Te vacía por defecto todos los estilos que trae css

Todos los estilos son clases

# 1. Reglas Generales

* Sintaxis de valores personalizados :
  + Usa **[]** para valores personalizados (ej: **bg-[#b33030]**).
  + No uses espacios dentro de los corchetes; reemplázalos por guiones bajos (**\_**).
  + Ejemplo: **font-[Times\_New\_Roman]**.

# 2. Colores de Fondo (bg-)

* Colores fijos :

1

<div className="bg-red-500">...</div> <!-- Color rojo predefinido -->

* Colores personalizados :

1

<div className="bg-[#b33030]">...</div> <!-- Hex personalizado -->

* Opacidad :
  + Sintaxis: **bg-{color}/{opacidad}** (0-100) o **rgba()**.

1

2

<div className="bg-red-500/50">...</div> <!-- 50% de opacidad -->

<div className="bg-[rgba(41,12,62,0.5)]">...</div> <!-- RGBA personalizado -->

# 3. Gradientes

* Lineales :

2

<div className="bg-linear-to-r from-cyan-500 to-blue-500">...</div> <!-- Gradiente horizontal -->

<div className="bg-[linear-gradient(to\_right,#3b82f6,#22c55e)]">...</div> <!-- Personalizado -->

* Radiales :

1

<div className="rounded-full bg-radial from-pink-400 from-40% to-fuchsia-700">...</div>

# 4. Texto (text-)

* Color de texto :

1

<div className="text-red-500">Texto rojo</div>

* Tamaño :

1

2

3

<div className="text-xs">Extra pequeño</div>

<div className="text-xl">Extra grande</div>

<div className="text-2xl">2xl (múltiplos de xl)</div>

* Alineación :

1

2

<div className="text-center">Centrado</div>

<div className="text-right">Alineado a la derecha</div>

# 5. Flexbox

* Alineación básica :

1

<div className="flex justify-center items-center">...</div> <!-- Centro H y V -->

* Dirección de la cuadrícula :

1

<div className="flex flex-col">...</div> <!-- Columna -->

* Espaciado entre elementos :

1

<div className="flex gap-4">...</div> <!-- 1rem (16px) de separación -->

# 6. Tamaño (Alto/Ancho: h-, w-)

* Tamaños predefinidos :
  + Escala: **0**, **1**, **2**, **4**, **8**, **16**, **32**, **64** (ej: **h-16** = 64px).

1

<div className="w-full h-screen">Ocupa todo el ancho y alto de la pantalla</div>

* Personalizados :

1

<div className="w-[200px] h-[100px]">...</div>

# 7. Margen y Padding (m-, p-)

* Sintaxis :
  + **m{dirección}-{tamaño}** (ej: **mt-4** = margen superior de 1rem).
  + Direcciones: **t** (top), **b** (bottom), **l** (left), **r** (right), **x** (horizontal), **y** (vertical).

1

2

<div className="px-4 py-2">Padding horizontal y vertical</div>

<div className="mx-auto">Centrado horizontal</div>

# 8. Grid

* Columnas :

1

<div className="grid grid-cols-3">...</div> <!-- 3 columnas -->

* Span de columnas/filas :

1

2

<div className="col-span-2">Ocupa 2 columnas</div>

<div className="row-span-2">Ocupa 2 filas</div>

# 9. Responsive Design

* Prefijos por breakpoint :
  + **sm:**, **md:**, **lg:**, **xl:**, **2xl:**.

1

<div className="md:text-center lg:text-left">...</div>

# 10. Posicionamiento (relative, absolute)

* Posición absoluta :

1

2

3

⌄

<div className="relative">

<div className="absolute top-0 right-0">...</div>

</div>

* Z-Index :

1

<div className="z-10">...</div> <!-- Por encima de otros elementos -->

# 11. Bordes (rounded-, border-)

* Redondeado :

1

2

<div className="rounded-full">Círculo</div>

<div className="rounded-tl-lg">Esquina superior izquierda redondeada</div>

* Bordes :

1

<div className="border-2 border-red-500">Borde rojo de 2px</div>

# 12. Efectos Hover y Transiciones

* Hover :

1

<div className="hover:bg-blue-500">Cambia color al pasar el mouse</div>

* Transiciones :

1

<div className="transition-all duration-300 ease-in-out">...</div>

# 13. Sombras (shadow-)

* Predefinidas :

1

2

<div className="shadow-lg">Sombra grande</div>

<div className="shadow-2xl">Sombra muy grande</div>

* Color personalizado :

1

<div className="shadow-md shadow-blue-500">Sombra azul</div>

# 14. Animaciones

* Clases predefinidas :

1

2

3

<div className="animate-spin">Gira infinitamente</div>

<div className="animate-bounce">Rebota</div>

<div className="animate-pulse">Pulso de opacidad</div>

# @apply

Para reutilizar un conjunto de clases de tailwindcss, se busca el css que contiene al @import "tailwindcss"; y hay se crea la clase que va a contener en ves de atributos a los nombres de otras clases de tailwindcss, en react es en index.css

@import "tailwindcss";

.clase\_comun{

    @apply text-3xl font-bold text-amber-200;

}

# Estilos de Texto (Bold, Italic, Normal)

Controla el grosor (font-weight ) y el estilo (font-style ) del texto.

**Clases Principales**

1. Normal (texto estándar):

<div className="font-normal">Texto normal</div>

1. Bold (texto en negrita):

<div className="font-bold">Texto en negrita</div>

1. Italic (texto en cursiva):

<div className="italic">Texto en cursiva</div>

1. Underline (subrayado):

<div className="underline">Texto subrayado</div>

1. Line-through (tachado):

1

<div className="line-through">Texto tachado</div>

**Ejemplos Combinados**

* Negrita + Cursiva :

1

<div className="font-bold italic">Texto en negrita y cursiva</div>

* Subrayado + Tachado :

1

<div className="underline line-through">Texto subrayado y tachado</div>

**Variantes de Grosor (Font-Weight)**

Tailwind incluye clases para diferentes grosores de fuente:

* **font-thin** (100)
* **font-light** (300)
* **font-medium** (500)
* **font-semibold** (600)
* **font-black** (900)

Ejemplo :

1

2

<div className="font-light">Texto ligero</div>

<div className="font-black">Texto muy grueso</div>

**Notas Adicionales**

* Personalización : Si necesitas grosores o estilos no incluidos por defecto, puedes extender la configuración de Tailwind en **tailwind.config.js**.
* Combinación con colores : Los estilos de texto funcionan junto con clases de color (ej: **text-red-500 font-bold**).

**Ejemplo Visual en Código**

<div className="text-red-500">

<p className="font-normal">Texto normal</p>

<p className="font-bold">Texto en negrita</p>

<p className="italic">Texto en cursiva</p>

<p className="underline">Texto subrayado</p>

<p className="line-through">Texto tachado</p>

<p className="font-bold italic underline">Combinación de estilos</p>

</div>

Formato para Word :

* Usa viñetas o números para listar las clases.
* Incluye bloques de código con fuente monoespaciada (ej: Courier New).
* Si es posible, agrega una breve descripción de cada estilo y su uso común (ej: "Negrita para destacar títulos").

# Group:

Es para poder disparar eventos, como el hover, pero que dependan del evento del padre, ósea por ejemplo si tenemos un contenedor que dentro tiene un botón y queremos que al hacer hover sobre el padre el botón cambie de color, entonces hay que atar de alguna manera al padre con el hijo, para eso, al padre se le agrega la clase “group” y al hijo “group-hover:propiedad”, de esta manera el hijo responde al hover del padre

        <div className="group mt-2 border-2 border-red-500 p-5 hover:bg-amber-300">

          <div className="group-hover:bg-amber-600  border-2 border-red-500 bg-black ">

            Boton

          </div>

        </div>

## Subgrupos

Cuando se tiene un grupo padre, pero esto elemento también contiene alguno otro que también va aser grupo, pasaría que los efectos se propagarían y es probable que no se quiera esto, ósea que los eventos del subgrupo solo los disparen sus elementos, para evitar esto se puede agregar una especie de identificador a cada grupo:

1. group/identificador1 y luego en el hijo group/identificador1:propiedad
2. En sus subgrupos seria entonces group/identificador2 y luego en sus hijos group/identificador2:propiedad

<div className="group/general mt-3 border-2 border-red-500 p-5 hover:bg-blue-950">

          <span className="text-red-500 ">group/general</span>

          <div className="group-hover/general:bg-amber-950 mt-5 border-2 border-red-500 bg-black ">

            <span className="text-red-500">

              group-hover/general:bg-amber-950

            </span>

          </div>

          <div className="group/individual mt-3 border-2 border-red-500 p-5 hover:bg-blue-950 bg-black">

            <span className="text-red-500 ">group/individual</span>

            <div className="group-hover/individual:bg-amber-950 mt-5 border-2 border-red-500 bg-black ">

              <span className="text-red-500">

                group-hover/individual:bg-amber-950

              </span>

            </div>

          </div>

        </div>

# overflow-hidden

Si se esta ajustando una imagen, y por ejemplo se intenta que su padre sea una circunferencia y la imagen es cuadrada, originalmente no se notaria lo del padre porque la imagen se sale del contenedor, para evitar eso se utiliza este atributo que oculta todo el exceso de la imagen

# -mt-32

Para lograr un desplazamiento fuera de un componente se puede poner negativo el margen, de modo que actúa que en ves de ser una distancia entre componentes, este se sobrepone al componente anterior (al margen que le corresponde)



# text-transparent

Pone el texto tranasparente, se suele usar junto a bg-clip-text

# bg-clip-text

Recorta el fondo para que coincida exactamente con el texto, esto sirve para mostrar por ejemplo un texto cuyo fondo sea un gradiente o una imagen. Se suele usar junto a bg-clip-text

Por ejemplo:

<div className="bg-gradient-to-r from-pink-500 to-violet-500 font-extrabold bg-clip-text text-transparent">

            con el color de un fondo gradiente

          </div>



# disabled

disabled:propiedad le pone una propiedad cuando esta disable el input o elemento

# outline y ring

**¿Qué es el outline en CSS?**

El **outline** en CSS es una propiedad que dibuja una línea alrededor de un elemento, fuera de su borde (**border**) . A diferencia del borde, el **outline** no afecta el diseño ni ocupa espacio en la caja del elemento, lo que significa que no desplaza otros elementos.

**Diferencias clave entre outline y border:**

1. Espacio ocupado :
   * El **border** forma parte del modelo de caja del elemento y puede afectar el diseño (añade dimensiones).
   * El **outline** no afecta el diseño ni ocupa espacio; se dibuja fuera del elemento sin alterar su posición.
2. Personalización :
   * El **border** puede tener diferentes estilos, grosores y colores para cada lado.
   * El **outline** es uniforme alrededor del elemento y no se puede personalizar por lados individuales.
3. Uso común :
   * El **outline** se utiliza principalmente para mejorar la accesibilidad, como resaltar elementos enfocados (por ejemplo, cuando un usuario navega con el teclado). También se usa para depuración o efectos visuales.

**Propiedades del outline en CSS**

El **outline** tiene varias subpropiedades que pueden configurarse:

1. **outline-width** : Define el grosor del contorno.

css

Copy

1

outline-width: 2px;

1. **outline-style** : Define el estilo del contorno (similar a **border-style**).

css

Copy

1

outline-style: solid; /\* Otros valores: dashed, dotted, double, none \*/

1. **outline-color** : Define el color del contorno.

css

Copy

1

outline-color: red;

1. **outline** (shorthand) : Combina las tres propiedades anteriores.

css

Copy

1

outline: 2px solid red;

1. **outline-offset** : Define la distancia entre el borde del elemento y el contorno.

css

Copy

1

outline-offset: 5px;

**¿Qué es el outline en Tailwind CSS?**

En Tailwind CSS , el **outline** se maneja mediante clases predefinidas que permiten aplicar rápidamente estilos de contorno a los elementos. A continuación, te explico cómo funciona:

**Clases principales para outline en Tailwind:**

1. Estilo del contorno :
   * **outline-none**: Elimina el contorno (útil para eliminar el foco predeterminado en botones y enlaces).
   * **outline**: Aplica un contorno sólido predeterminado.
   * **outline-dashed**: Aplica un contorno con estilo de líneas discontinuas.
   * **outline-dotted**: Aplica un contorno con estilo de puntos.
2. Color del contorno :
   * Puedes usar clases como **outline-red-500**, **outline-blue-300**, etc., para definir el color del contorno.

1

<button className="outline outline-red-500">Botón con contorno</button>

1. Grosor del contorno :
   * Tailwind no tiene clases específicas para el grosor del contorno, pero puedes usar la clase **ring** para lograr un efecto similar (más adelante hablaremos de **ring**).
2. Offset del contorno :
   * Tailwind no incluye clases directas para **outline-offset**, pero puedes agregarlo manualmente usando clases personalizadas o extendiendo la configuración de Tailwind.

**Ejemplo en Tailwind CSS**

Aquí tienes un ejemplo práctico de cómo usar el **outline** en Tailwind:

html

Copy

1

2

3

<div className="outline outline-2 outline-red-500 outline-offset-4">

Este elemento tiene un contorno rojo.

</div>

**Explicación:**

* **outline**: Habilita el contorno.
* **outline-2**: Define el grosor del contorno (2px).
* **outline-red-500**: Define el color del contorno.
* **outline-offset-4**: Agrega un espacio de 4px entre el borde del elemento y el contorno.

**¿Qué es ring en Tailwind CSS?**

Además del **outline**, Tailwind introduce una propiedad llamada **ring**, que es una implementación más avanzada y visualmente atractiva del concepto de contorno. El **ring** es especialmente útil para crear efectos de foco o resaltado.

**Clases principales para ring:**

1. Activar el anillo :
   * **ring**: Agrega un anillo alrededor del elemento.

1

<button className="ring ring-red-500">Botón con anillo</button>

1. Color del anillo :
   * Usa clases como **ring-red-500**, **ring-blue-300**, etc.

1

<div className="ring ring-blue-300">Anillo azul</div>

1. Grosor del anillo :
   * **ring-0**: Sin anillo.
   * **ring-1**: Anillo delgado.
   * **ring-2**: Anillo más grueso.

1

<div className="ring ring-2 ring-green-500">Anillo grueso</div>

1. Opacidad del anillo :
   * Usa **ring-opacity-{value}** para ajustar la transparencia del anillo.

1

<div className="ring ring-red-500 ring-opacity-50">Anillo semitransparente</div>

1. Offset del anillo :
   * Usa **ring-offset-{value}** para definir la distancia entre el elemento y el anillo.

1

<div className="ring ring-red-500 ring-offset-4">Anillo con offset</div>

**Cuándo usar outline vs ring en Tailwind**

1. Usa **outline** cuando :
   * Necesites un contorno simple y básico.
   * Quieras eliminar el contorno predeterminado de elementos enfocados (**outline-none**).
2. Usa **ring** cuando :
   * Quieras un efecto de resaltado más visual y atractivo.
   * Necesites ajustar la opacidad, grosor o distancia del contorno.
   * Trabajes con efectos de foco o hover dinámicos.

**Conclusión**

* CSS nativo : El **outline** es una herramienta útil para resaltar elementos sin afectar el diseño.
* Tailwind CSS : Proporciona clases convenientes como **outline** y **ring** para aplicar estos estilos de manera rápida y consistente.

Ambos son útiles dependiendo del caso de uso. Si necesitas algo sencillo, usa **outline**. Si buscas un efecto más moderno y personalizable, opta por **ring**. 😊

# invalid

En los input, cuando por el html es invalido se le pueden dar mas propiedades y concatenar estados

invalid:propiedad

invalid:estado:propiedad

<input

            className="invalid:focus:ring-red-400 focus:outline-none focus:ring-1 focus:ring-purple-600 border border-gray-300 w-full px-3 py-2 mb-4 rounded-md"

            type="email"

            placeholder="Correo"

          />