



Trucos y utilidades



- Usa `reset.css` Posición absoluta
- El asterisco (*) junto a un selector Utiliza atajos CSS
- Consigue formas geométricas con CSS Entiende la diferencia entre Class e ID Evita selectores CSS cuando sea posible Aprende el uso de `!important`
- Cómo centrar horizontalmente con CSS Alinear texto verticalmente en un menú
- Aplica el mismo CSS a diferentes clases o selectores Comprueba que el navegador soporta tu CSS
- Cómo reemplazar texto con una imagen Efectos hover
- Añadir espacio de borde (`padding`) dentro de la caja – `div` Capitalizar la primera letra de un párrafo
- Colores para los enlaces activos y visitados Usar `:before`
- Usar `:after`
- Redimensiona las imágenes para que entren en su sitio Selectores de hijos
- Convertir todas las letras a mayúsculas o a minúsculas Usar variables
- Truncar textos Usar condicionales
- Crear un layout masonry
- Ocultar elementos sin afectar al SEO de tu web Centrado
- Alineación vertical (para una línea de texto) tamaño de caja: caja de borde;
- `: antes`
- `: después`
- Altura de la pantalla vertical Enlaces telefónicos de estilo

Usa reset.css

Cada navegador puede interpretar a su manera las etiquetas HTML, aunque sea a niveles mínimos. Además, también algunos usuarios asigna sus propias preferencias de navegación asignando colores o tipografías propias.

El 'reset CSS' lo que hace básicamente es poner a cero o 'resetear' todos los valores antes de que apliques tus propios estilos.

Un ejemplo de reset.css sería:

```
*{  
    margin: 0;  
    padding: 0;  
    list-style: none;  
    text-decoration: none;  
    border: none;  
    outline: none;  
}
```

De esta manera puedes resetear los estilos básicos y asegurarte de que comienzas con hojas de estilo realmente en blanco.

Posición absoluta

Si quieres controlar dónde se sitúa exactamente cualquier elemento en el sitio web, necesitas posicionamiento absoluto.

Este código CSS configura la posición de un elemento a 40px de la parte superior y 50 de la parte derecha:

```
position: absolute;  
top: 40px;  
right: 50px;
```

El asterisco (*) junto a un selector

El asterisco (*) te permite seleccionar todos los elementos de un selector concreto.

Por ejemplo, si utilizas *h2 para añadir estilos CSS, lo estarás haciendo a todos los elementos en tu documento con una etiqueta <h2>.

Utiliza atajos CSS

Los atajos CSS te da una forma más sencilla y corta de escribir tu código CSS, reduce el tamaño de los archivos CSS y hace que el código sea más sencillo y limpio.

Por ejemplo:

```
.p {background-color: #ffffff;
background-image: url(background.jpg);
background-repeat: repeat;
background-position: topleft;
}
```

Quedaría convertido en:

```
.p {background-color: #fff url(background.jpg) repeat top left}
```

Consigue formas geométricas con CSS

No todas las formas en CSS tienen que ser cuadradas o rectangulares. Puedes aprovechar la propiedad CSS `shape-outside` para conseguir otro tipo de formas geométricas. Por ejemplo, puedes conseguir formas circulares o elipses para que el texto de un DIV se ajuste de esa manera alrededor de una imagen:

```
shape-outside: circle(50%);
```

```
shape-outside: ellipse(130px 140px at 20% 20%);
```

Incluso puedes conseguir otro tipo de formas basadas en imágenes externas: `shape-outside:`

```
url(/images/pelota.jpg);
```

Entiende la diferencia entre Class e ID

Los dos selectores funcionan de forma diferente y es algo que suele confundir a los principiantes.

Básicamente, una clase (class) se representa con un punto (.) y puede reutilizarse varias veces.

Un ID se representa con una almohadilla (#) y es único, no puede repetirse ni reutilizarse.

Evita selectores CSS cuando sea posible

Intenta simplificar tu CSS evitando selectores cuando sea posible. Por ejemplo: `ul li { }`

```
table tr td { }
```

Puede simplificarse a: `li { }`

```
td { }
```

Aprende el uso de !important

Un estilo marcado con !important será tenido en cuenta incluso si hay otra regla que lo machaque. Por ejemplo:

```
.p {background-color: white !important!;}
```

El código anterior hará que el color de fondo sea blanco, incluso si existen otras reglas que indiquen otros colores de fondo diferentes.

```
.page {color: black !important; color: blue;}
```

Aquí se tomará el color negro, aunque el azul se haya indicado después, ya que está marcado con !important.

Por otro lado, no es bueno abusar de !important, úsalo solo para sobrecribir en casos concretos, pero utilizarlo de forma general va a crearte un gran lío en tu código CSS.

Cómo centrar horizontalmente con CSS

En ocasiones, centrar con CSS necesita de algún truco, porque depende de qué es lo que quieres que se centre.

Centrar texto

Para centrar texto, utiliza text-align: center:

```
h1 { text-align: center;}
```

Este código CSS centraría horizontalmente las cabeceras. Igualmente puedes cambiar 'center' por 'right' o 'left' si quieres alinear a derecha o izquierda.

Centrar contenido

Para centrar contenido, por ejemplo, para centrar un DIV que ocupa un 50% del ancho de la pantalla puedes utilizar márgenes automáticos:

```
#div {  
display: block;  
margin: auto;  
width: 50%;  
}
```

Este código centraría un DIV dentro de otro DIV:

```
<div id="fuera">  
  
  <div id="dentro">Foo foo</div>  
  
</div>  
  
#dentro {  
  
  width: 50%;  
  
  margin: 0 auto;  
  
}
```

Alinear texto verticalmente en un menú

Alinear texto verticalmente con CSS puede ser tan sencillo como hacer coincidir la altura con el line-height:

```
.nav li{  
  
    line-height:75px;  
  
    height: 75px;  
  
}
```

Aplica el mismo CSS a diferentes clases o selectores

Es posible que quieras añadir el mismo atributo CSS a diferentes elementos del código, aunque se trate de elementos que no están relacionados entre sí.

Puedes hacerlo de manera sencilla, ahorrando espacio y código. Por ejemplo, para añadir el mismo atributo de color a cabecera, barra lateral y pie de página:

```
.header, sidebar, footer {  
  
color: #eee;  
  
}
```

Comprueba que el navegador soporta tu CSS

No todos los navegadores admiten las mismas propiedades, y no todos lo hacen de la misma manera.

Una buena forma de asegurarte de que un navegador acepta el CSS que estás escribiendo es utilizando @supports. Por ejemplo:

```
@supports (display: grid) {  
    .main {  
        display: grid;  
    }  
}
```

Este código solo afectará a los navegadores que soporten display: grid.

Cómo reemplazar texto con una imagen

Esta es una técnica que se utiliza sobre todo para substituir texto en cabeceras por imágenes. Por ejemplo, para substituir este título en la cabecera por una imagen:

```
<h1>Titulo de cabecera</h1>
```

Podríamos utilizar:

```
h1 {  
  
background:url("titulo-cabecera.jpg") no-repeat;  
  
text-indent:-9999px;  
  
width:400px;  
  
height:150px;  
  
}
```

Efectos hover

Los efectos hover son los que ocurren cuando alguien pasa el puntero del ratón sobre un elemento en pantalla, y se consiguen tan fácil como añadir :hover de esta manera:

```
.h1 {  
color: red;  
font-size: 80px;  
}  
  
.h1:hover {  
color: black;  
}
```

El ejemplo anterior convierte el texto de cabecera h1 en rojo, pero al pasar el ratón por encima el texto se volvería negro.

Si quieres darle un toque de efecto extra al asunto, puedes utilizar algún efecto:

```
.h1:hover {  
color: black;  
transition: all 0.5s;  
}
```

Al añadir 'transition' lo que haces es que el efecto se produzca con una transición que dure los segundos que hayas indicado.

Añadir espacio de borde (padding) dentro de la caja – div

Este truco soluciona un problema con el que te puedes encontrar muy a menudo. Al añadir un borde (padding) a un div, este borde se va a sumar al exterior del div, quedando un espacio extra alrededor.

Es decir, si tienes un div con tamaño de 200px de ancho y 200px de alto y le añades un padding de 10px alrededor, al final el div va a ocuparte 210×200 px

La solución es utilizar box-sizing: border-box;

```
.div {  
  
width: 200px;  
  
height: 200px;  
  
padding: 10px;  
  
box-sizing: border-box;  
  
}
```

De esta manera, los 10px del borde quedarían incluidos dentro del tamaño del div, que no superaría los 200px de ancho y alto.

Capitalizar la primera letra de un párrafo

Este truco CSS permite aplicar el efecto de capitalizar la primera letra de un párrafo. De esta manera, este primer carácter aparece desproporcionadamente grande, pero da un efecto elegante que llama la atención, similar a cómo comenzaban los textos de libros antiguos.

Para conseguir este efecto puedes aplicar este truco CSS:

```
p:first-letter{  
  
    display:block;  
  
    float:left;  
  
    margin:4px;  
  
    color:blue;  
  
    font-size:500%;  
  
}
```

Utilizando :first-leter, puedes hacer referencia al primer carácter de cualquier texto.

Colores para los enlaces activos y visitados

Algo que suelen olvidar los diseñadores a la hora de generar su CSS son los colores de los enlaces. Por ejemplo, puedes indicar colores diferentes para los enlaces que el usuario haya visitado y los que no haya visitado. Un ejemplo claro lo tienes en las páginas de resultado de Google, donde los enlaces no visitados y ya visitados tienen colores indicativos diferentes:

```
a:link { color: blue; } a:visited { color: purple; }
```

Usar :before

Este selector CSS te permite elegir un elemento CSS y añadirle un contenido delante cada vez que aparezca, también puede insertar un contenido.

Por ejemplo:

```
a:before {  
    content: "♥";  
    color: red;  
}
```

Este código añadiría un pequeño corazón de color rojo delante de cada enlace.

Usar :after

De la misma manera que utilizas :before, también puedes utilizar :after para insertar contenido detrás de elementos concretos.

```
p:after {  
  content: "Seguir leyendo";  
  color: #red;  
}
```

De esta forma añadirías automáticamente el texto ‘Seguir leyendo’ detrás de cada párrafo, y este texto aparece en color rojo.

Redimensiona las imágenes para que entren en su sitio

En ocasiones puedes encontrarte con que necesitas que una imagen se adapte a una anchura concreta mientras que la altura se ajusta proporcionalmente. Lo puedes hacer fácilmente así:

```
img {  
  
  max-width:100%;  
  
  height:auto;  
}
```

Selectores de hijos

Los selectores de hijos te permiten seleccionar directamente todos los hijos descendientes de otro. Por ejemplo:

```
#header > a
```

Este código te permite seleccionar todos los enlaces situados dentro de la cabecera de una vez.

Convertir todas las letras a mayúsculas o a minúsculas

En ocasiones vas a necesitar asegurarte de que todas las letras en un texto aparezcan como mayúsculas o como minúsculas. Esto puede ser particularmente útil en encabezados, para mostrar nombres de usuario, etc:

```
h1 {  
  
  text-transform: uppercase;  
}
```

Convierte todas las letras a mayúsculas.


```
h1 {  
text-transform: lowercase;  
}
```

Convierte todas las letras a minúsculas.

```
h1 {  
text-transform: capitalize;  
}
```

Convierte a mayúsculas la primera letra de cada palabra.

Usar variables

CSS tiene variables y ya no necesitas hacer uso de un preprocesador para usarlas. La ventaja de utilizar variables al escribir código en CSS es que facilita el mantenimiento a largo plazo.

Imagina que tienes que modificar el color primario de todo tu sitio web. Sin el uso de variables tendrías que revisar cada propiedad en donde se aplicó y cambiar uno a uno el valor anterior por el nuevo. Usando variables reemplazas únicamente el valor declarado y automáticamente se actualiza en el resto de las propiedades.

La sintaxis para crear una variable en CSS es:

```
:root {  
  --variable-name: value;  
}
```

El valor que puede tomar la variable es cualquier elemento que reciba una propiedad en CSS. Por ejemplo, un color. Para hacer uso de las variables utilizamos la función `var()` y dentro de la función especificamos la variable que queremos usar.

```
:root {  
  --p-color: #0044ee  
}  
  
p.variables {  
  color: var(--p-color);  
}
```

Truncar textos

Si tienes un texto muy largo puedes acortarlo con CSS. Mi forma preferida para acortarlo es usando ellipsis que no es otra cosa que los ... de toda la vida. Así damos la indicación de que hay más información de la que se está visualizando.

La propiedad que hace esta magia es `text-overflow`.

```
p.truncate {  
  overflow: hidden;  
  white-space: nowrap;  
  text-overflow: ellipsis;  
}
```

Usar condicionales

CSS te permite definir si se va a aplicar un estilo o no, dependiendo del estado de un checkbox . Se hace a través de la pseudo-clase `:checked`, la cual sirve para validar si el checkbox está en true o en false.

Esto da muchísimas posibilidades a la hora de maquetar un sitio, por ejemplo, crear un menú responsivo sin utilizar JavaScript o lo que se te ocurra.

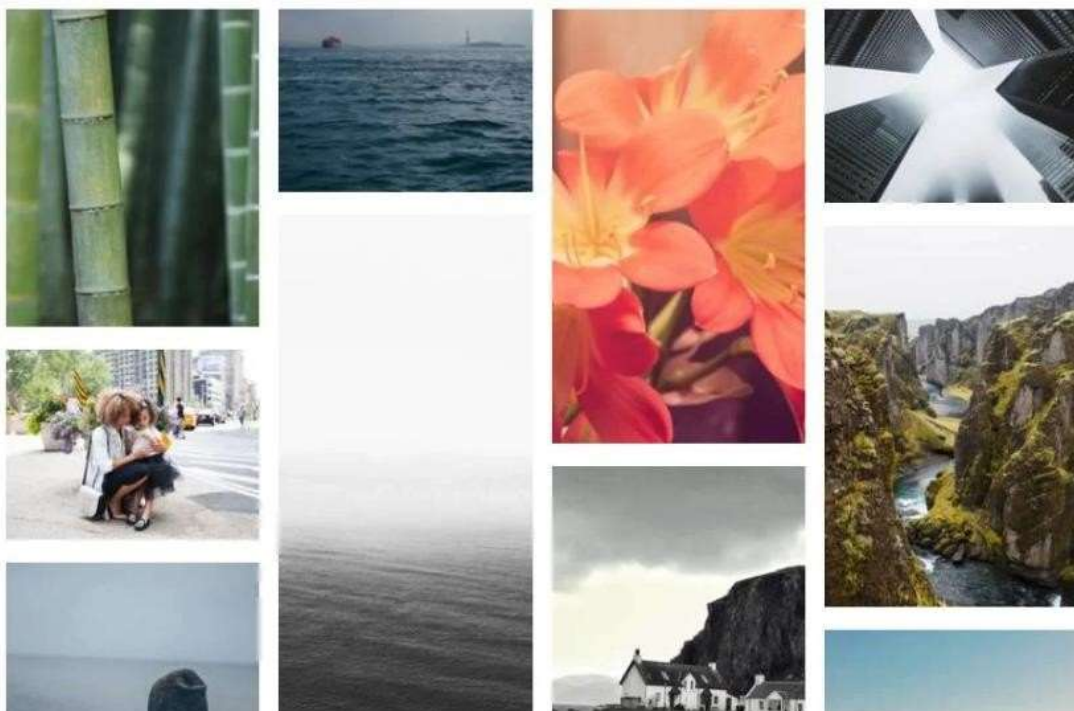
Ahora, imagina que dependiendo del estado del checkbox quieres que el background del siguiente elemento sea azul o verde.

```
<input type=checkbox> Click me!  
<div class="conditional">¿Azul o verde?</div>
```

El CSS necesario para lograrlo es el siguiente.

```
div.conditional {  
  color: white;  
  background: blue;  
  text-align: center;  
  padding: 10px;  
  margin-top: 10px;  
}  
  
input:checked + div {  
  background: green;  
}
```

Crear un layout masonry



Seguramente ya has visto el layout masonry en sitios como Pinterest. En este layout los elementos se ordenan de forma óptima teniendo en cuenta el espacio vertical disponible.

¿Sabías que necesitas menos de 5 líneas de código CSS para conseguirlo?

Como ya hemos hecho antes, primero se debe definir el HTML que se va a utilizar. En este caso, el HTML consta de imágenes de diferentes alturas. Para conseguir imágenes aleatorias puedes usar la API de Picsum.

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

En el CSS solo se tiene que indicar el ancho y el máximo de columnas que tendrá el layout. Esto se hace con las propiedades de column-width y column-count respectivamente, para obtener el ordenamiento deseado.

```
div.container {
  column-count: 4;
  column-width: 200px;
  padding: 20px;
}

img {
  width: 100%;
  max-width: 100%;
  margin-bottom: 15px;
}
```

Ocultar elementos sin afectar al SEO de tu web

Ocultar elementos como “display: none;” o “visibility: hidden;” del documento HTML de una web no es bien recibido por los motores de búsqueda. A pesar de que su uso excesivo no es recomendable, el siguiente comando CSS representa una excelente posibilidad para ocultar dichos elementos:

```
.hidden {
  position: absolute;
  top: -9999px;
  left: -9999px;
}
```

Centrado

Centrar es complicado, porque depende de lo que intente centrar. Echemos un vistazo al CSS de los elementos a centrar, según el contenido.

Texto

El texto se centra con el alineación de texto: centro;. Si lo desea a cualquier lado, use el izquierdo o el derecho en lugar del centro.

Contenido

A div (o cualquier otro elemento) se puede centrar agregando la propiedad del bloque y luego usando márgenes automáticos. El CSS se vería así:

```
#div1 {  
  display: block; margin: auto;  
  width: anything under 100%  
}
```

La razón por la que puse «cualquier cosa por debajo del 100%» para el ancho es porque si fuera 100% de ancho, entonces sería de ancho completo y no necesitaría centrarse. Es mejor tener un ancho fijo, como 60% o 550px, etc.

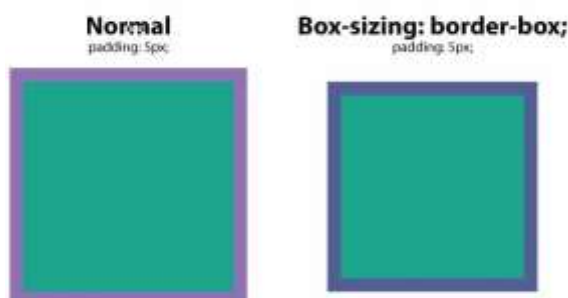
Alineación vertical (para una línea de texto)

Utilizará esto en un menú de navegación CSS, casi puedo garantizarlo. La clave es hacer que la altura del menú y la altura de línea del texto sean iguales. Veo mucho esta técnica cuando regreso y edito sitios web existentes para clientes. He aquí un ejemplo:

```
.nav li {  
  line-height: 50px; height: 50px;  
}
```

tamaño de caja: caja de borde;

Este es uno de los favoritos entre muchos diseñadores web, porque resuelve el problema de los problemas de relleno y diseño. Básicamente, cuando configura un cuadro con un ancho específico y le agrega relleno, el relleno aumenta el tamaño del cuadro. Sin embargo, con tamaño de caja: caja de borde;, esto se niega y las cajas mantienen el tamaño que deben tener.



: antes

Este CSS es un selector que le permite elegir un elemento CSS e insertar contenido antes de cada elemento con una clase específica aplicada. Supongamos que tiene un sitio web en el que desea un texto específico antes de cada etiqueta H2. Nos gustaría esta configuración:

```
h2:before { content: "Read: ";  
<span class="Apple-converted-space"> color: #F00;</span>  
}
```

Esto es muy útil, especialmente si está utilizando una fuente de iconos. Puede colocar iconos antes de ciertos elementos y aplicarlos globalmente.

My Big Bold Title

But I must explain to you how all this mistaken idea of denouncing
pleasure and praising pain was born and I will give you a complete
account of the system, and expound the actual teachings of the
great explorer of the truth, the master-builder of human happiness.
No one rejects, dislikes, or avoids pleasure itself, because it is
pleasure, but because those who do not know how to pursue pleasure
are naturally misfortunate consequences that are extremely painful.

Read: My Big Bold Title

But I must explain to you how all this mistaken idea of denouncing
pleasure and praising pain was born and I will give you a complete
account of the system, and expound the actual teachings of the
great explorer of the truth, the master-builder of human happiness.
No one rejects, dislikes, or avoids pleasure itself, because it is
pleasure, but because those who do not know how to pursue pleasure
are naturally misfortunate consequences that are extremely painful.

: después

Al igual que el selector: before, puede usar: after para insertar contenido globalmente en elementos específicos. Un uso práctico sería agregar «leer más» después de cada extracto de un blog. Así es como lo haría.

```
p:after{
  content: "-Read more...";color:#f00;
}
```

My Big Bold Title

But I must explain to you how all this mistaken idea of denouncing pleasure and praising pain was born and I will give you a complete account of the system, and expound the actual teachings of the great explorer of the truth, the master-builder of human happiness. No one rejects, dislikes, or avoids pleasure itself, because it is pleasure, but because those who do not know how to pursue pleasure rationally encounter consequences that are extremely painful.

My Big Bold Title

But I must explain to you how all this mistaken idea of denouncing pleasure and praising pain was born and I will give you a complete account of the system, and expound the actual teachings of the great explorer of the truth, the master-builder of human happiness. No one rejects, dislikes, or avoids pleasure itself, because it is pleasure, but because those who do not know how to pursue pleasure rationally encounter consequences that are extremely painful.

[Read More...](#)

Altura de la pantalla vertical

A veces, desea que una sección ocupe toda la pantalla, sin importar el tamaño de la pantalla. Puede controlar esto con vh, o ver la altura. El número anterior es un porcentaje, por lo que si desea que ocupe el 100% del navegador, debe establecerlo en 100. Puede establecerlo en un valor como 85% para acomodar un menú de navegación fijo.

Crea una clase para el contenedor y aplica la cantidad de vh que quieras que tenga. Una cosa que puede necesitar modificar es el valor de la consulta de medios para pantallas u orientaciones específicas, como teléfonos en modo vertical. Imagínese estirar una imagen horizontal para que se ajuste al modo retrato. Eso simplemente no se vería bien.

```
.fullheight { height: 85vh; }
```

Enlaces telefónicos de estilo

Si tiene un enlace que llama a un número de teléfono cuando un usuario lo toca en su teléfono, es posible que tenga problemas para diseñarlo con el selector de enlace activo tradicional. En su lugar, utilice el siguiente CSS:

```
a[href^=tel] {
  <span class="Apple-converted-space"> color: #FFF;</span>
  <span class="Apple-converted-space"> text-decoration: none;</span>
}
```