

Tailwind CSS

Instalación e inicio de Tailwind css

Descargar [node.js](#)

Debemos descargar e instalar node js para usar su npm (Node Package Management) en la instalación de tailwindcss.

Normalmente junto con node.js se instala el [npm](#), pero no la última versión, así que procedemos a instalarla: `npm install -g npm`

Comprobamos la versión que tenemos de node.js y npm.

`node -v`

`npm -v`

Iniciamos tailwind css

`/*`

`npx tailwindcss -i ./src/input.css -o ./build/css/style.css --watch`

Ver cambios

SIEMPRE hay que realizar esta línea si salimos de VSC, si no, no observaremos cambios.

Probar: `npm run tailwind`

Debería hacer lo mismo

`*/`

`npx tailwindcss init`

Te dirá que necesita instalar los siguientes paquetes: `tailwindcss@3.3.2`

Procedemos a decir que si (y)

Vemos que aparece un archivo: `tailwind.config.js`

Creamos los Folders

build: Dentro de este directorio estará el archivo `index.html`


src: Crear esta carpeta, más adelante se usará.

El archivo `tailwind.config.js` se situará fuera de los folders por defecto.

Decirle a Tailwind donde está nuestro HTML

Le decimos dónde conseguir las clases que estamos usando para incluir en la carpeta de estilo que crearemos con Tailwind.

Abrir `tailwind.config.js` y dentro de `content` poner lo siguiente:



```
1  /** @type {import('tailwindcss').Config} */
2  module.exports = {
3    content: ['./build/*.html'],
4    theme: {
5      extend: {},
6    },
7    plugins: [],
8  }
```

Esto hace que busque no solo en `index.html` sino en todos los archivos HTML que se encuentren dentro de la carpeta `build`.

Lista de youtube con la guía paso a paso: [Lista de YouTube](#). Canal: [Dave Gray](#).

Guía de instalación oficial de Tailwind css: [Guía](#).

Recursos de la guía: [Resources](#)

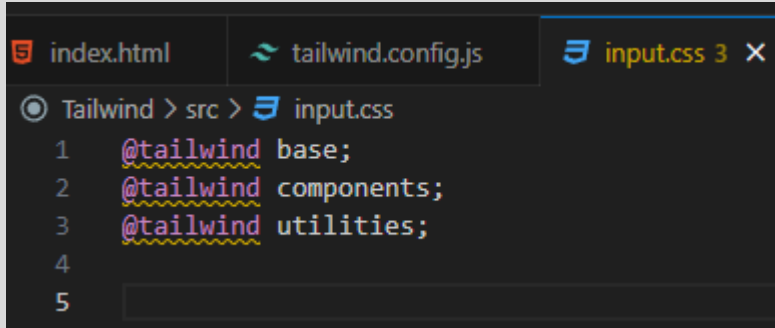
Creemos un directorio fuente (source directory)

Dentro de este crearemos el archivo input.css e introduciremos lo siguiente:

@tailwind base;

@tailwind components;

@tailwind utilities;



```
index.html  tailwind.config.js  input.css 3 X
Tailwind > src > input.css
1  @tailwind base;
2  @tailwind components;
3  @tailwind utilities;
4
5
```

Con esto, Tailwind compila el CSS desde el archivo input.css y lo crea dentro de la carpeta *build*. Con esto también podemos definir nuestras clases.

Después de guardar, veremos que Visual Studio Code (VSC) detecta que hay errores, esto lo solucionamos de esta manera:

File → Preferences → Settings

Se abre una consola y debemos poner lo siguiente:

Unknown → Ignore

IMPORTANTE

/* Yo no hice esto, no vi bien ignorar estos errores así que utilicé la extensión de VSC:

Post CSS Language Support y ya no aparecen más. Más info [aquí](#)

También instalé la extensión Tailwind CSS IntelliSense. Trabaja bien con la extensión de Post CSS

Para abrir los settings.json en VSC:

Ctrl + Shift + P

Preferences: Open User

Por ahora lo dejamos en blanco, no hace falta realizar cambios

*/

Entrada y salida de archivos

npx tailwindcss -i ./src/input.css -o ./build/css/style.css

-i Indica la entrada de los archivos de css

-o indica la salida de los archivos de css

Mirar esta pág: [Dev.to](#)

Nos crea la carpeta con el output

Con esto le decimos que use las clases que nosotros vamos a definir

Compilar el css

npx tailwindcss -i ./src/input.css -o ./build/css/style.css --watch

Ver cambios

SIEMPRE hay que realizar esta línea si salimos de VSC, si no, no observaremos cambios.

Lista de youtube con la guía paso a paso: [Lista de YouTube](#). Canal: [Dave Gray](#).

Guía de instalación oficial de Tailwind css: [Guía](#).

Recursos de la guía: [Resources](#)

Extensiones a instalar

Tailwind CSS IntelliSense

Para autocompletar y otras funcionalidades

Te enseña colores y otras clases si pasamos el ratón por encima y cuando estamos escribiendo.

PostCSS Language Support

Preprocesador y también transpila código moderno CSS.

Opcionales:

Live Server

Ver cambios mientras aplicas código.

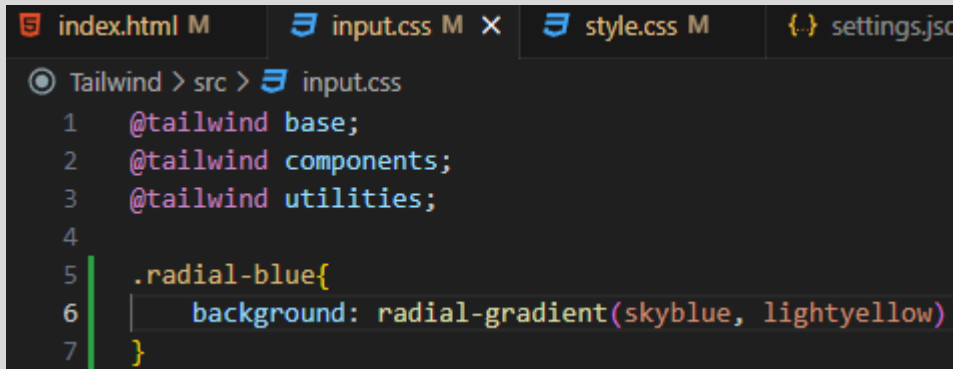
Inline Fold

Organiza el código de Tailwind en el HTML para que sea más visible.

Añadir nuestras propias clases

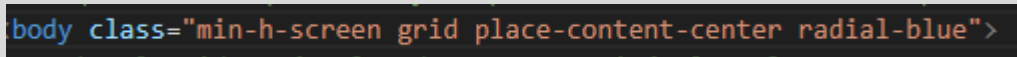
Hay elementos que tailwind no da apoyo o suministra (supply) así que nosotros las debemos definir, por ejemplo: radio gradient class

Primero, debemos ir al archivo input.css y definir la clase, por ejemplo:



```
index.html M  input.css M x  style.css M  {} settings.js
Tailwind > src > input.css
1  @tailwind base;
2  @tailwind components;
3  @tailwind utilities;
4
5  .radial-blue{
6    background: radial-gradient(skyblue, lightyellow)
7  }
```

Una vez hecha la clase, debemos ir al html y llamarla:



```
body class="min-h-screen grid place-content-center radial-blue">
```

Subir a GitHub

Lista de youtube con la guía paso a paso: [Lista de YouTube](#). Canal: [Dave Gray](#).

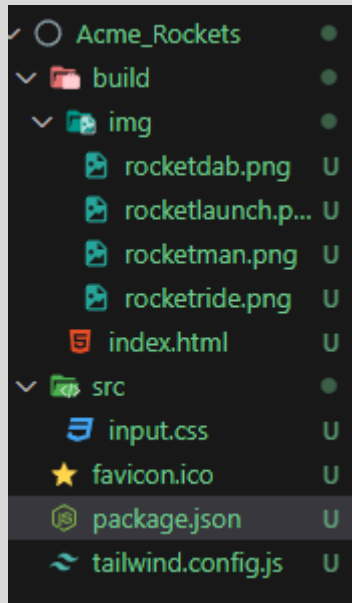
Guía de instalación oficial de Tailwind css: [Guía](#).

Recursos de la guía: [Resources](#)

Website with TailWind

[Iniciamos Tailwind y creamos los folders](#)

(Mirar instalación e inicio de TailWind)



Iniciamos npm

npm init -y (La "-y")

Recordar que ya lo teníamos instalado de antes a nivel global a su última versión

Es lo mismo que [aquí](#) pero dentro de scripts y dentro de **package.json**

/*Te crea un archivo **package.json** que es otro tipo de configuración */

Development dependency

npm i -D prettier-plugin-tailwindcss

Instalado dentro del **package.json** y crea la carpeta **node_modules**

Crear archivo .gitignore

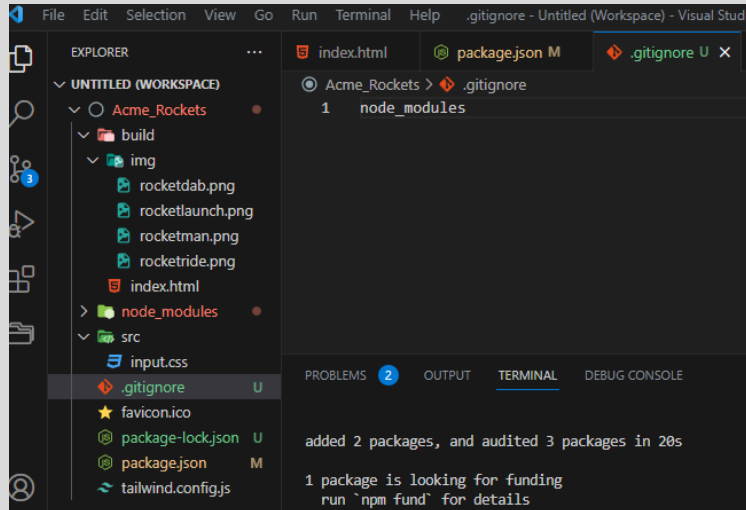
Esto es para que la carpeta **node_modules** no los envíes a github, no interesa.

Lista de youtube con la guía paso a paso: [Lista de YouTube](#). Canal: [Dave Gray](#).

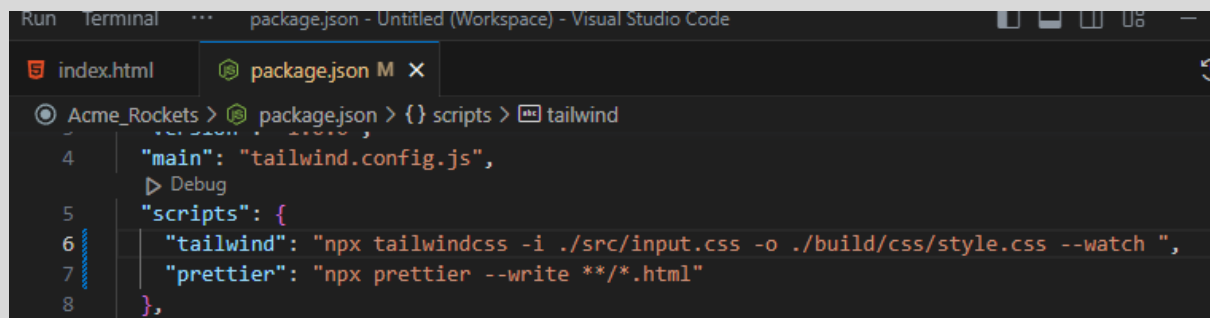
Guía de instalación oficial de Tailwind css: [Guía](#).

Recursos de la guía: [Resources](#)

Poner dentro el archivo que queremos que no se envíe (node_modules)



Dentro de package.json lo siguiente:



Esto es para decirle a prettier solo busque html y solo formatee. También es importante por el tema del orden de clases. Minuto 9:28 del 2º video.

Terminal:

npm run tailwind

Se creará el folder de css que estará dentro de build