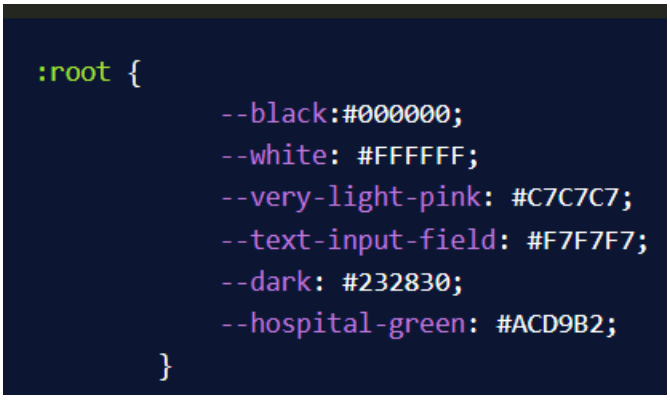
Curso Práctico de Frontend Developer

La **maquetación** o **diagramación web** consiste en transformar un diseño gráfico —boceto— (hecho por UX/UI en Figma o Scketch) en una interfaz funcional en términos de programación que entienda un navegador o dispositivo específico.

¿Cuál es la utilidad de un sistema de diseño?
La principal ventaja de implementar un sistema de diseño es que facilita las tareas de diseñadores y desarrolladores en el proceso de creación. También agiliza la toma de decisión entre equipos.

- **Variables en CSS:** En CSS, llamamos variables a las propiedades personalizadas. Contienen valores específicos que se pueden reutilizar muchas veces en un documento.
  - Se establecen mediante la notación de dos guiones.
  - Ejemplo: --nombre-variable: valor;
  - Se acceden mediante la función var()
  - Ejemplo: propiedad: var(--nombre-variable);
  - Normalmente las declaramos dentro del selector **:root** para que su alcance (scope) sea global.



```
:root {
        --black:#000000;
        --white: #FFFFFF;
        --very-light-pink: #C7C7C7;
        --text-input-field: #F7F7F7;
        --dark: #232830;
        --hospital-green: #ACD9B2;
      }
```

También puedes nombrar a tus variables según su función. Ejemplos: --background-color, --primary-color, etcétera.

- **Display Grid** para centrar elementos: con solo dos líneas de código podemos centrar nuestro contenido.
  - display: grid; place-items: center;
- El shorthand property **place-items** te permite alinear elementos, tanto horizontal como verticalmente, en un contendor con Grid o Flexbox. Es decir, es la abreviatura de las propiedades **align-items** y **justify-items**. Si no le estableces el segundo valor va a utilizar el primero para ambas alineaciones.

¿Cómo ordenar los estilos?
Una manera de hacerlo es según su propósito. Siguiendo el siguiente orden:

1. Posicionamiento
2. Modelo de caja
3. Tipografía
4. Visuales
5. Otros

- **Figure** <figure> <img /> </figure> es una etiqueta que permite almacenar una imagen en su interior. Es una mejor práctica comparada con usar solamente un contenedor div. Como complemento al contenedor figure, se utiliza la etiqueta figcaption <figcaption></figcaption>, que permite darle una pequeña descripción a la imagen, como el autor, fuente o algo por el estilo, que se mostrará usualmente abajo de la imagen.
  - Es importante considerar que la etiqueta **figure** no es únicamente para imágenes;
  - El elemento HTML **<figure>** representa contenido independiente, a menudo con un título. Por lo general, se trata de una imagen, una ilustración, un diagrama, un fragmento de código, o un esquema al que se hace referencia en el texto principal, pero que se puede mover a otra página o a un apéndice sin que afecte al flujo principal.

```
<figure>
          <img
            src="./pics/tinified/small.jpg"
            alt="Es una imagen de un perrito"
          />
          <figcaption>Es una imagen de un perrito</figcaption>
</figure>
```

- El elemento **picture** ha sido diseñado para proveer soporte nativo para imágenes adaptativas en HTML5. Cuando se utiliza en conjunto con <source> e <img> actúa como una imagen que será cargada de manera diferente de acuerdo a las propiedades del dispositivo en el que se muestre. En otras palabras, los navegadores que lo soporten cargarán una imagen diferente (provistas por los elementos source) para cada tipo de dispositivo especificado.

IMAGEN EN LA SIGUIENTE PAGINA

```
<picture>
  <source media="(min-width: 1280px)" srcset="/assets/images/isaac-newton-l.jpg">
  <source media="(max-width: 520px)" srcset="/assets/images/isaac-newton-s.jpg">
  <img class="float-right" src="/assets/images/isaac-newton.jpg" alt="Sir Isaac Newton">
</picture>
```

- Un **menú desplegable o, lista desplegable**, es un elemento de control gráfico que muestra al usuario una variedad de opciones de una categoría que puede elegir para realizar una acción como una compra. Esta tiene dos estados: activa o inactiva. Cuando está inactiva, enseña un solo valor.

- La propiedad **object-fit** indica cómo el contenido de un elemento reemplazado, por ejemplo un <img> o <video>, debería redimensionarse para ajustarse a su contenedor.

¿Qué logra **object-fit** en CSS?
A las **imágenes** le aplicamos la propiedad object-fit, ya que resuelve como el contenido se ajustará a su contendor. Sus valores pueden ser:

- **contain** → mantiene la relación de aspecto mientras le ajusta dentro del contendedor.
- **cover** → mantiene la realción de aspecto, pero la ajusta para llenar el contenedor.
- **fill** → modifica su tamaño para llenar el contenedor.
- **none** → no se redimensiona.
- **scale-down** → el contenido se dimensiona como si none o contain estuvieran especificados, lo que resulta en un tamaño de objeto concreto más pequeño.

¿Qué es la propiedad **transform** CSS?
La propiedad CSS **transform** te permite modificar el espacio de coordenadas del modelo de formato visual CSS. Usándola, los elementos pueden ser trasladados, rotados, escalados o sesgados de acuerdo a los valores establecidos.
transform: none; es su valor por default. Ejemplos básicos:

- **transform: translateX(40px);** Move the element on the horizontal axis.
- **transform: translateY(20px);** Move the element on the vertical axis.
- **transform: translateY(100%);** You can use percentage values: the percentage is relative to the **element itself**, and not the parent.
- **transform: translate(20px, -10%);** You can use translate() with two values: the first value is for the **horizontal** axis and the second value is for the **vertical** axis.
- **transform: scaleX(1.5);** Scale the element on the horizontal axis.
- **transform: scaleY(0.4);** Scale the element on the vertical axis.

- **transform: scaleY(-2);** You can use negative values: it will invert the element.
- **transform: scale(0.8, 0.8);** You can use scale() with two values: the first value is for the **horizontal** axis and the second value is for the **vertical** axis. By using the same value for both, you can scale proportionally.
- **transform: rotate(45deg);** Rotate the element. You can use:
  - **degrees** from 0 to 360deg
  - **gradians** from 0 to 400grad
  - **radians** from 0 to 2πrad
  - **turns** from 0 to 1turn
- **transform: skewX(15deg);** Skew the element on the horizontal axis.
- **transform: skewY(45deg);** Skew the element on the vertical axis.
- **transform: skew(10deg, -20deg);** You can use skew() with two values: the first value is for the horizontal axis and the second value is for the vertical axis.
- **transform: rotate(5deg) scale(1.1, 1.1) translate(-20%, 30px);** You can combine multiple transformations by separating them with a space.