

- 1) Définir un framework
- 2) Créer q — Angular SPA
- 3) router-outlet
- 4) app-routing.module.ts
- 5) un Component
- 6) du Binding
- 7) les types des formulaires + les méthodes techniques de chacun
- 8) les types des formulaires

9) C'est q- app. module ots

10) C'est q- la différence entre

[] et { }

11) C'est q- *ngfor

12) C'est q- e classe.

1) Node JS : Env d'exécution le code JS
en dehors du browser

2) NPM : Node Package Manager

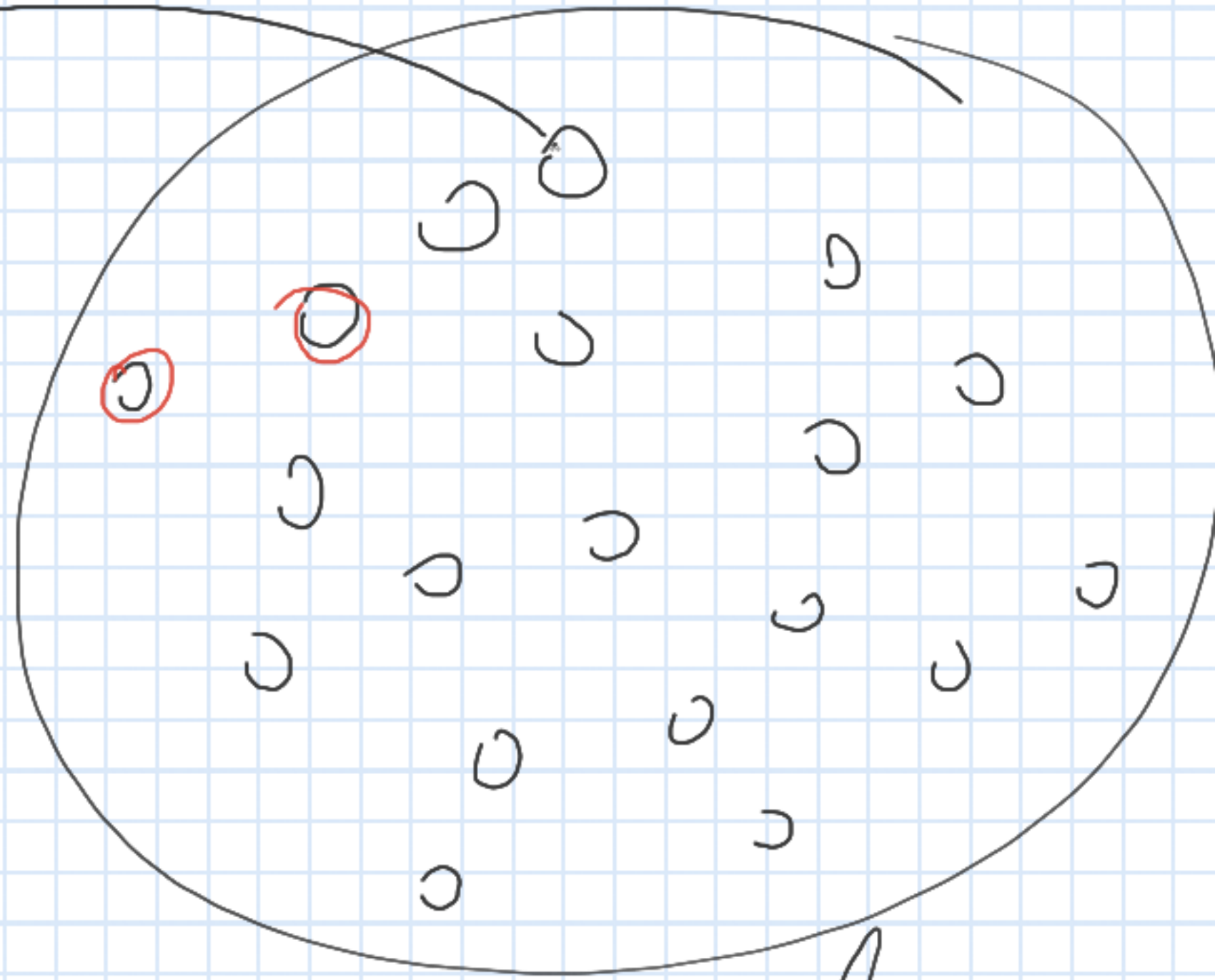
3) Git → Git Bash

4) npm i -g @angular/cli@Version
module

Angular

NPM

gestionnaire
de
modules
packages



install

uninstall

update

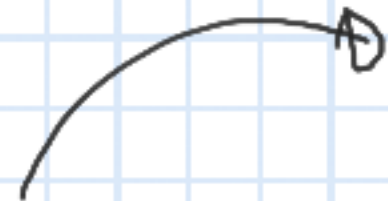
modules

2009

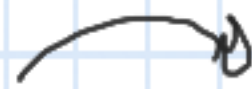
windows :

95

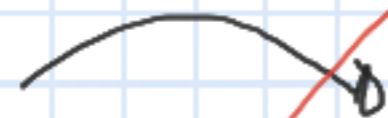
XP



7



Vista



8 x

MS DOS

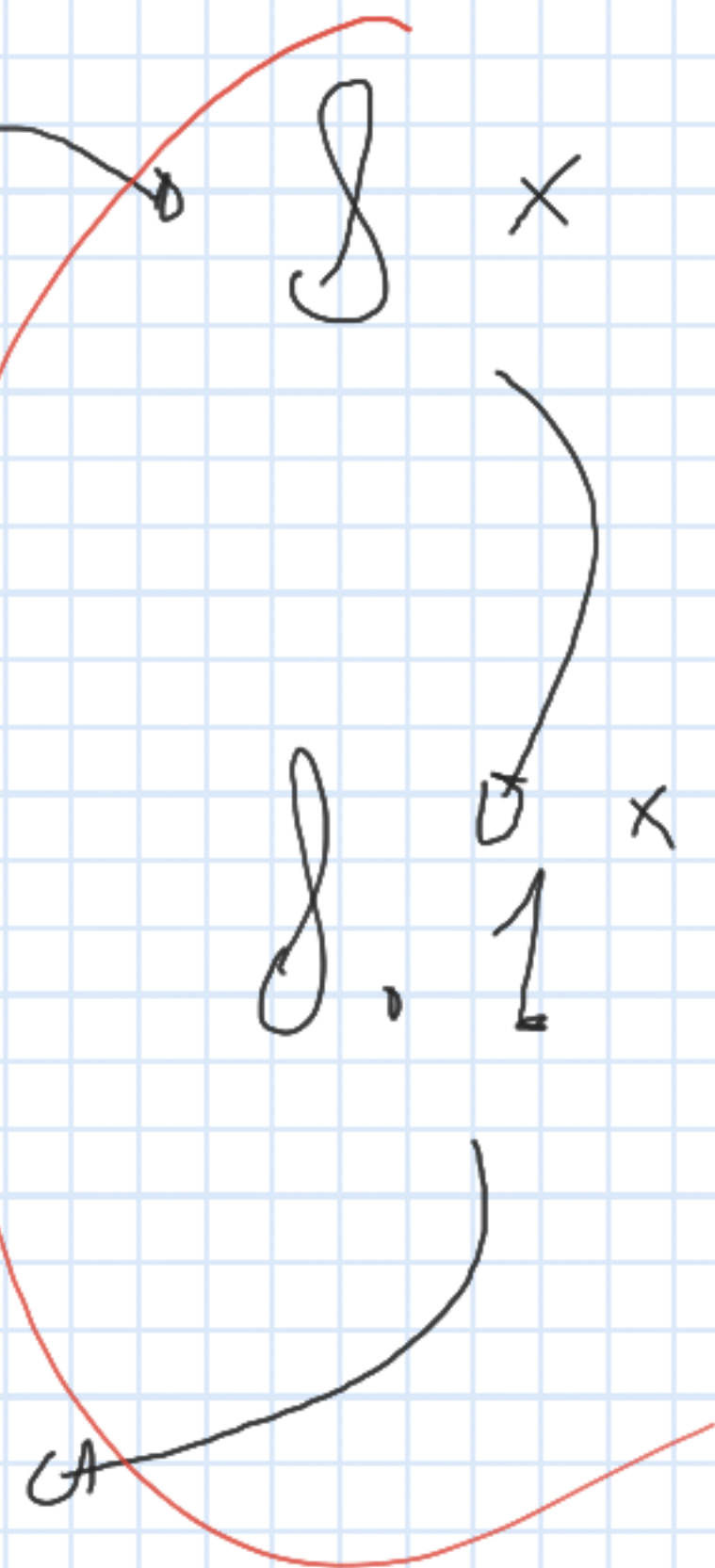
8.1 x

11

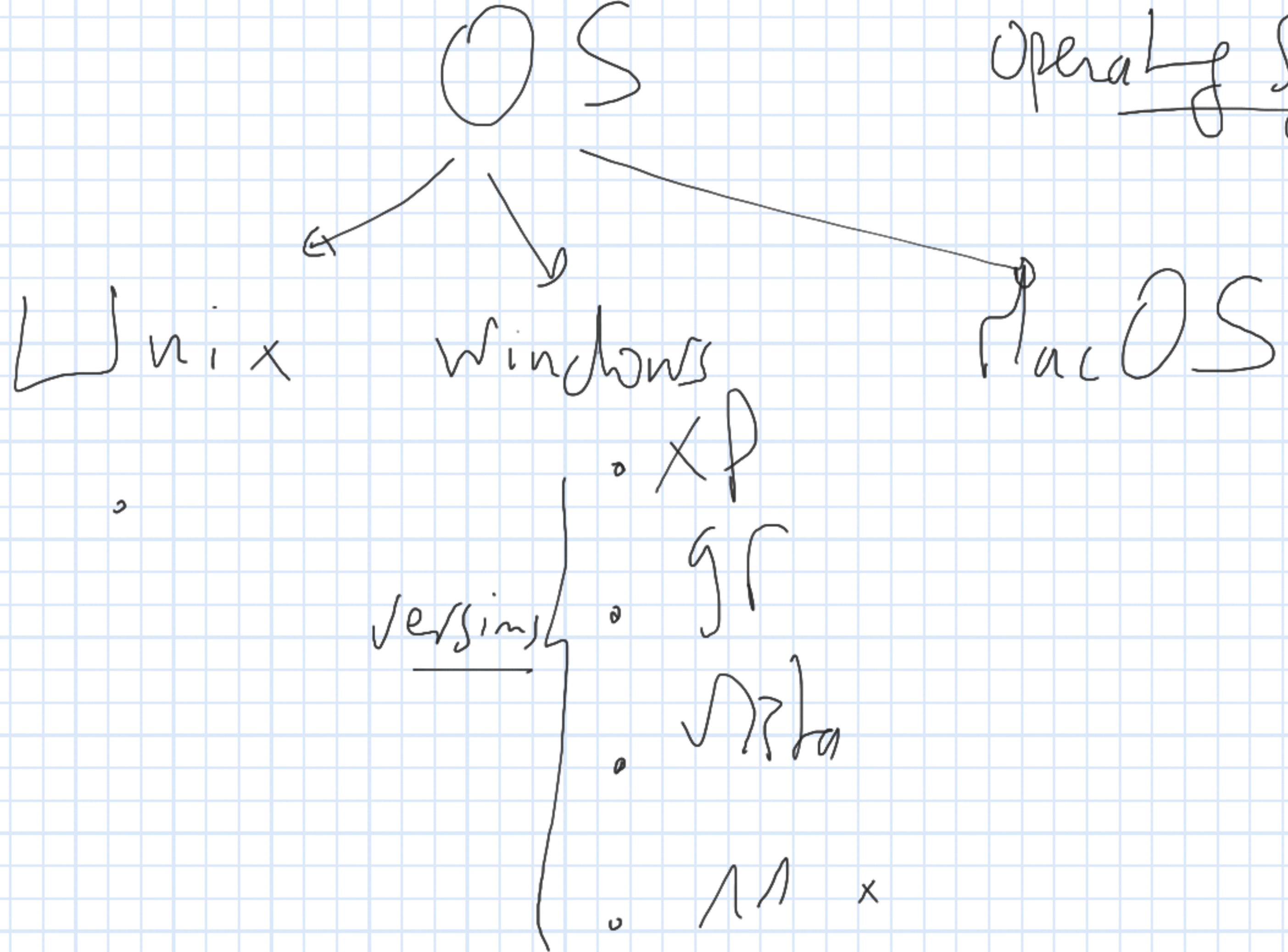
10



9



Operating Systems



Git Bash

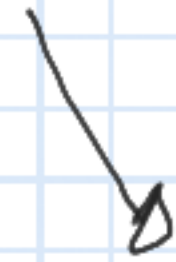
:

terminal/
invite da Cmd
Cmd
powershell
CONSOLE

\$ @commande
↓
prompt

\$ new No-Projet
>

only ?



la navigat' entre les

> stylesheet ?
pages
CSS
Cascading Style Sheet

\$ n f serve

\$ n f s

\$ n f s - O

open browser

\$ n f s - O -- port = 4205

http: / local host: 4200
Hyper Text
Transfer
Protocol
server
local (default)
Unique

index.html

(SPA)

Single Page Application

tout le projet s'exécute

de la seule page web index.html

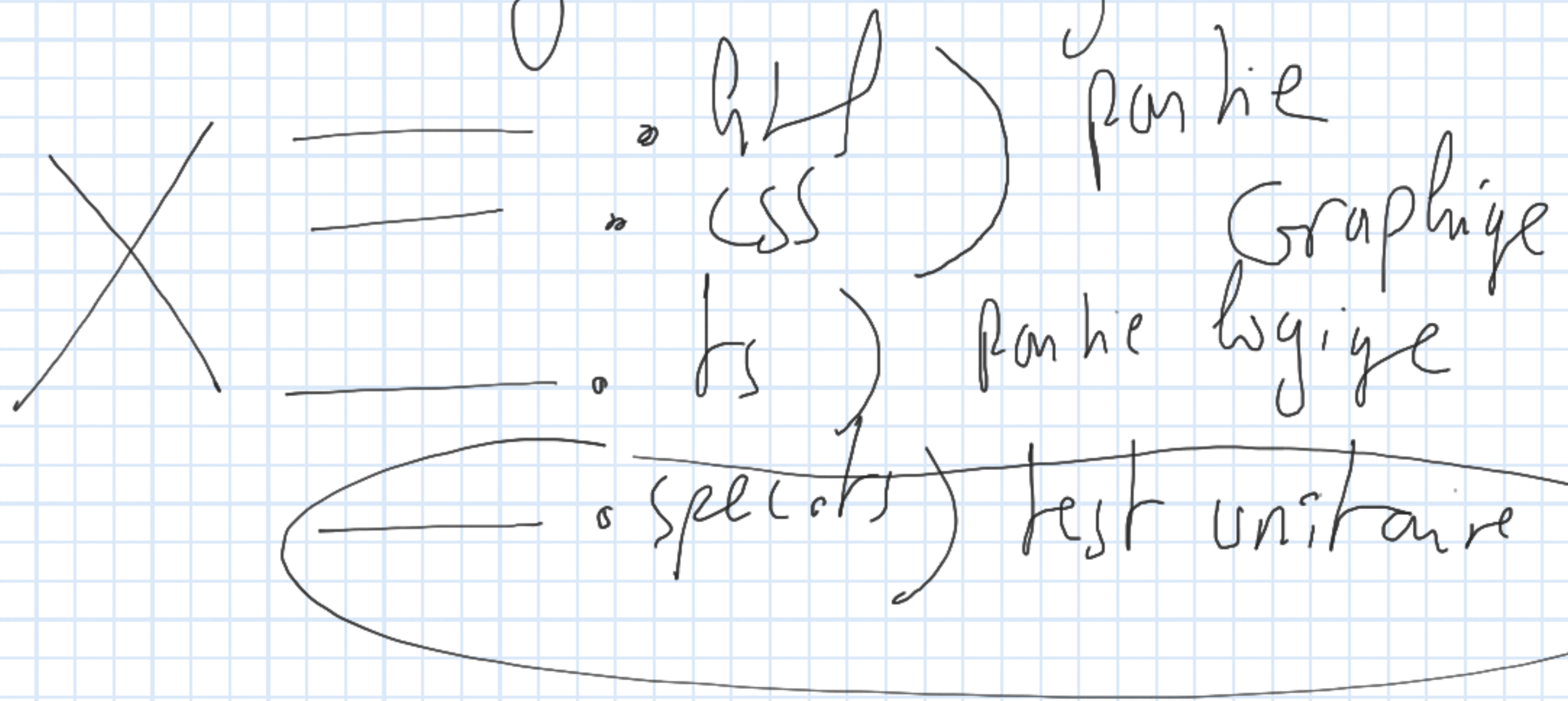
index.html



<app-root>

Selection
du composant
principal

Component : un bloc d'—page web
réutilisable ayant 4 files :



- selector : pr faire Appl au
<app - ...

- templateUrl : fichier HTML

- styleUrls : fichier CSS

Comp : une classe ayant une
partie graphique.

export class A {

pour
importer
le Comp

}

Class PC : modélisat° d'un objet
de la vie réelle ayant des attributs et des
méthodes. (fonct°)

Name : Asus
RAM : 16 RAM
Disque : 1 To SSD
Color : Blanc

obj 1

HP
4 RAM
512 GB SSD
Blanc

obj 2

Lenovo
8 RAM
512 GB SSD
Noir

obj 3

Class Uses

8 MD

objects

U
I

U
I

U
I

U
I

U
I

Sex

fn
fn
Add
Age

height
weight
ell
llwa

TC

export class A {

✓ x: number = 10;

✓ y: boolean = true;

✓ z: string;

✓ a;

b = "hello";

}

export class C {

attributs

a : number;
b : number;

Les méthodes

add (a, b) {

return a + b + this.b;

mult (x, y) {

return x * y + this.a;

export {
 class X {
 a & b; = 10
 A (a, b) { return a + b; }
 B () { return this.A (5, this.a) + 9; }
 }
 }

Appl a →
 method

}

Data Binding : liaison entre le TS et le HTML

\longleftrightarrow
[()]

login

html
event Binding

style interpolat

login () { }

X : number = 10;

path = "assets/images/a.png";

{ X }

property