**PostCSS**

आइए आपके जीवन को आसान बनाएं ✌

**What is postcss :**

Just blank your mind for now and remember only one thing that postcss is just a tool that will provide you with some syntax prefixes, by that prefixes you can write your CSS in an accessible, more readable, and compressed way.

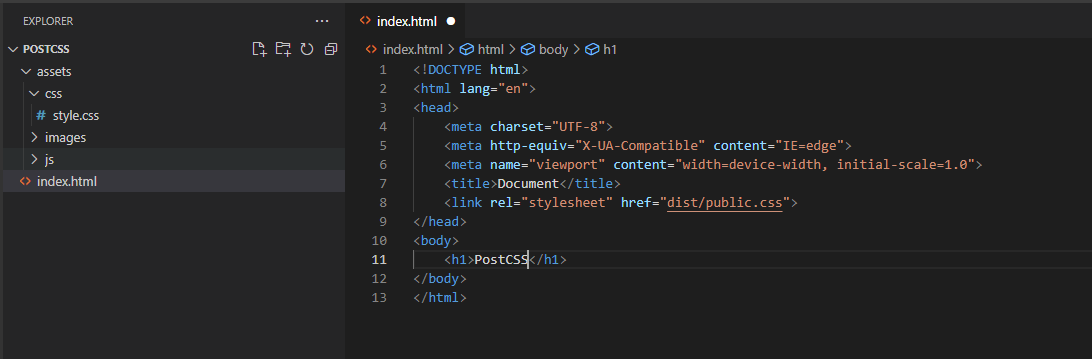
Just be with me and we will learn PostCSS together and also after this you will know why you should use PostCSS.

Let’s get started by instraling PostCSS

**Installation :**

FIrst we need to install node.js to provide a development environment for postcss.

Download node [here](https://nodejs.org/en/download/).

After installing the node we need to install post css in your root dir of the web or WordPress theme. 

As you can see the folder structure of my web page. It has just only index.html and some other stuff in the assets folder.

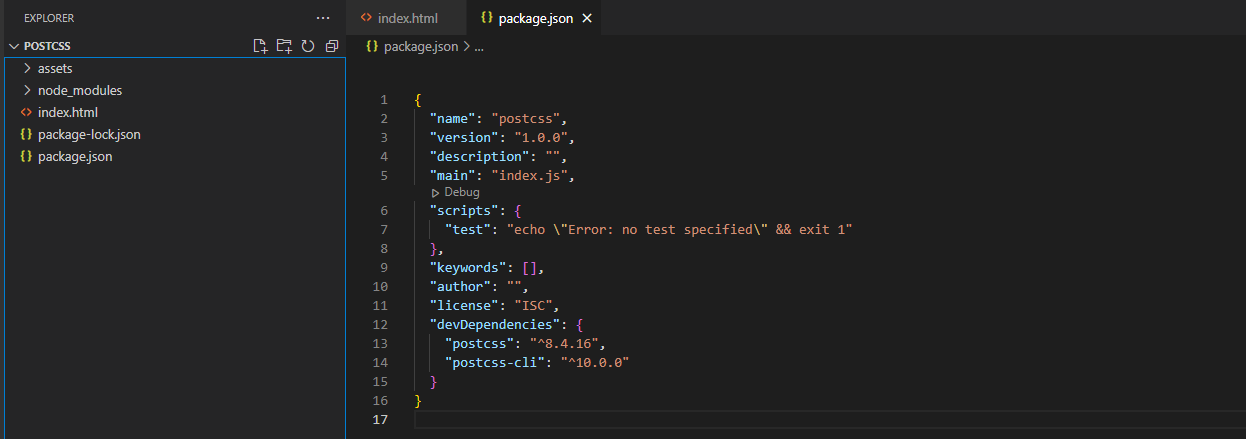
Now I will run the following command:

npm init -y

npm i postcss

npm i -D postcss postcss-cli

Now you have these files and folders



Let's move further and think that you are creating one web page, you will do your css in one file named style.css. After completing your website your style.css has around 2000 lines of code which include navbar css, buttons css, sliders css, media query, etc.

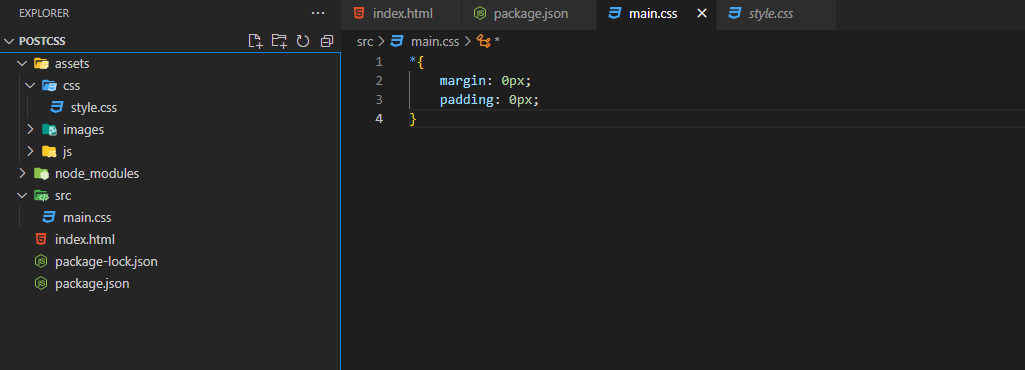
This is not a good way to write css cause if need to change some media query then you go down to line no 1538 and change it 😮. Just suppose if your client says that I don't like this theme color 😒 and I need all buttons should have a green background. At that point, you have 26 buttons on your website and now you will find all the css of each button and change it. Is this an organized way to code🙄??

Let's organize our code by separating some files and by creating some folders.

* **base/**
* **components/**
* **layout/**
* **pages/**
* **themes/**
* **abstracts/**
* **vendors/**

For more details about this please visit [this](https://sass-guidelin.es/#the-7-1-pattern).

For the only learning purpose, I will create one folder called src at the root of the website. And create one new file called main.pcss ( you can name it main.css as well ).



Let’s do one exercise.

In postcss we have two types of css files one is to write postcss and one is to read vanilla css for the browser. In our case, we have style.css in assets and it is linked to the webpage. So whenever a website load website have style.css in the assets folder. We are writing postcss in main.css and it will convert into vanilla css. But how to convert it??

Open package.json and go down to

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1"

},

Change test to postcss and write following command in doublecots.

"scripts": {

"postcss": "postcss ./src/main.css –output ./assets/css/style.css –watch"

},

Now this is the fun part. Run the following command :

npm run postcss

Now you can see that your style.css file have the same line of code that is in the main.css

Let's do one more thing create one new file button.css in the src folder and in the main.css file import this file by

@Import url(button.css);

And remove other code, do some code in button.css file and save both files. Go and see style.css you will find there is the same code as main.css and that is obvious. But we need to have a code of button.css how we can do that?

Let’s install one plugin of PostCSS called [‘postcss import’](https://github.com/postcss/postcss-import)

npm install -D postcss-import

npm i postcss-partial-import

After that, we need to create one js file called postcss.config.js to use all the plugins of the postcss.

The code for postcss.config.js :

module.exports = {

plugins: [

require('postcss-import')

],

}

Now re-run the command

npm run postcss

And see the style.css, you will have all code of button.css

Now you have the ability to separate your css files as per your requirements and just import them into the main.css

Let’s move further and install one more famous plugin of the post css called [postcss nested](https://github.com/postcss/postcss-nested)

npm install --save-dev postcss postcss-nested

And just require it in the postcss.config.js file by

require('postcss-nested)

Now you have the ability to use nested plugins and write your css like this

button{

color: white;

background-color: black;

&:hover{

color: red;

}

}

Output in style.css

button{

color: white;

background-color: black;

}

button:hover{

color: red;

}

This is amazing right 😄??

We are not done yet.

Let’s install two more plugin purgecss and cssnano.

This plugins will minify your css and reduce its size.

Let’s install both of the plugins by this commands

npm i -D @fullhuman/postcss-purgecss postcss

npm install cssnano postcss --save-dev

And not its time to update our postcss.config.js file

module.exports = {

plugins: [

require('postcss-import'),

require('postcss-nested'),

require('@fullhuman/postcss-purgecss')({

content: ['./\*\*/\*.html']

}),

require('cssnano'),

],

}

That’s it we are good to go with the minified version of our css.

**Thank you for Reading**

क्यों हो गइना जिंदगी आसान ✌