**Node JS**

Node module peut être soit dans le dossier dans lequel on se trouve actuellement, soit dans le dossier utilisateur.

Angular CLI : Command Line Interface (interface en ligne de commande).

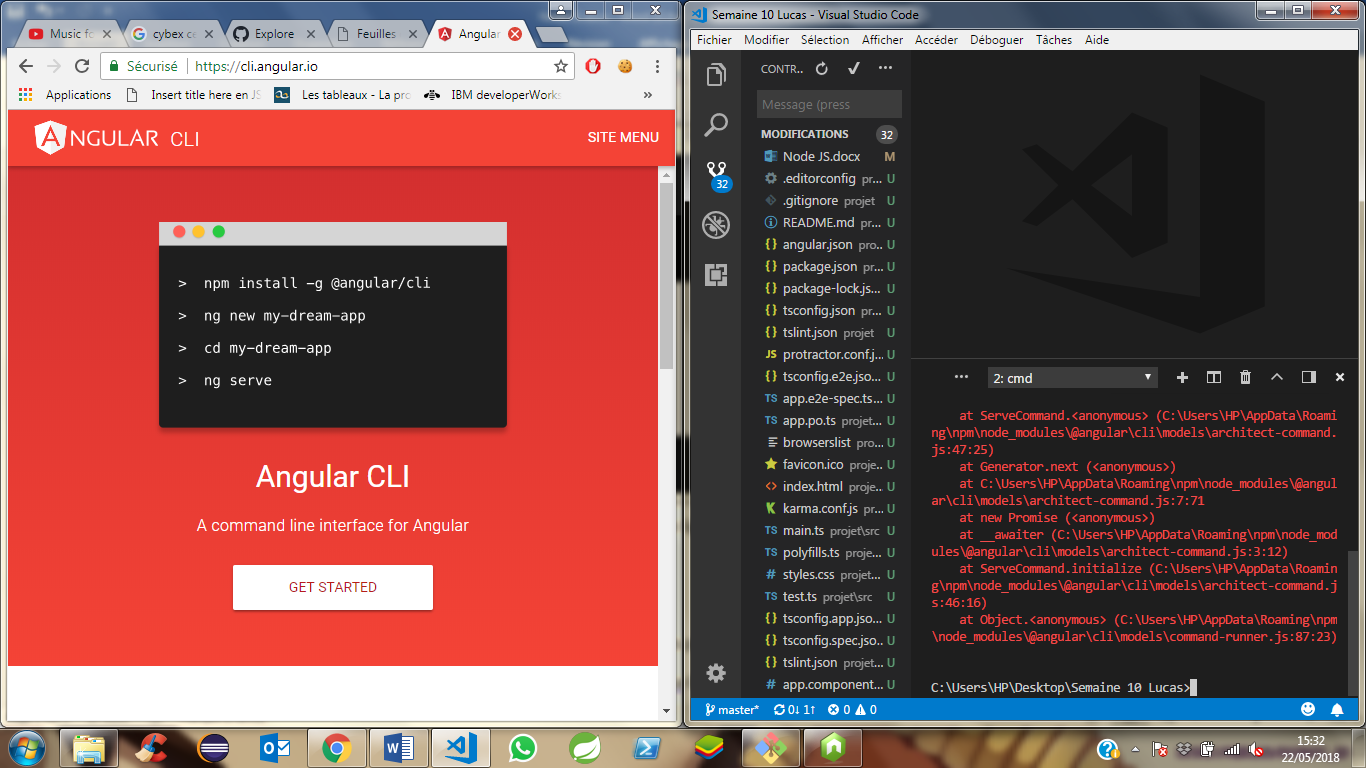
Install local vs install global :

npm i –g 🡪 global (dossier utilisateur)

npm i 🡪 local (installe uniquement dans les projets présents dans le dossier workspace)

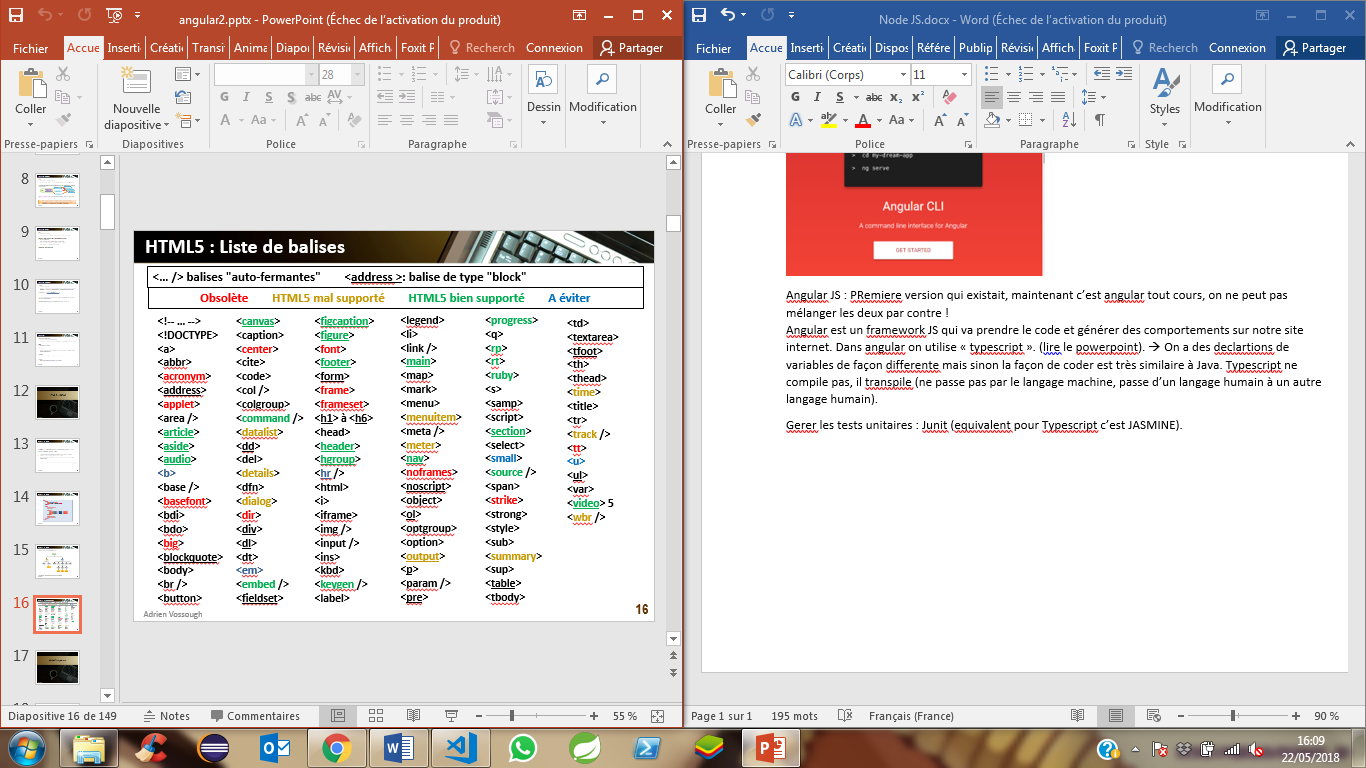
Il faut faire gaffe de TOUJOURS, TOUJOURS (grâce a GIT) avoir un truc a montrer qui fonctionne et qui donne un aperçu de ce que notre site fait.

Pour installer Angular :

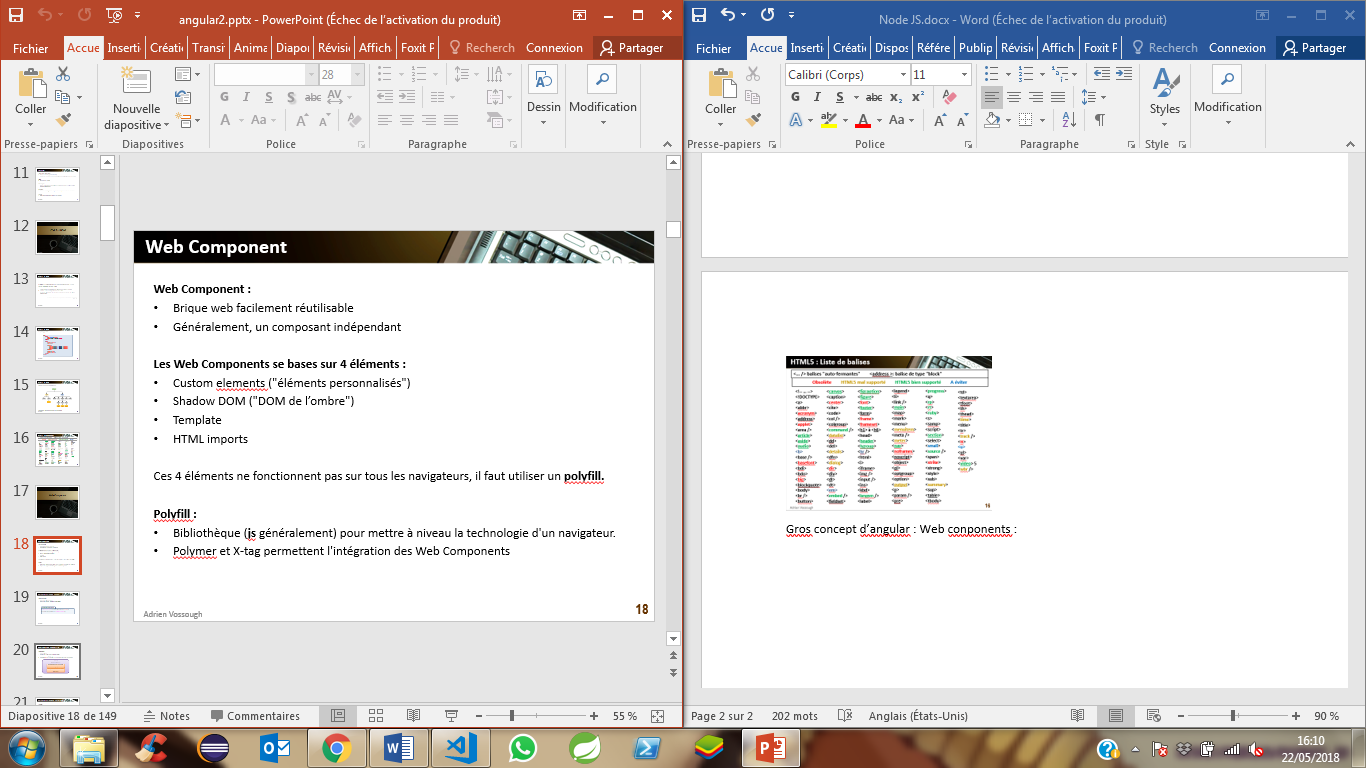
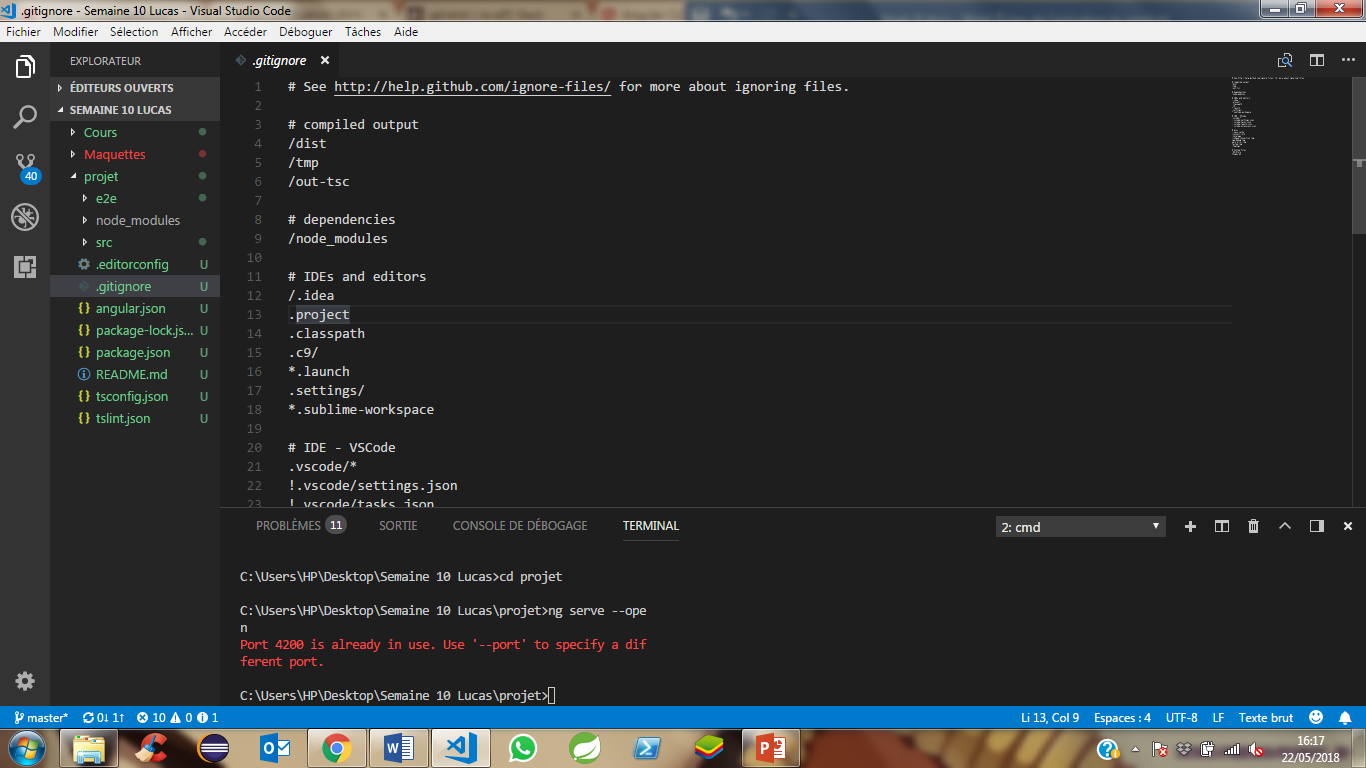


Angular JS : PRemiere version qui existait, maintenant c’est angular tout cours, on ne peut pas mélanger les deux par contre !  
Angular est un framework JS qui va prendre le code et générer des comportements sur notre site internet. Dans angular on utilise « typescript ». (lire le powerpoint). 🡪 On a des declartions de variables de façon differente mais sinon la façon de coder est très similaire à Java. Typescript ne compile pas, il transpile (ne passe pas par le langage machine, passe d’un langage humain à un autre langage humain).

Gerer les tests unitaires : Junit (equivalent pour Typescript c’est JASMINE).

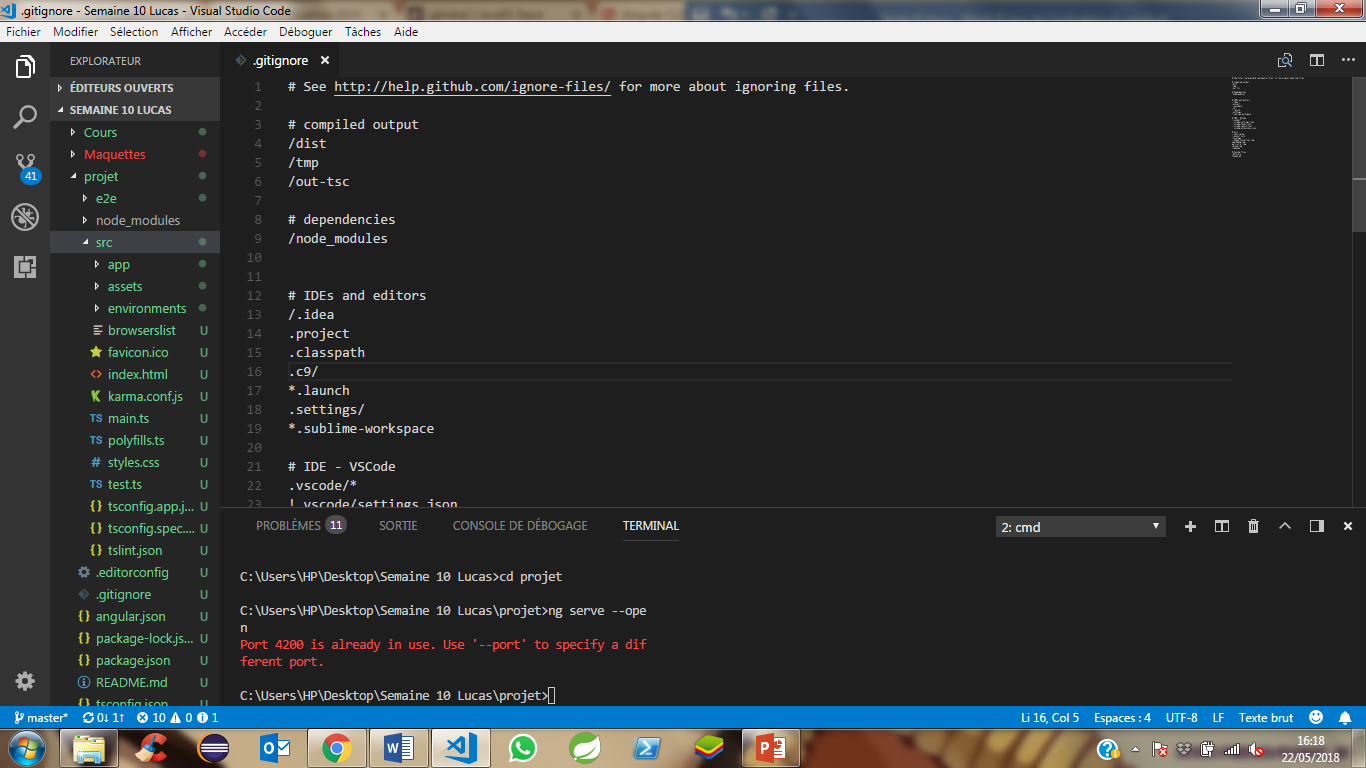


Gros concept d’angular : Web conponents :

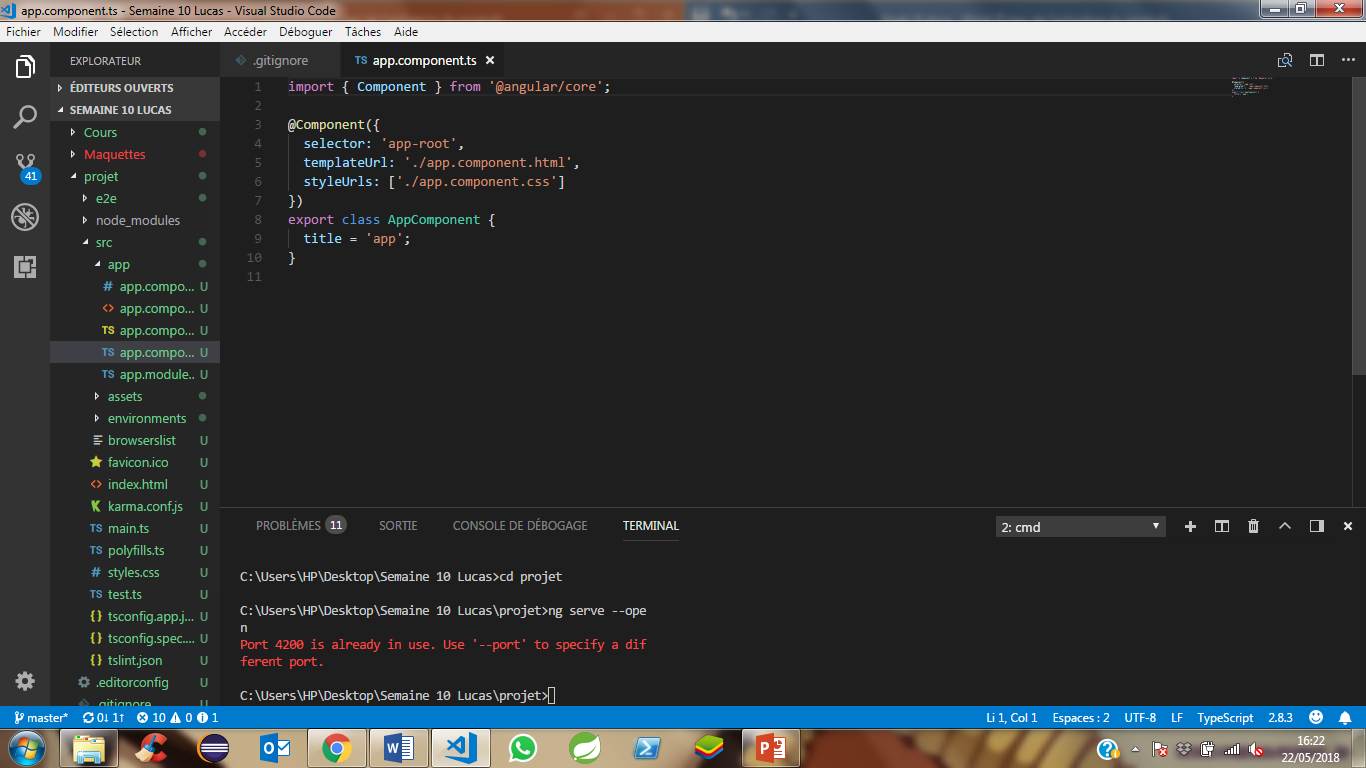
  


Dans le .gitignore on entre tout les fichier qu’on ne veut pas mettre dans le GIT.

Le src c’est notre application, angular genere plein de fichiers pour notre projet :

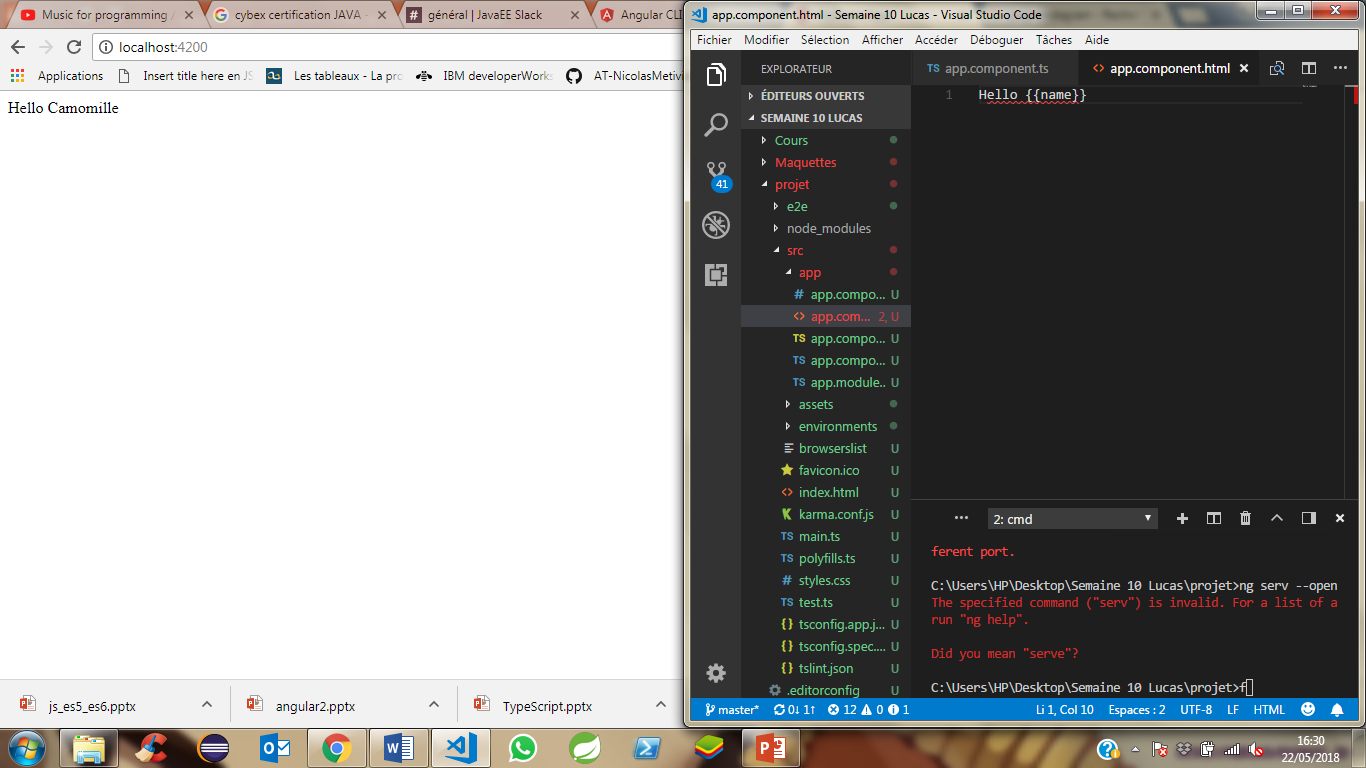


Composant : Brique de notre fichier HTML composant une partie de notre fichier HTML.

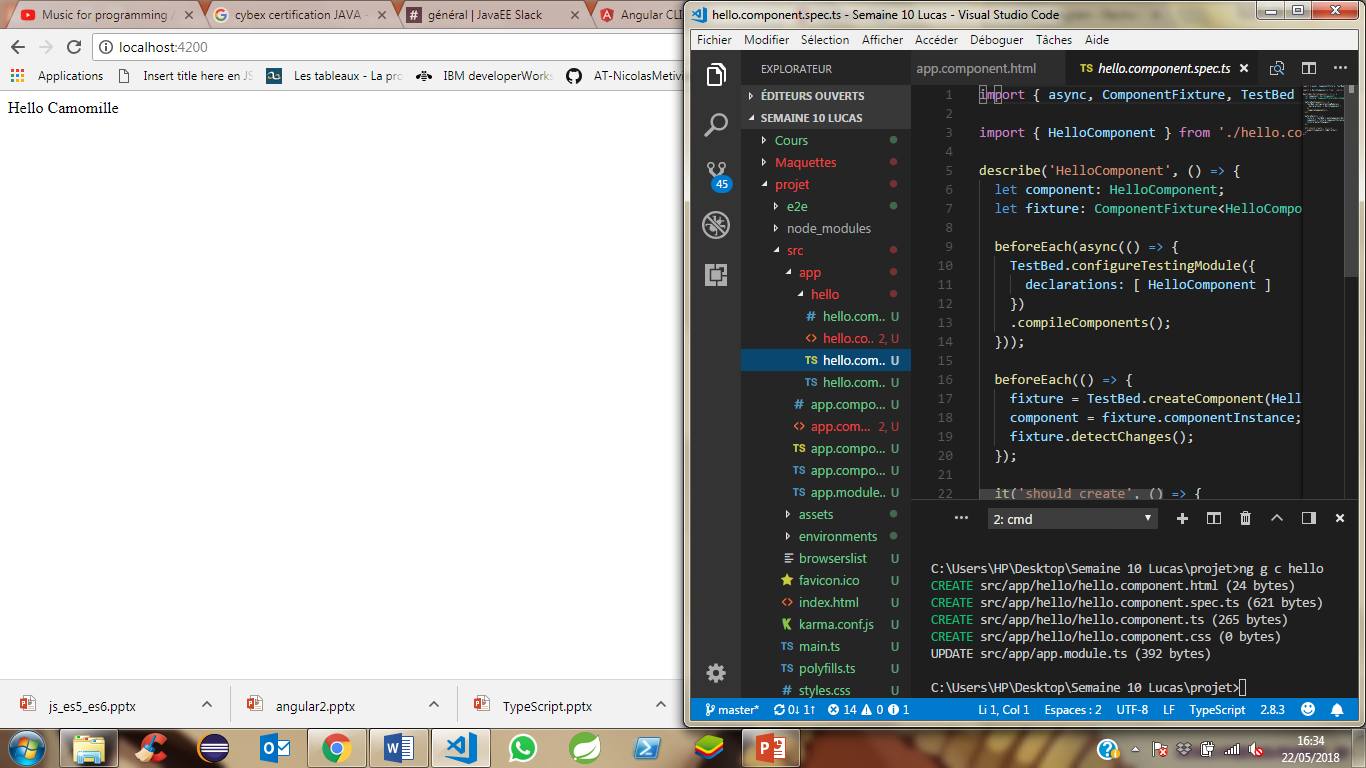


Ce composant contient plusieurs trucs : Un fichier de style (css) qui ne concerne que ce composant. Le fichier HTML qui est un template qui va permetrer de sortir des variables.

Grace à ca on peut fairer apparaitre des variables directement dans le code HTML (grace aux moustaches (double accolade)) :



**Pour créer de nouveau composants, on ecrit dans la vitre de commande >ng g c hello qui s’appelle hello. Cree les fichiers pour le composant (dossier contenant un fichier css, un html, un typescript et 2 fichiers qu’on utilisera pas nous).**

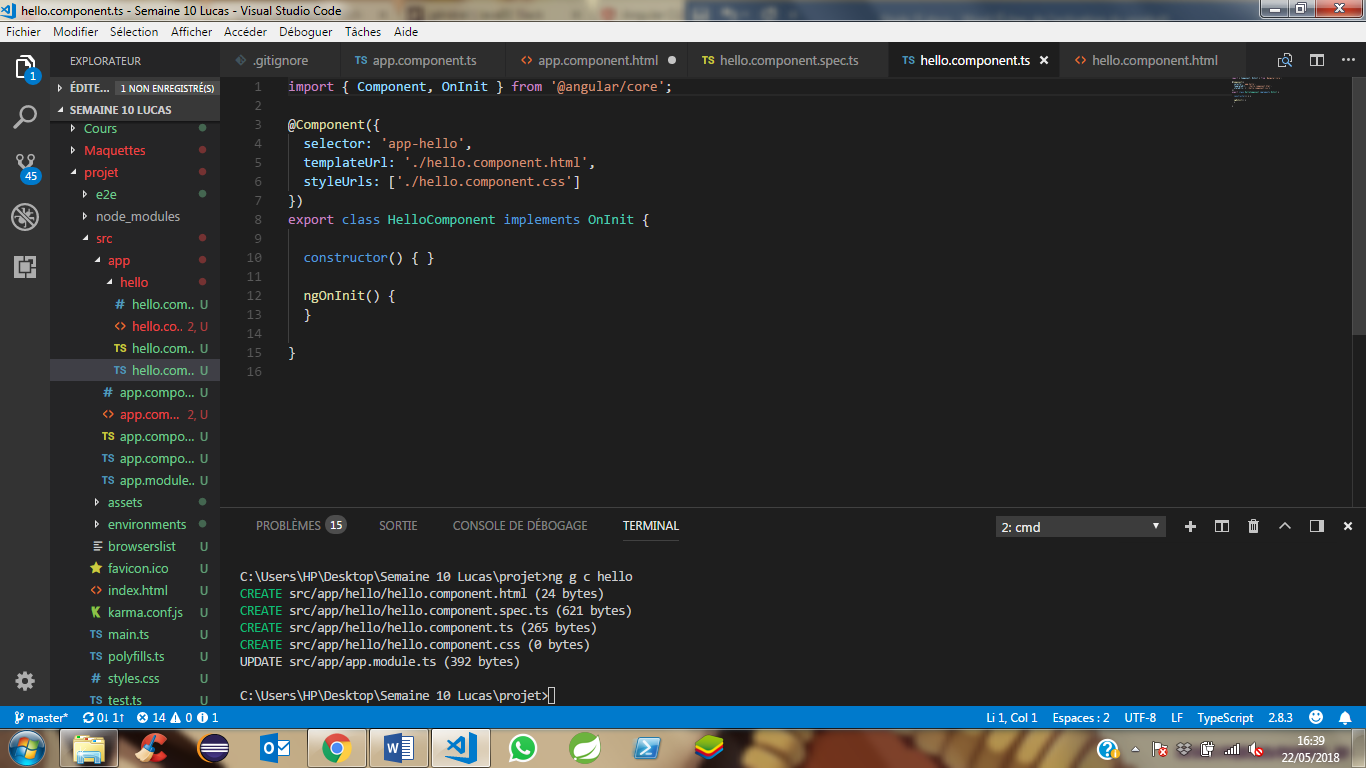


Dans angular, en general on fait des single Pages HTML, on a une seule page qu’on rend ultra dynamique pour qu’elle se modifie.

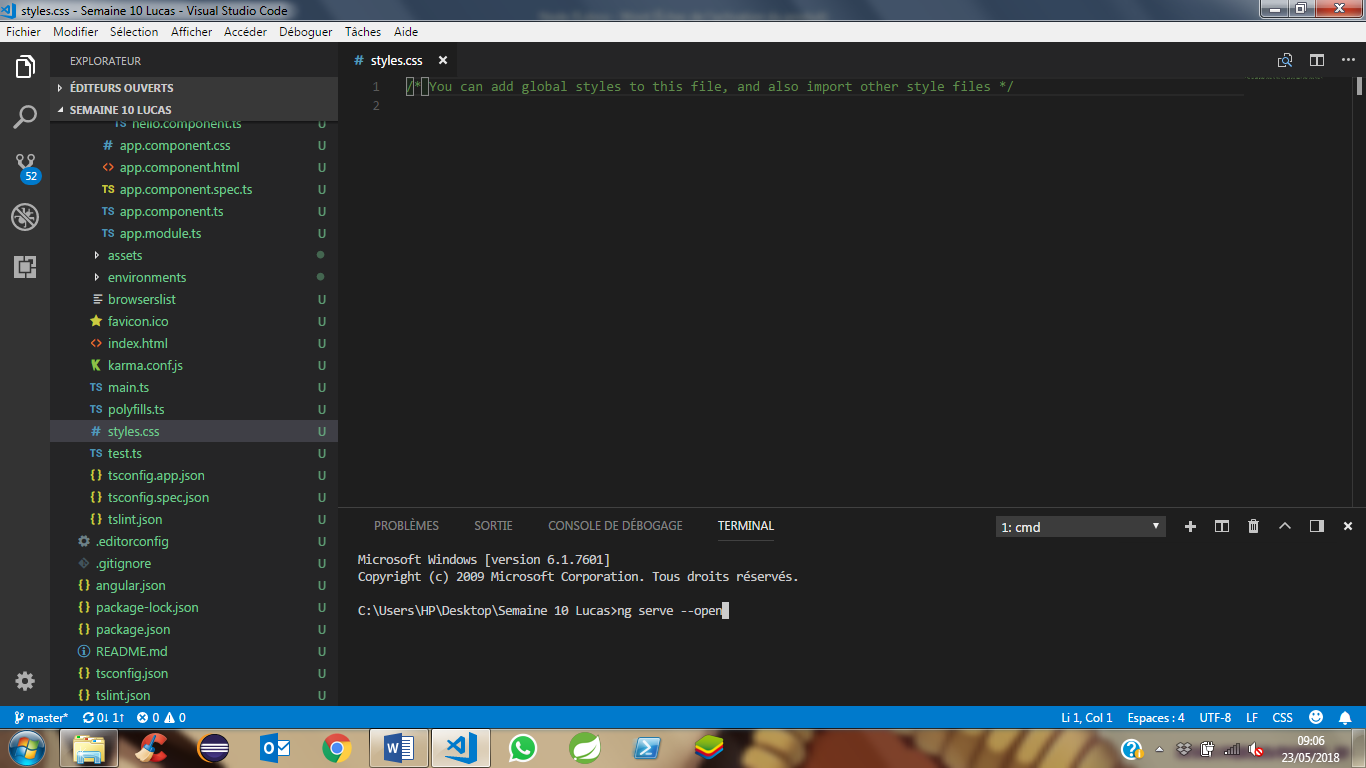
Pour utiliser un composant, il faut utiliser son **Selecteur** (en haut de la page du .ts) :

C’est là dedans qu’on déclare nos variables, avec des lignes de type :

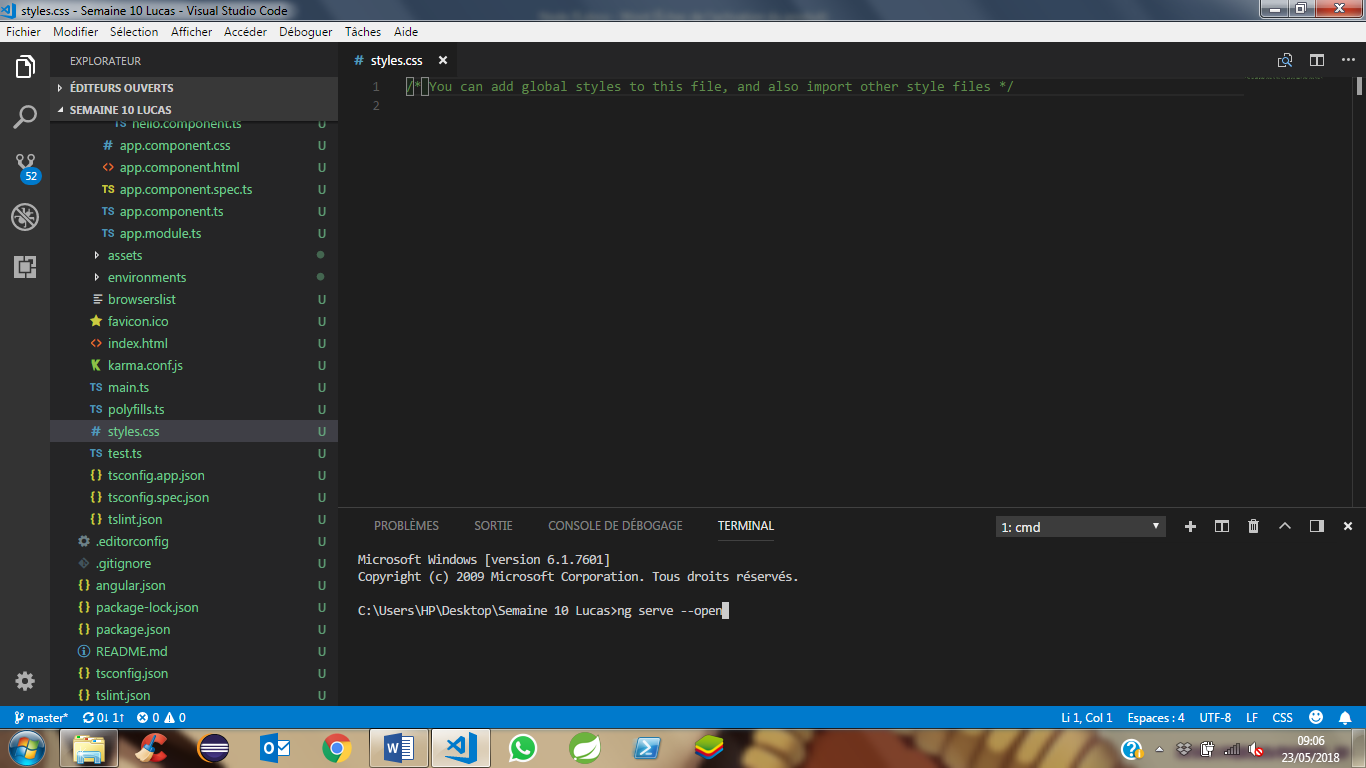
etudiant={'nom':'Bonvin', 'age':26};



Pour appeler et écrire ce qui se situe dans un composant on le met entre balises son Sélecteur dans le HTML principal



**POUR LANCER LE SERVEUR (se trouver dans le dossier workspace) :**



**(1)Data bynding :** La variable est dans le typescript, aura un impact sur le html mais du HTML on ne peiut pas impacter le timescript (one way data bynding).

Trois types de data byngind, de ts vers html, de html vers css et de l’un vers l’autre, c’est ça qui fait que c’est BIEN plus rapide qu’en java script ! Car ici la variable est automatiquement transferee vers le HTML, en java il faudrait tout le temps faire getBy…(…).

Créer un retour de methode :

getEtudiant():string {

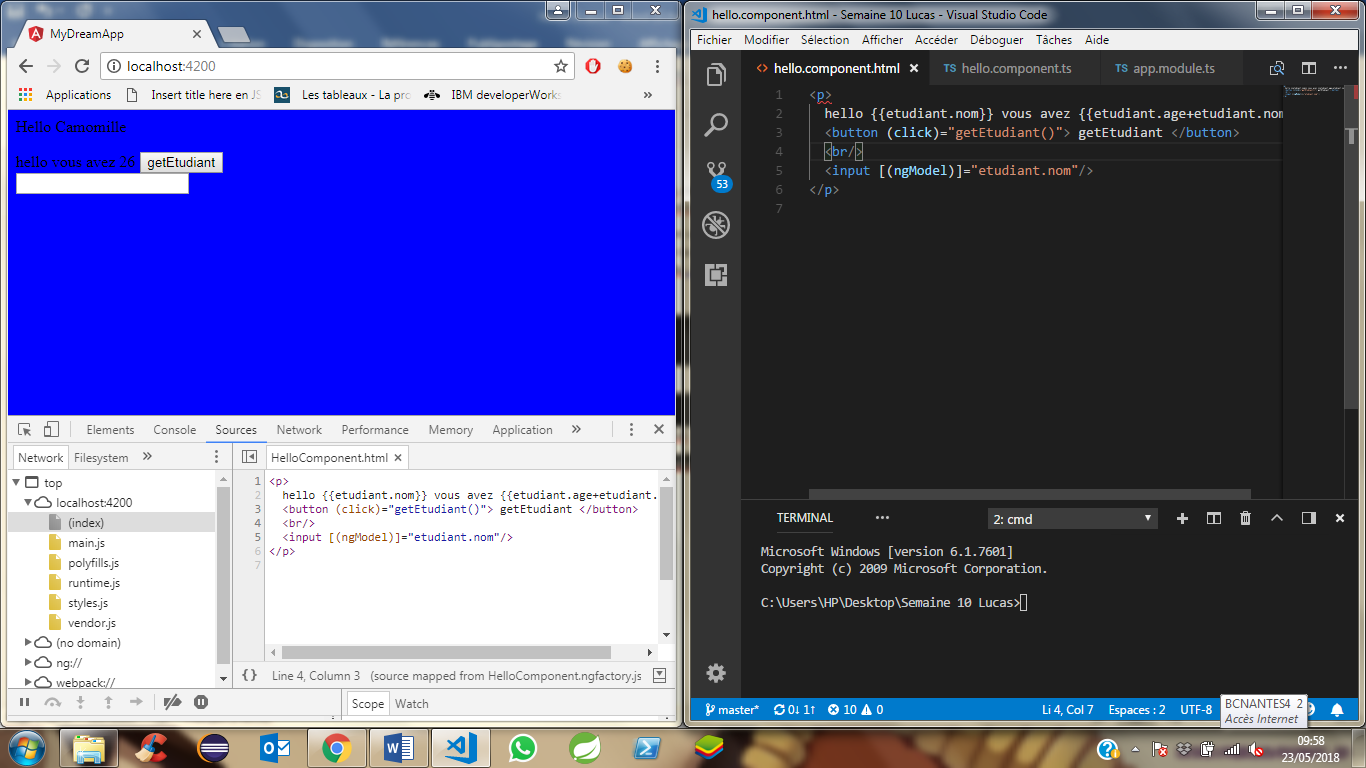
return ‘hello’ ;}

Créer une methode sans retour :

getEtudiant() {}

(2)Un exemple de databynding HTML vers Timescript :

<button (click)="getEtudiant()"> getEtudiant </button>

(3) Data bynding qui va dans les 2 sens: 

On repete, mais l’aventage par rapport à JS c’est qu’on peut modifier très facilement les valeurs avec le binding !

Connaitre les définitions de : Components(app.modules.ts)/Modules/Pipes/Directives

MAJ+ALT+F : Remettre en forme.

**IMPORTANT :**

Pour faire des fonctions JAVA et JS dans des balises, on ecrit \*ng…

<div \*ngIf="etudiant.saved">Etudiant enregistré</div>

<button (click)="saveEtudiant()">Enregistrer Etudiant</button>

<ul \*ngFor="let e of etudiants"></ul>

**Ce que ça fait c’est que ça fait une boucle en pregnant tout les elements de la liste, et les trie en function d’un Boolean, soit ils sont true soit ils sont false. Ce boolean est basé sur la fonction qui est mise en paramètre dans le « filter ». Cette fonction en parametre de filter est basée sur un paramètre, si ce paramètre ets present c’est true, sinon c’est false. On peut faire ça car comme on le sait en JAVASCRIPT une fonction peut être stockée dans une variable, permettant de mettre une fonction en paramètre. Les LAmbdas permettent de déclarer rapidement une fonction, en dessous le lambda c’est** a => a.id === id. Reviens a faire function (m){return m.id ===id}

On peut meme le filtrer avec plusieurs conditions : a => a.id === id && a.truc === machin ;

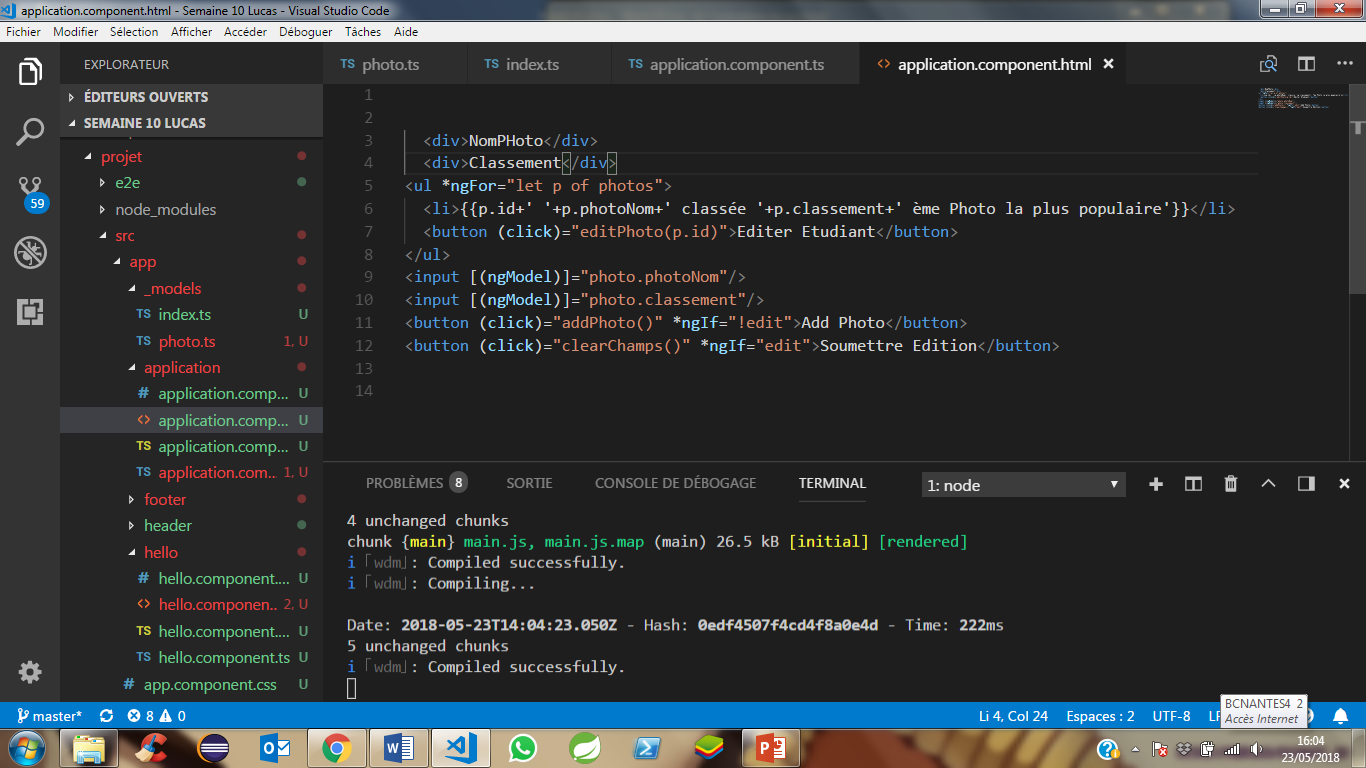
Abstraction : ON prend quelquechose de la vie reel et on le met sous forme de code (ex: instance d’animal)

**Façon la plus facile et rapide de faire getByID :**

return this.photos.filter(a => a.id === id)[0];

**Les Lambda : Ce qui se trouve dans la parenthère de la methode facile du getById**

**Les documents code de ce cours ont étés :**



**Classe HTML :** 