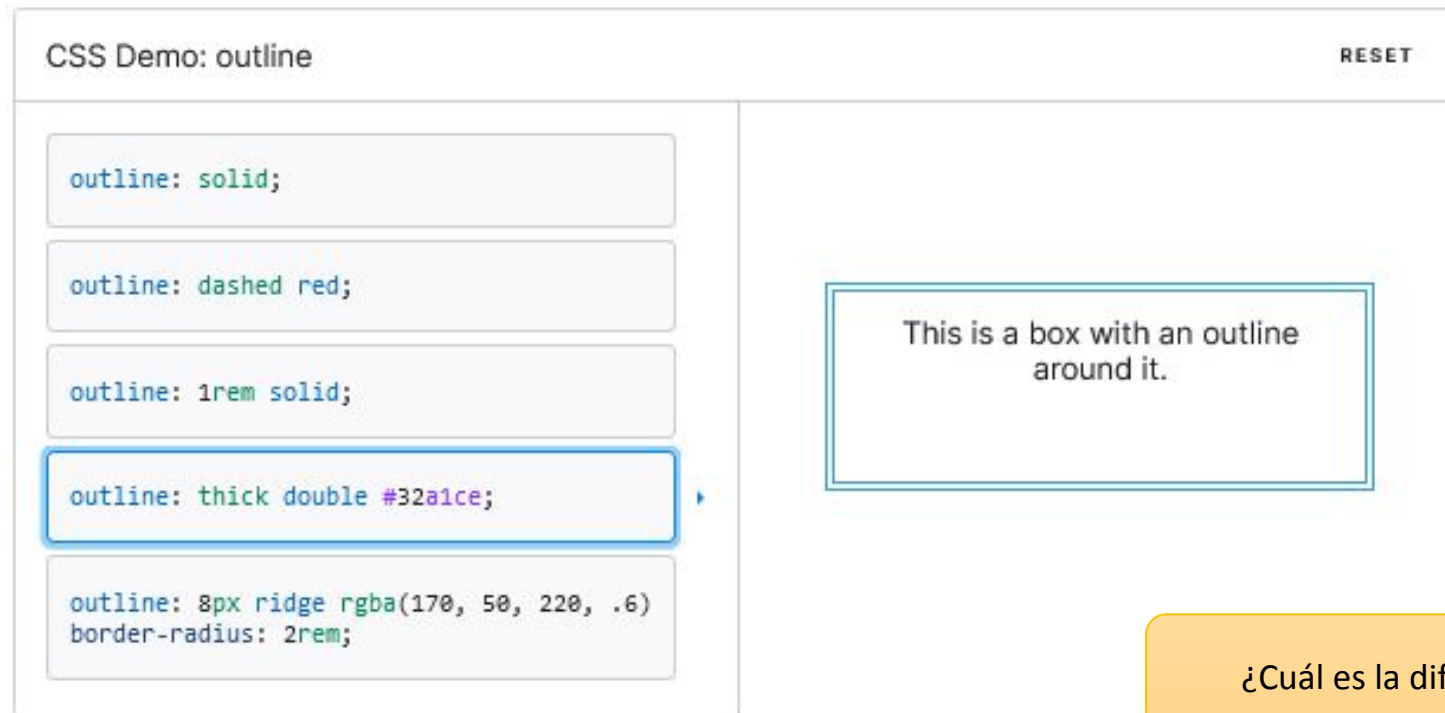
Diseña un contenedor `<div>` con varios párrafos `<p>`. Cambia el `display` de los párrafos a `inline-block`, y ajusta sus anchos para ver cómo se comportan con espacio alrededor.

Reto 3:

Usa `display: none` para ocultar un mensaje dentro de un `<div>`. Después, usa JavaScript o una clase CSS para cambiar su `display` a `block` al hacer clic en un botón, haciendo que el mensaje aparezca en la pantalla.

Propiedades avanzadas outline

Shorthand para establecer una o más de las propiedades individuales de contorno `outline-style`, `outline-width` y `outline-color`



¿Cuál es la diferencia entre border y outline?

Propiedades avanzadas outline

Reto 1:

Crea un botón y aplícale un **outline** rojo.

Reto 2:

Usa el pseudo-clase **:focus** para agregar un **outline** de 1px azul al botón cuando esté en foco.

Reto 3:

Usa outline en un formulario y personalízalo en función del estado (**:focus**, **:hover**).

Propiedades avanzadas quotes

indica cómo debe renderizar las citas el navegador

```
/* Valor con palabras clave */
```

```
quotes: none;
```

```
/* Valor <string> */
```

```
quotes: "«" "»"; /* Especifica open-quote y close-quote a las marcas de cita francesas */
```

```
quotes: "«" "»" "⟨" "⟩"; /* Especifica dos niveles de marcas de cita */
```


Propiedades avanzadas quotes

```
blockquote {  
    quotes: "“" "”" "„" "‟";  
}  
q::before {  
    content: open-quote;  
}  
q::after {  
    content: close-quote;  
}
```

Reto:

Agrega comillas automáticas en citas usando quotes.

¿Diferencia/uso de
blockquote y q?

¿y cite?

Propiedades avanzadas counter-reset y counter-increment

crea e inicializa los contadores a determinados valores.

CSS Demo: counter-reset

RESET

counter-reset: none;

counter-reset: chapter-count 0;

counter-reset: chapter-count;

counter-reset: chapter-count 5;

counter-reset: chapter-count -5;

Alice's Adventures in Wonderland
Chapter 1: Down the Rabbit-Hole
Chapter 1.6: The Pool of Tears
Chapter 1.7: A Caucus-Race and a Long Tale
Chapter 1.8: The Rabbit Sends in a Little Bill

```
ol {  
    counter-reset: section;  
}  
li::before {  
    counter-increment: section;  
    content: "Section " counter(section) ". ";  
}
```

Propiedades avanzadas content

en pseudo-elementos (::before o ::after) muestra el valor actual del contador

```
.item::before {  
  content: counter(contador-seccion) ". ";  
}
```

Además, existe una regla de recientísima inclusión en CSS, **@counter-style** que permite definir estilos de contador que no forman parte del conjunto predefinido de estilos.

<https://developer.mozilla.org/es/docs/Web/CSS/@counter-style>

Algunos estilos listos para usar:

<https://r12a.github.io/app-counters/>

(A) One
(B) Two
(C) Three
(D) Four
(E) Five
...

CSS @counter-style Demo

One
Two
Three
Four
Five
Six
Seven
Eight
Nine
Ten
11 Eleven
12 Twelve
13 Thirteen
14 Fourteen

Select a counter style from the list

fixed

```
@counter-style circled-digits {  
  system: fixed;  
  symbols:  One Two Three Four Five Six Seven Eight Nine Ten  
            11 Eleven 12 Twelve 13 Thirteen 14 Fourteen;  
  suffix: " ";  
}  
  
ul {  
  list-style: circled-digits;  
}
```

Propiedades avanzadas ejemplo contadores

```
<div class="seccion">
  <h2 class="titulo-seccion">Introducción</h2>
  <p>Contenido de la introducción...</p>
  <h2 class="titulo-seccion">Desarrollo</h2>
  <p>Contenido del desarrollo...</p>
  <h2 class="titulo-seccion">Conclusión</h2>
  <p>Contenido de la conclusión...</p>
</div>
```

```
.seccion {
  counter-reset: contador-seccion;
}

.titulo-seccion::before {
  counter-increment: contador-seccion;
  content: "Sección " counter(contador-seccion) ": ";
}
```

Sección 1: Introducción

Contenido de la introducción...

Sección 2: Desarrollo

Contenido del desarrollo...

Sección 3: Conclusión

Contenido de la conclusión...

Propiedades avanzadas ejemplo contadores

Reto: Anidar contadores

Capítulo 1: Capítulo

1.1 Sección A

Contenido de la sección A...

1.2 Sección B

Contenido de la sección B...

Capítulo 2: Capítulo

2.1 Sección A

Contenido de la sección A...

2.2 Sección B

Contenido de la sección B...

```
<div class="contenido">
  <div class="capitulo">
    <h2>Capítulo</h2>
    <div class="seccion">
      <h3>Sección A</h3>
      <p>Contenido de la sección A...</p>
    </div>
    <div class="seccion">
      <h3>Sección B</h3>
      <p>Contenido de la sección B...</p>
    </div>
  </div>
  <div class="capitulo">
    <h2>Capítulo</h2>
    <div class="seccion">
      <h3>Sección A</h3>
      <p>Contenido de la sección A...</p>
    </div>
    <div class="seccion">
      <h3>Sección B</h3>
      <p>Contenido de la sección B...</p>
    </div>
  </div>
</div>
```

```
.contenido {
  counter-reset: contador-capitulo;
}

.capitulo {
  counter-reset: contador-seccion;
}

.capitulo h2::before {
  counter-increment: contador-capitulo;
  content: "Capítulo " counter(contador-capitulo) ": ";
  font-size: 1.3em;
  font-weight: bold;
}

.capitulo .seccion h3::before {
  counter-increment: contador-seccion;
  content: counter(contador-capitulo) "." counter(contador-seccion) " ";
  font-weight: bold;
}
```

Propiedades avanzadas white-space

propiedad que controla cómo se manejan los espacios en blanco en el texto y cómo este se ajusta dentro de los contenedores.

Es útil cuando queremos que el texto respete saltos de línea.

```
/* Valores con palabras clave */  
white-space: normal;  
white-space: nowrap;  
white-space: pre;  
white-space: pre-wrap;  
white-space: pre-line;
```

- **normal** (valor por defecto): Forma estándar, se ignoran los saltos de línea.
- **nowrap**: Evita saltos de línea, fuerza que el texto aparezca en una sola línea.
- **pre**: Mantiene espacios y saltos de línea (similar a la etiqueta <pre>).
- **pre-wrap**: Mantiene los saltos de línea, y el texto se ajusta a las líneas dentro del contenedor.
- **pre-line**: Similar a pre-wrap pero reduce espacios adicionales.

Propiedades avanzadas white-space

CSS Demo: white-space

RESET

`white-space: normal;`

`white-space: nowrap;`

`white-space: pre;`

`white-space: pre-wrap;`

`white-space: pre-line;`

`white-space: break-spaces;`

But ere she from the
church-door stepped She
smiled and told us why: 'It
was a wicked woman's curse,'
Quoth she,
 'and what care I?' She
smiled, and smiled, and
passed it off Ere from the
door she stept—

Propiedades avanzadas white-space

Fuente

```
<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor  
incididunt ut labore et dolore magna aliqua.
```

```
Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea  
commodo consequat.
```

```
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat  
nulla pariatur.
```

```
Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit  
anim id est laborum.</p>
```

CSS + Resultado

```
p { white-space: pre-line } }
```

Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua.
Ut enim ad minim veniam,
quis nostrud exercitation ullamco laboris nisi ut
aliquip ex ea commodo
consequat. Duis aute irure dolor in reprehenderit in
voluptate velit esse
cillum dolore eu fugiat nulla pariatur. Excepteur sint
occaecat cupidatat
non proident, sunt in culpa qui officia deserunt mollit
anim id est laborum.

Propiedades avanzadas white-space

Reto 1:

Crea un contenedor de texto y usa `white-space: pre` para que respete los saltos de línea.

Reto 2:

Aplica `white-space: nowrap` a un título y observa cómo el texto se mantiene en una sola línea, aun si el contenedor es estrecho.

Técnicas avanzadas gradientes

El tipo de datos [CSS <gradient>](#) indica un tipo de [<image>](#) que consiste de una transición progresiva entre dos o más colores (Degradado).

Un gradiente de CSS no es un [<color>](#) pero tampoco es una imagen con [dimensiones intrínsecas](#); es decir, que no tiene tamaño natural o preferido, ni una relación preferida. Su tamaño concreto coincidirá con los elementos a los que se aplica.

Pruébalo

CSS Demo: <gradient> RESET


background: linear-gradient(#f69d3c, #3f8

background: radial-gradient(#f69d3c, #3f8

background: repeating-linear-gradient(#f6

background: repeating-radial-gradient(#f6

background: conic-gradient(#f69d3c, #3f87

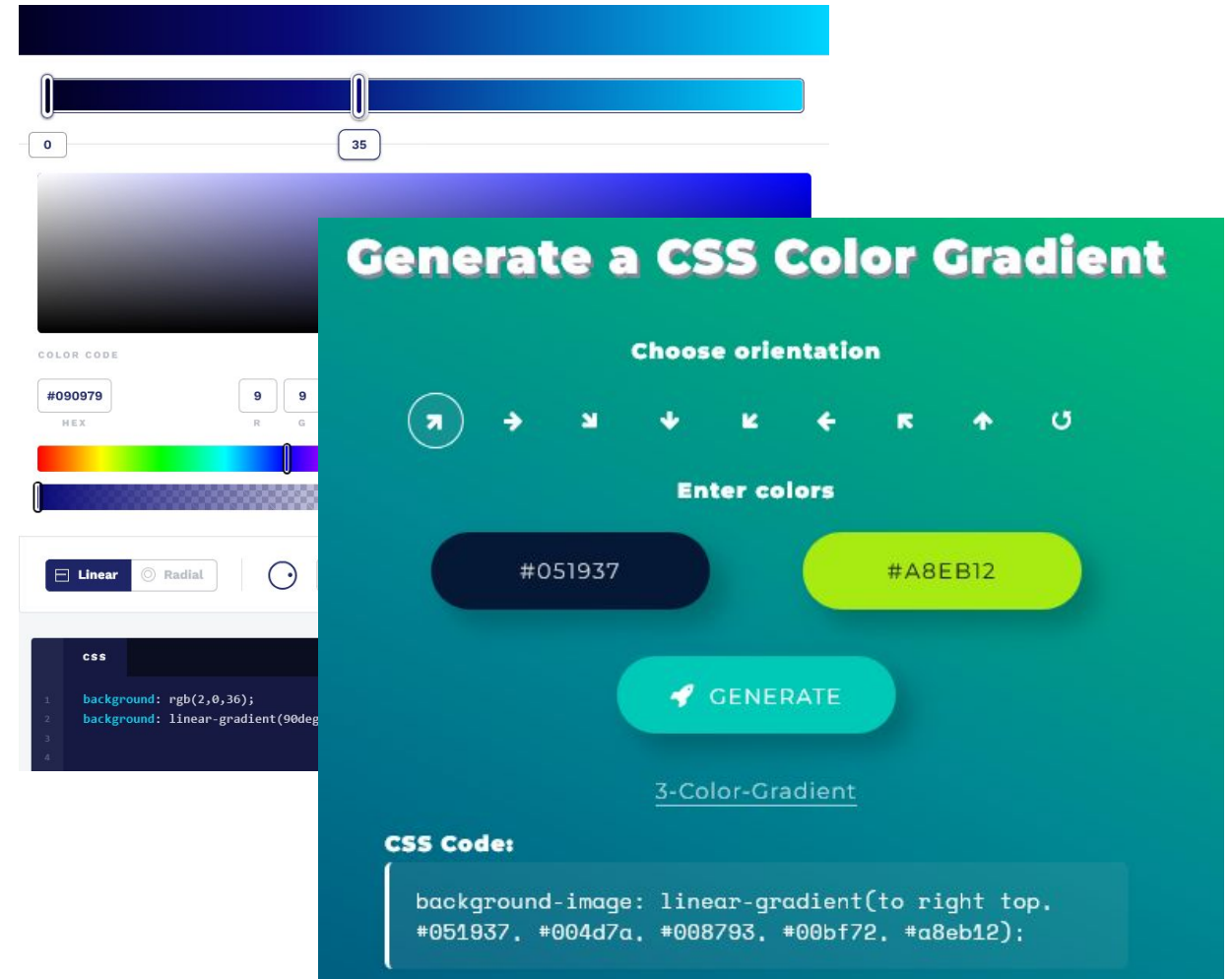


- Lineales:
transición a lo largo de una línea recta.
- Radiales:
transición desde un punto central hacia afuera.
- Cónicos:
transición en forma de círculo, de rueda de colores.

Técnicas avanzadas gradientes

Ojo con la accesibilidad y el rendimiento

- `background-image`:
Se utiliza para establecer el gradiente como imagen de fondo.
- `linear-gradient()`:
Define un gradiente lineal.
- `radial-gradient()`:
Define un gradiente radial.
- `conic-gradient()`:
Define un gradiente cónico.



<https://cssgradient.io/>
<https://mycolor.space/gradient>
<https://www.css-gradient.com/>

Técnicas avanzadas gradientes

Múltiples puntos de color, ángulos personalizados en el gradiente lineal, modifica la posición del centro en los gradientes radiales, patrones repetitivos con gradientes, transparencias con *transparent*, incluso funciones matemáticas para crear gradientes más sofisticados...

```
.gradiente {  
  width: 200px;  
  height: 100px;  
  background-image: linear-gradient(to right, blue, red);  
}
```



Probar:

```
background-image: radial-gradient(circle at center, blue, red);  
background-image: conic-gradient(from 180deg at 50% 50%, blue, green, yellow, red, transparent);  
...
```

Reto 1:

Crear un botón con un gradiente que cambie de color al pasar el ratón por encima.

Reto 2:

Crear un encabezado con un gradiente que simule un amanecer o atardecer.

Técnicas avanzadas

Imágenes embebidas

permite incluir imágenes directamente en nuestro código CSS, sin necesidad de referenciarlas desde un archivo externo



Representa los datos binarios de la imagen como una cadena de caracteres ASCII

La imagen se convierte en parte del código CSS

Se elimina la necesidad de realizar una petición HTTP adicional para cargar la imagen

¿Por qué utilizar imágenes embebidas?

Rendimiento: puede mejorar el rendimiento de una página, especialmente si las imágenes son pequeñas y se utilizan varias veces.

Al evitar múltiples solicitudes HTTP, se reduce el tiempo de carga de la página.

Mantenimiento: al tener la imagen integrada en el CSS no es necesario administrar archivos de imagen por separado. Bueno y malo.

Flexibilidad: permite aplicar estilos CSS directamente a la imagen, como filtros, opacidad, etc.

Técnicas avanzadas

Imágenes embebidas

```
.embebida {  
  width: 10px;  
  height: 10px;  
  background-image:  
url(data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAAUAAAACAYAAACNbyblAAAAHE1EQVQI12P4//8/w38GIAXDIBKE0D  
HxgljNBAAO9TXL0Y4OHwAAAABJRU5ErkJggg==);  
  background-size: cover;  
}
```



Técnicas avanzadas

Imágenes embebidas

También podemos utilizar otro formato de texto que ya conocemos...



El SVG está codificado como %3C (representando <) y %3E (representando >), para que sea compatible con la URL y el CSS

```
.embebidaxvg {  
  width: 150px;  
  height: 150px;  
  background-image: url("data:image/svg+xml,%3Csvg xmlns='http://www.w3.org/2000/svg' viewBox='0 0 100  
100'%3E%3Ccircle cx='50' cy='50' r='40' fill='%23FF6347' /%3E%3Ctext x='50%' y='55%' font-size='30'  
text-anchor='middle' fill='white'%3EA%3C/text%3E%3C/svg%3E");  
  background-size: contain;  
}
```

Técnicas avanzadas

Mapas de imágenes

permiten crear zonas interactivas dentro de una sola imagen en HTML, cada una enlazada a una URL o una acción específica

`` con `usemap`: referencia al mapa

`<map>` y `<area>`: definen las áreas

`shape`: define la forma (rect, circle, poly)

`coords`: especifica coordenadas

`href`: enlaza a una URL

```

<map name="basicMap">
  <area shape="rect" coords="10,10,50,50" href="https://www.paginadelrect.com">
  <area shape="circle" coords="100,100,30" href="https://www.paginadelcircle.com">
</map>
```


Técnicas avanzadas

Mapas de imágenes

HTML Demo: <map>

RESET

HTML

CSS

1 <map name="infographic">

2 <area

3 shape="poly"

4 coords="130,147,200,107,254,219,130,228"

5 href="https://developer.mozilla.org/docs/Web/

HTML"

6 target="_blank"

7 alt="HTML" />

8 <area

9 shape="poly"

10 coords="130,147,130,228,6,219,59,107"

11 href="https://developer.mozilla.org/docs/Web/

CSS"

12 target="_blank"

13 alt="CSS" />


14 <area

15 shape="poly"

16 coords="130,147,200,107,130,4,59,107"

17 href="https://developer.mozilla.org/docs/Web/

OUTPUT



Técnicas avanzadas

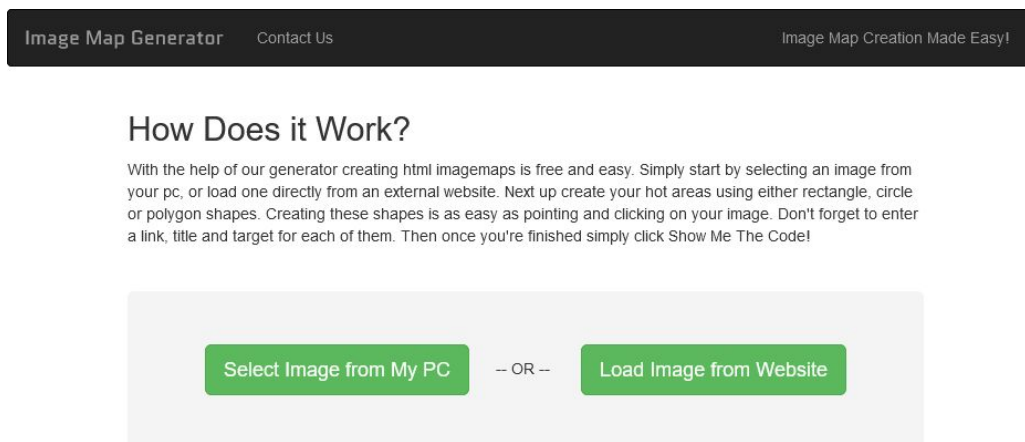
Mapas de imágenes

! las `<area>` no admiten CSS
... pero se pueden crear efectos con otros elementos superpuestos en CSS

CSS con efectos hover:

contenedores div invisibles encima de las áreas que se vuelvan visibles en `:hover`.
cambio de fondo para mostrar información o resaltar la sección.

JavaScript para detectar el mouseover y click en áreas y modificar otros elementos CSS



Existen herramientas para dibujar las áreas sobre la imagen y obtener las coordenadas de forma fácil
<https://www.image-map.net/>

...

Técnicas avanzadas

Variables en hojas de estilos

CSS permite utilizar variables para almacenar valores reutilizables
así puede ser más fácil gestionar cambios

Para declarar propiedades personalizadas (variables) usamos un nombre que comience con dos guiones -- y un valor que puede ser cualquier valor válido de CSS.

Como cualquier otra propiedad, la escribimos dentro de un set de reglas de estilo, que establece su ámbito.

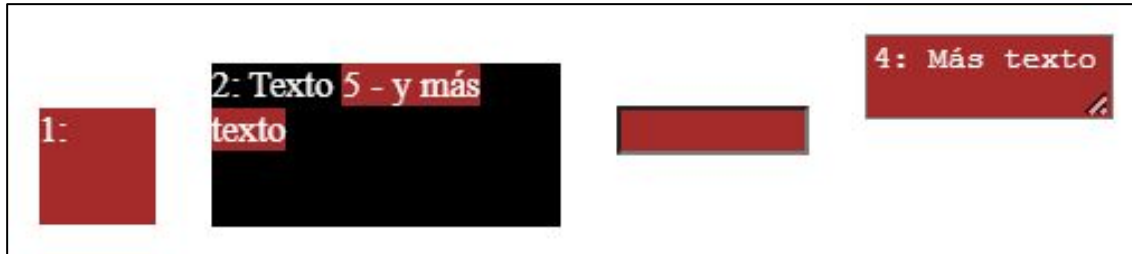
:root selecciona el elemento raíz del documento
puede ser útil para declarar variables CSS globales
<https://developer.mozilla.org/es/docs/Web/CSS/:root>

```
:root {  
  --color-primario: #3498db;  
}  
.titulo {  
  color: var(--color-primario);  
}
```

Técnicas avanzadas

Variables en hojas de estilos

```
<div>
  <div class="uno">1:</div>
  <div class="dos">2: Texto <span class="cinco">5 - y más texto</span></div>
  <input class="tres" />
  <textarea class="cuatro">4: Más texto</textarea>
</div>
```



```
:root {
  --main-bg-color: brown;
}

.uno {
  color: white;
  background-color: var(--main-bg-color);
  margin: 10px;
  width: 50px;
  height: 50px;
  display: inline-block;
}

.dos {
  color: white;
  background-color: black;
  margin: 10px;
  width: 150px;
  height: 70px;
  display: inline-block;
}

.tres {
  color: white;
  background-color: var(--main-bg-color);
  margin: 10px;
  width: 75px;
}

.cuatro {
  color: white;
  background-color: var(--main-bg-color);
  margin: 10px;
  width: 100px;
}

.cinco {
  background-color: var(--main-bg-color);
}
```

Técnicas avanzadas

Variables en hojas de estilos

Las propiedades personalizadas (variables) heredan.

Si no se establece ningún valor para una variable en un elemento, se utiliza el valor de su elemento padre.

```
<div class="uno">
  <div class="dos">
    <div class="tres"></div>
    <div class="cuatro"></div>
  </div>
</div>
```

```
.dos {
  --test: 10px;
}

.tres {
  --test: 2em;
}
```

- elemento class="dos": 10px
- elemento class="tres": 2em
- elemento class="cuatro": 10px (heredado de su padre)
- elemento class="uno": valor no válido (predeterminado)

Técnicas avanzadas

Variables en hojas de estilos

Reto 1:

Define una paleta de 4 colores utilizando variables CSS para una página web. Crea estilos para el fondo del body, los títulos (h1, h2, h3), el texto normal (p) y los botones (.boton), que cambian de color al pasar el ratón.

Usa al menos 4 variables:

fondo principal, color de texto, color de botones y color de hover para botones.

?? ? Se podrá acceder a las variables CSS desde JS ?? ?

Reto 2:

Crea una página con dos propiedades personalizadas, una para el color gris y otra para el color verde. Crea dos botones que permitan establecer el fondo de la página a color gris o verde; es decir, en cuyo comportamiento en JavaScript asignes el valor de las propiedades personalizadas.

Técnicas avanzadas

Estilos alternativos

Permite ofrecer opciones de estilo alternativas en una misma página

Por ejemplo, modos como oscuro/claro

```
<link rel="stylesheet" href="tema-claro.css" title="claro">  
<link rel="alternate stylesheet" href="tema-oscuro.css" title="oscuro">
```

- Permanentes ("persistent"): las hojas de estilos que se aplican siempre.
- Preferentes ("preferred"): las hojas de estilos alternativas que se aplican por defecto.
- Alternativas ("alternate"): las hojas de estilos alternativas que el usuario puede seleccionar.

Técnicas avanzadas

Estilos alternativos

Permanentes: rel="stylesheet" y title no aparece

```
<link rel="stylesheet" type="text/css" href="css/estilos.css" />  
<link rel="stylesheet" type="text/css" href="css/otros_estilos.css" />
```

Preferentes: rel="stylesheet" y title aparece

```
<link rel="stylesheet" title="Alternativa" type="text/css" href="css/estilos.css" />
```

Alternativas: rel="alternate stylesheet" y title puede aparecer

```
<link rel="alternate stylesheet" title="Alternativa" type="text/css"  
href="css/estilos.css" />
```


Técnicas avanzadas

Estilos alternativos

Reto:

Crea un archivo HTML con dos estilos alternativos definidos en hojas de estilo externas.

Incluye dos botones para activar cada uno de los estilos alternativos:
[tema moderno](#) y [tema clásico](#).

Usa el atributo `rel="alternate stylesheet"` para definir los estilos alternativos y JavaScript para activarlos.

Media queries

permiten aplicar estilos en función de las características de los dispositivos, como el ancho de pantalla, la orientación, la resolución, etc.

esto permite crear diseños *responsive*, adaptan el contenido y los estilos para diferentes tamaños de pantalla, desde móviles hasta monitores grandes

Se aplican en usando la palabra clave `@media` seguida de la condición (o condiciones) que deben cumplirse para aplicar los estilos específicos

```
body {  
    font-size: 18px;  
    padding: 20px;  
}
```

```
/* Estilo para dispositivos con ancho máximo de 600px */  
@media (max-width: 600px) {  
    body {  
        font-size: 16px;  
        padding: 10px;  
    }  
}
```

Media queries

En JavaScript puede usar el método `Window.matchMedia()` y `MediaQueryList.addListener()` para recibir notificaciones cada vez que cambie el estado de una consulta.

Opcionalmente, asocian las reglas a un tipo de medio:

```
@media print {  
  body {  
    font-size: 10pt;  
  }  
}  
  
@media screen {  
  body {  
    font-size: 13px;  
  }  
}  
  
@media screen, print {  
  body {  
    line-height: 1.2;  
  }  
}
```

https://developer.mozilla.org/es/docs/Web/CSS/@media#tipos_de_medios

Verifican características del navegador o del dispositivo:



A word cloud on a purple background listing various media query features. The words are in different colors and sizes, representing their frequency or importance. The features listed include: monochrome, display-mode, resolution, color-index, light-level, pointer, aspect-ratio, color, height, update-frequency, grid, any-hover, width, any-pointer, hover, orientation, scan, inverted-colors, overflow-inline, and overflow-block.

https://developer.mozilla.org/es/docs/Web/CSS/@media#a#caracter%C3%ADsticas_de_medios_media_feature

Media queries

Reto:

Crea una página HTML que muestre un diseño diferente en dispositivos con ancho menor a 768px (móviles) y en pantallas de escritorio.

- Define el estilo para una barra de navegación que esté visible en pantallas grandes y oculta en móviles.
- Cambia el tamaño del texto de los párrafos en función del ancho de la pantalla.



Efectos, transiciones y animaciones

- **Efectos:**
Aplican propiedades como box-shadow, opacity, y transform.
- **Transiciones:**
Permiten que los cambios entre estados se realicen de forma gradual.
- **Animaciones:**
Control total para crear movimientos complejos y secuenciales.
Usan la propiedad @keyframes para definir un conjunto de pasos.

CSS3 introdujo efectos, transiciones y animaciones para mejorar la interactividad y presentación de las páginas web. Anteriormente, se necesitaban tecnologías como Flash o JavaScript para añadir animaciones.

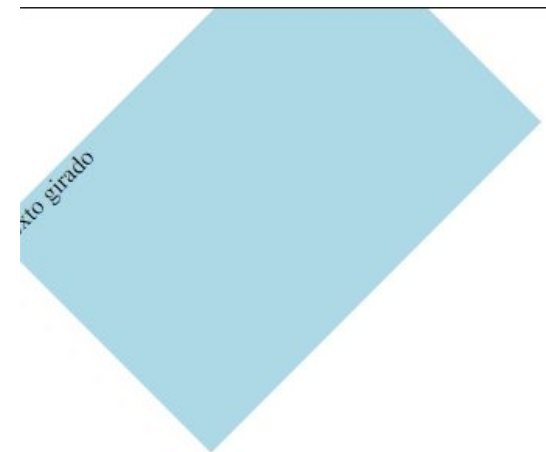
Efectos, transiciones y animaciones

Efectos

```
.card {  
  width: 300px;  
  height: 200px;  
  background: lightblue;  
  box-shadow: 5px 5px 15px rgba(0, 0, 0, 0.3);  
}
```



```
.card {  
  width: 300px;  
  height: 200px;  
  background: lightblue;  
  transform: rotate(-45deg);  
}
```



Efectos, transiciones y animaciones

Más efectos

filter

`grayscale(50%)`

`blur(5px)`

`blur(px)`
`brightness(%)`
`contrast(%)`
`drop-shadow()`
`grayscale(%)`
`hue-rotate(deg)`
`invert(%)`
`opacity(%)`
`saturate(%)`
`sepia(%)`
`url()`

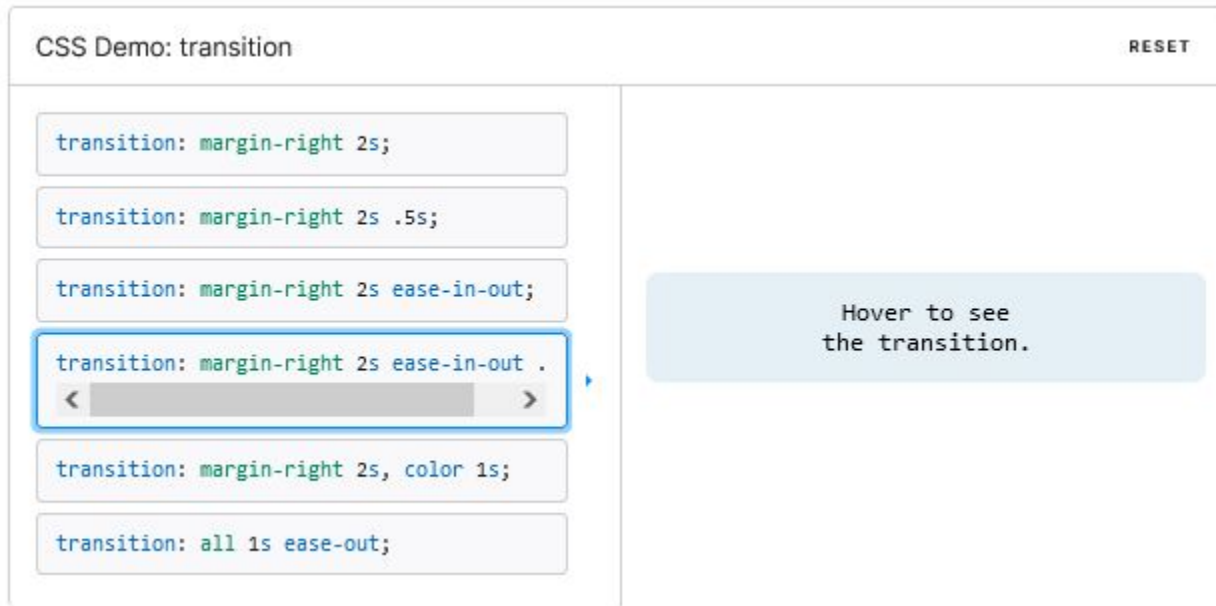
transform

`scale(1.5)`

`translate()`
`rotate()`
`scaleX()`
`scaleY()`
`scale()`
`skewX()`
`skewY()`
`skew()`
`matrix()`

Efectos, transiciones y animaciones

Transiciones



```
<a class="target">Hover over me</a>
```

```
.target {  
  font-size: 2rem;  
  background-color: palegoldenrod;  
  transition: background-color 2s 500ms;  
}  
  
.target:hover {  
  background-color: darkorange;  
}
```


Efectos, transiciones y animaciones

Efectos y transiciones

```
<div class="tarjeta">
  <h2>Tarjeta de Ejemplo</h2>
  <p>Efecto de sombra y escalado hover.</p>
</div>
```

```
.tarjeta {
  width: 300px;
  padding: 20px;
  background-color: #f0f0f0;
  border-radius: 8px;
  box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);
  transition: transform 0.3s, box-shadow 0.3s;
}

.tarjeta:hover {
  transform: scale(1.05);
  box-shadow: 0px 8px 16px rgba(0, 0, 0, 0.2);
}
```

Tarjeta de Ejemplo

Efecto de sombra y escalado hover.

Tarjeta de Ejemplo

Efecto de sombra y escalado hover.

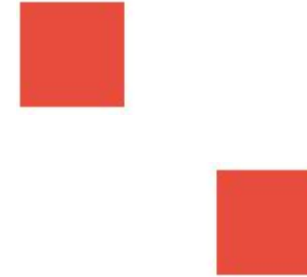
Efectos, transiciones y animaciones

Animaciones

permiten definir movimientos y transformaciones continuas en los elementos sin requerir la interacción del usuario

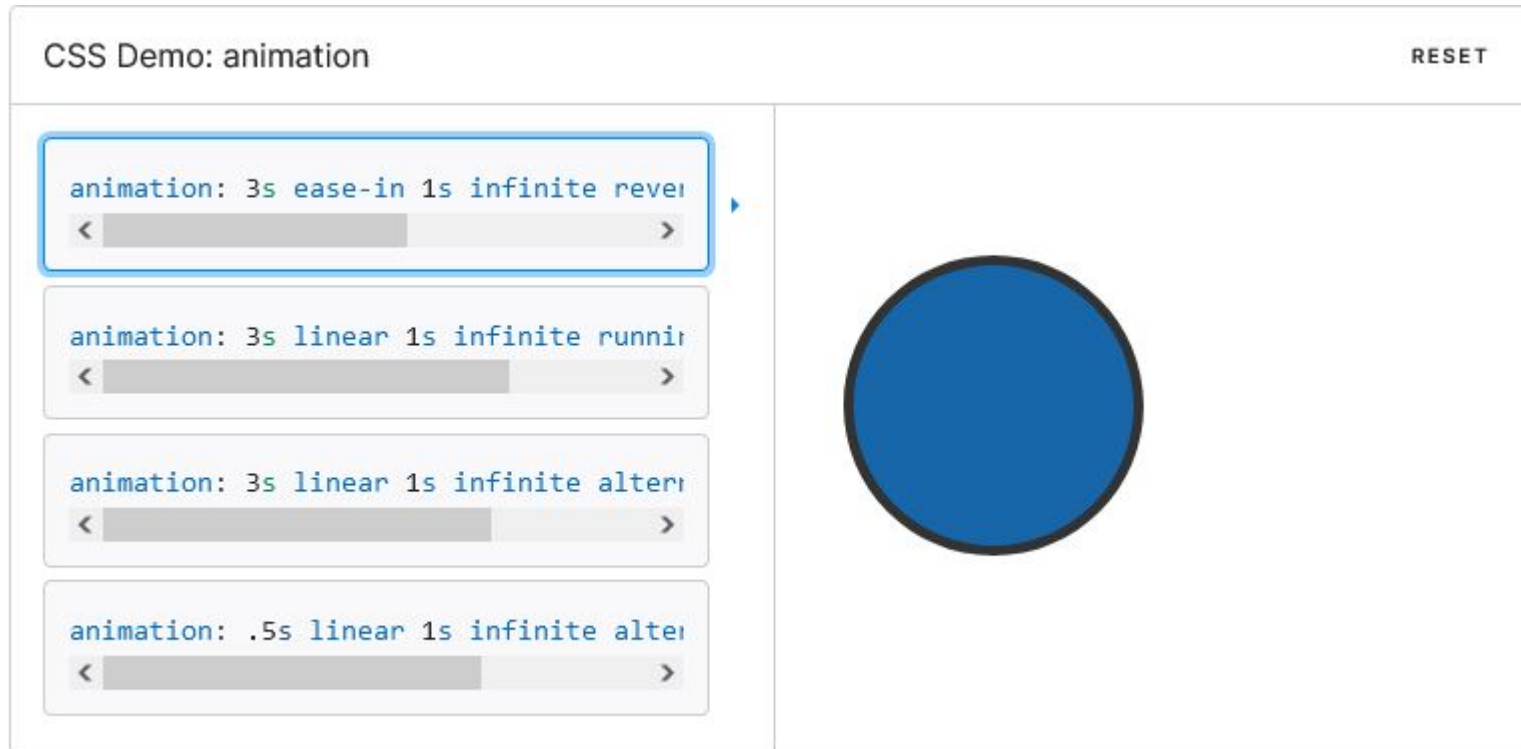
```
<div class="caja-animada"></div>
```

```
.caja-animada {  
  width: 50px;  
  height: 50px;  
  background-color: #e74c3c;  
  position: relative;  
  animation: movimiento 2s infinite alternate;  
}  
  
@keyframes movimiento {  
  from {  
    left: 0;  
  }  
  to {  
    left: 100px;  
  }  
}
```



Efectos, transiciones y animaciones

Animaciones



Efectos, transiciones y animaciones

Animaciones

```
<div class="sun"></div>
```

```
:root {  
  overflow: hidden; /* hides any part of the sun below the horizon */  
  background-color: lightblue;  
  display: flex;  
  justify-content: center; /* centers the sun in the background */  
}  
  
.sun {  
  background-color: yellow;  
  border-radius: 50%; /* creates a circular background */  
  height: 100vh; /* makes the sun the size of the viewport */  
  aspect-ratio: 1 / 1;  
  animation: 4s linear 0s infinite alternate sun-rise;  
}
```

```
@keyframes sun-rise {  
  from {  
    /* pushes the sun down past the viewport */  
    transform: translateY(110vh);  
  }  
  to {  
    /* returns the sun to its default position */  
    transform: translateY(0);  
  }  
}
```



