CSS

Selectores de Atributo más info.

[attr]	Selecciona elementos con el atributo attr independientemente de su valor.
[attr=val]	Selecciona elementos con el atributo attr exactamente igual a “val”.
[href ^= val]	Selecciona elementos cuyo atributo attr comienza por “val”.
[attr$=val]	Selecciona elementos cuyo atributo attr termina por “val”.
[attr*=val]	Selecciona elementos cuyo atributo attr contiene “val”.
[attr~=val]	Selecciona la palabra “val”, tanto si empieza o acaba con “val”, como si está rodeada de espacios.
[attr|=val]	Selecciona elementos con el atributo attr cuyo valor sea exactamente “val” o empieza por “val-”.

Pseudo Classes [más info.]

:active
:checked
:default
:dir()
:disabled
:empty
:enabled
:first
:first-child
:first-of-type
:fullscreen
:focus
:hover
:indeterminate
:in-range
:invalid
:lang()
:last-child
:last-of-type
:left
:link
:not()
:nth-child()
:nth-last-child()
:nth-last-of-type()
:nth-of-type()
:only-child
:only-of-type
:optional
:out-of-range
:read-only
:read-write
:required
:right
:root
:scope (en-US)
:target
:valid
:visited

Pseudo Elementos más info.

::after

::before

::first-letter

::first-line

::selection

::backdrop

::placeholder

::marker

::spelling-error

::grammar-error

Combinadores [más info.]

A B	Un elemento seleccionado por B descendiente de un elemento seleccionado por A
A, B	Un elemento seleccionado por A yo B
A > B	Un elemento seleccionado por B y es hijo directo de un elemento seleccionado por A.
AB	Selecciona A y también B.
A + B	Un elemento seleccionado por B y es el siguiente hermano de un elemento seleccionado por A (o sea, el siguiente hijo del mismo padre).
A ~ B	Un elemento seleccionado por B y es uno de los siguientes hermanos del elemento seleccionado por A (uno de los siguientes hermanos del mismo padre).

Tamaños Fijos [más info.]

px	Píxeles
mm, cm, in	Milímetros, Centímetros, Pulgadas (inches)
pt, pc	Puntos (1/72 de pulgada), Picas (12 puntos)

Tamaños Relativos más info.

%	Porcentaje del elemento padre relativo
em	Altura letra “M” en mayúscula del tamaño actual de la fuente.
rem	Tamaño de em del tamaño por defecto de la página.
ex	Altura letra “x” en minúscula. No soportada por todos los navegadores.
ch	Ancho del número 0. No soportada por todos los navegadores.
vw, vh	Tamaño de la pantalla. vw ancho y vh alto.
vmax, vmin	Es el tamaño máximo y mínimo entre vh y vw
sin unidades	Hay elementos (margin, padding...) que podemos no poner unidades. margin: 0;

Especificidad [más info.]

!important Es la regla que busca siempre antes que nada.

última regla La última regla que encuentre en el documento será la que se le aplique

específico Podemos añadir reglas muy específicas. La que se aproxime más al elemento será la que se aplique:

```
p {
    color: black;
}
```

```
p.red {
    color: red;
    background: white;
}

.red {
    color: white;
    background: red;
}
```

```
p.red > span {
    color: blue;
}
```

Modelo de Cajas más info.

En css todos los elementos se pueden comportar como cajas. Su tamaño dependerá de la propiedad “**box-sizing**”.

content-box	Valor por defecto. El tamaño de la caja no incluye el padding ni borde ni relleno. Éstos se sumarán a su width y height. Hará que la caja sea más grande de lo que deseamos.
border-box	La propiedad width y height incluirán el padding, borde y relleno. El tamaño será exactamente el que se le indique.

Media Queries más info.

Con los media Queries podemos crear estilos únicos para escenarios en concreto

```
@media (max-width: 678px) {
  .show-mobile {
        display: block;
  }
}
```

```
@media (min-width: 679px) {
  .show-mobile {
        display: none;
  }
}
```

var() más info.

En css podemos usar variables. Estas se definen con var(--nombre-variable, valorDefecto)

```
p {
    color: var(--color, #000);
}

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

calc() más info.

Función para calcular tamaños en css

```
calc(100px + 20vw)
calc(100% / 6)
calc(var(--w) / 2)
```

clamp() más info.

clamp es una función, que se usa sobretodo en las fuentes, que nos permite indicar el tamaño mínimo que tiene que tener la letra, el deseado y el máximo.

Normalmente el valor del medio (el deseado) hace referencia al tamaño de la página. Así el tamaño de la fuente se adapta a la pantalla.

clamp(mínimo, deseado, máximo)

clamp(1.5rem, 2.5vw, 4rem);

Ejemplo

transition más info.

En css los cambios de estilo en los elementos pueden tener transición. Para aplicar el efecto de transición tienes que añadir la propiedad **transition**

Propiedades de transition y valores por defecto

transition-delay: 0s
transition-duration: 0s
transition-property: all
transition-timing-function: ease

animation [más info.]

Para crear una animación necesitas 2 cosas:
Añadir la **propiedad animation** al elemento y la animación (**keyframes**) en sí.

```
@keyframes nombreAnimacion {
    from {opacity: 0}
    to {opacity:1}
}
```

```
#element {
    animation: 3s linear nombreAnimacion;
}
```

Propiedades de animation y valores por defecto

animation-name: none
animation-duration: 0s
animation-timing-function: ease

animation-delay: 0s
animation-iteration-count: 1
animation-direction: normal

animation-fill-mode: none
animation-play-state: running

FLEXBOX

Propiedad FlexBox más info.

La propiedad **display: flex** nos permite tener elementos adaptables al tamaño del padre de forma sencilla.

flex-direction	Define el eje principal y la dirección row | row-reverse | column | column-reverse
justify-content	Define cómo distribuye los elementos **horizontalmente** dentro del contenedor flex-start | flex-end | center | space-between | space-around | space-evenly
align-items	Define cómo distribuye los elementos **verticalmente** dentro del contenedor flex-start | flex-end | center | stretch | baseline
align-self	Se aplica a los hijos y reemplaza el align-items del elemento padre auto | flex-start | flex-end | center | baseline | stretch
flex-wrap	Define si los elementos tienen que estar en la misma línea o pueden cambiar de línea nowrap | wrap | wrap-reverse

Enlaces Flex

Froggy	Juego para mejorar nuestros conocimientos de Flex
Guía	Guía completa de Flex

GRID

Propiedad Grid más info.

La propiedad **display: grid** nos permite mostrar elementos en rejillas

grid-template-columns	Define la plantilla de columnas
grid-template-rows	Define la plantilla de filas
grid-gap	Define el espaciado de la rejilla
unidad fr	Podemos usar fracciones (fr) como unidad relativa
funciones	Grid tiene sus propias funciones

Función repeat() más info.

La función repeat nos permite repetir una sección de la rejilla a la columnas y las filas

Para crear 5 columnas iguales dentro del elemento grid:
grid-template-columns: 1fr 1fr 1fr 1fr 1fr;
grid-template-columns: repeat(5, 1fr);

Para crear 5 columnas: 3 columnas de 1fr y la primera y la última de 2fr:
grid-template-columns: 2fr 1fr 1fr 1fr 2fr;
grid-template-columns: 2fr repeat(3, 1fr) 2fr;

O para crear 2 series de columnas con 1fr, 2fr, y 3fr:
grid-template-columns: 1fr 2fr 3fr 1fr 2fr 3fr;
grid-template-columns: repeat(2, 1fr 2fr 3fr);

Función minmax() más info.

La función **minmax()** nos permite crear valores mínimos y máximos para cada columna o fila.

En este ejemplo, el mínimo será 100px y no habrá máximo, será automático:
grid-template-rows: minmax(100px, auto);

En este otro, el mínimo 1 fracción de la rejilla y el máximo será 10/100 de la altura de la página:
grid-template-rows: minmax(1fr, 10vh);

Posición columna/fila [más info.]

Grid nos permite indicar en qué posición ponemos cada fila o columna.
con la propiedad **grid-column** o **grid-row** podemos indicar:

1	Se sitúa en la posición 1
1 / 3	Empieza en la posición 1 y termina en la 3
span 2	Tiene un tamaño de 2
1 / span 2	Empieza en la posición 1 y tiene un tamaño de 2

Enlaces Grid

Grid Garden	Juego para mejorar nuestros conocimientos de Grid
Grid Critters	Juego para mejorar nuestros conocimientos de Grid
Guía	Guía completa de Grid

FRAMEWORKS

Pros y Contras

pros	Fácil de implementar
	Diseño funcional en la mayoría de navegadores
	Codificado con normas estándar
	Se supone que está testeado
contras	Importación de código innecesario
	Menor control de lo que está realmente sucediendo
	Limitaciones en las posibilidades de maquetación personalizada

Mejores Frameworks CSS (2022)

Bootstrap	El más popular de HTML, CSS y JavaScript
Tailwind CSS	Permite mucha flexibilidad. No contiene template por defecto
Bulma	Muy moderno basado en Flexbox
Foundation CSS	Responsive para frontend más avanzado del mundo
Materialize CSS	Basado en material design

Muy Ligeros

Frameworks con un peso inferior a 5KB.

Pure	Conjunto de módulos CSS pequeños que puede usar en cada web
Milligram	CSS minimalista.
Picnic CSS	Biblioteca ligera y hermosa.
Mobi.css	Framework CSS ligero, escalable y móvil.
mini.css	CSS mínimo, sensible y independiente del estilo.
sakura	Mínimo de CSS sin clase.
Tacit	Framework CSS para dummies, sin clases.
Chota	Un framework CSS realmente pequeño.

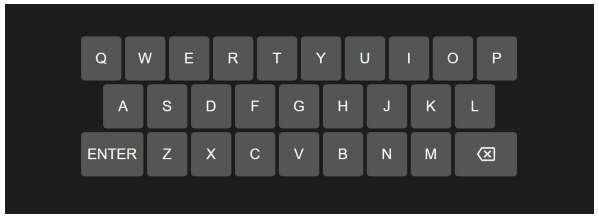
Base / Reset / Normalize

normalize.css	Alternativa moderna, HTML5-ready para los resets CSS
sanitize.css	Base para las buenas prácticas CSS.
minireset.css	Compacto reset CSS moderno.
inuitcss	Framework OOCSS extensible, escalable, Sass-compatible, OOCSS framework para proyectos UI duraderos

EJERCICIO

Ejercicio

Objetivo Crear un efecto de teclado en pantalla con html y css



Utilizar

1. **flex** para el body
2. **grid** para el teclado
3. **var()** para los colores y tamaños
4. **selector de Atributo** para las teclas
5. **clamp()** para que el teclado sea responsive

Empieza el Proyecto a través de esta base