ngOnInit() {

// méthode qui s'exécute automatiquement lors de l'appel du composant

corps

}

x : number = 10;

y : string = 'hello';

constructor (private router: Router) {

méthode qui crée des variables

corps }

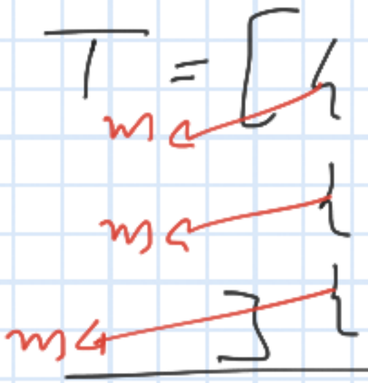
constructor (private fb : FormBuilder)

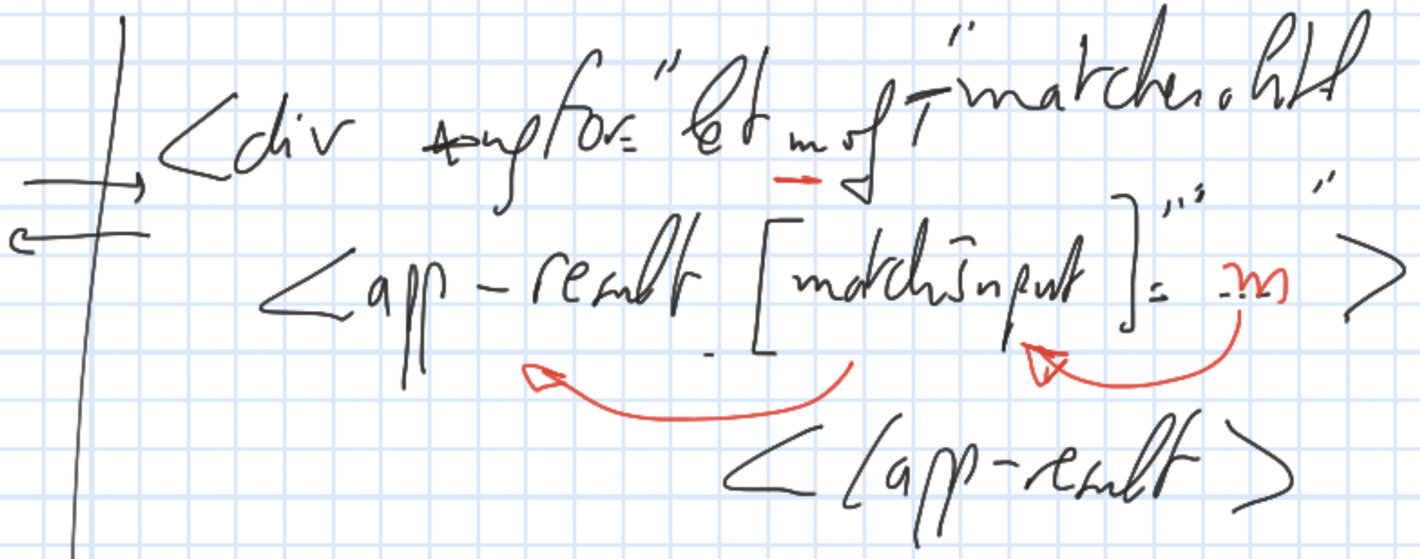
fb : variable
FormBuilder : Type

@Input ( ) : décorateur qui permet le passage des données d'un Parent à un Child

matcher.rs

T = [{ , { , }]

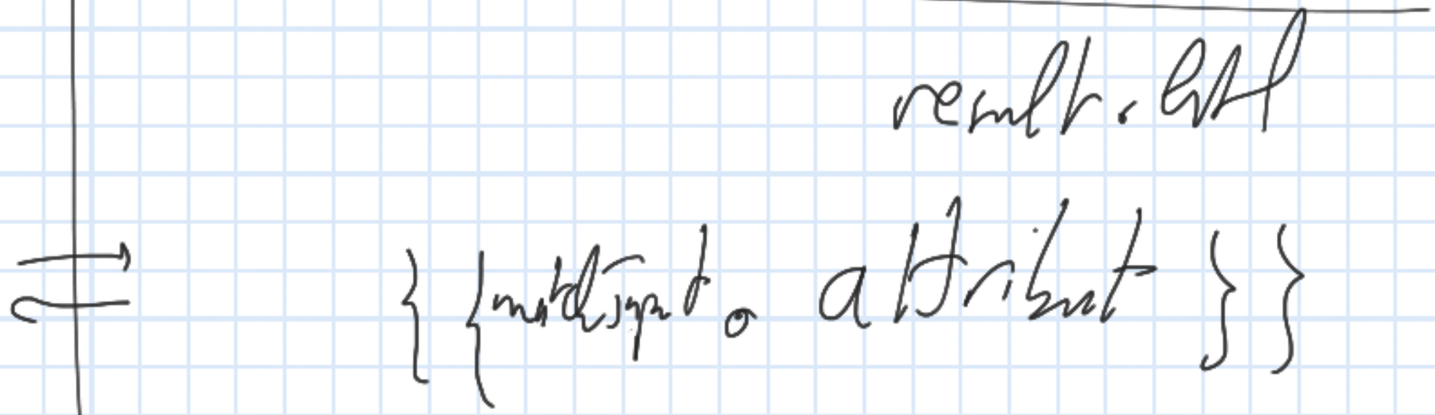




Component Parameter

@Input() matchInput: any;

result.rs



```
f— cte  A( ){

}

f—  B (a,b){
      rel—  a+b;
}
```

```
A( );
A( );

B (½, ½) ;
B ( 1, 3) ;
```

Forms

Template Driven

- ngForm
- ngModel / name
- (ngSubmit)

Reactive form

+ Auto validators (required, patterns)

- formGroup
- formControlName
- (click)

Forms :

app-module.ts

declarations

imports

providers

bootstrap

✓ Components générés

✓ Modules

FormsModule,

ReactiveFormsModule

Services

Component

✓ Principal

AppComponent

Reactive forms :

formBuilder : créateur des inputs

IDform = formBuilder.group ({ fN fLN ei })

Donner un id au formGroup

*myfor

*myIf

⇒ Directives Structurelles

Boucle Graphique

T = [{ }, { }, { }]

elt elt elt

x : bool = true ;

Condit Graphique

<div *myfor = "let elt of T">

<p *myIf = "x">

OK

</p>

↳ Nom variable

href (refresh)

routerLink (sans refresh)

Constructor(
private router : Router)

↑ N_ variable

↓ module de navigat°

Type prédéfini

Path

router.navigate('/')

locat°.replace('')

↓ page HTML

Directives

structurelles

*ngFor , *ngIf

↓

Comportement

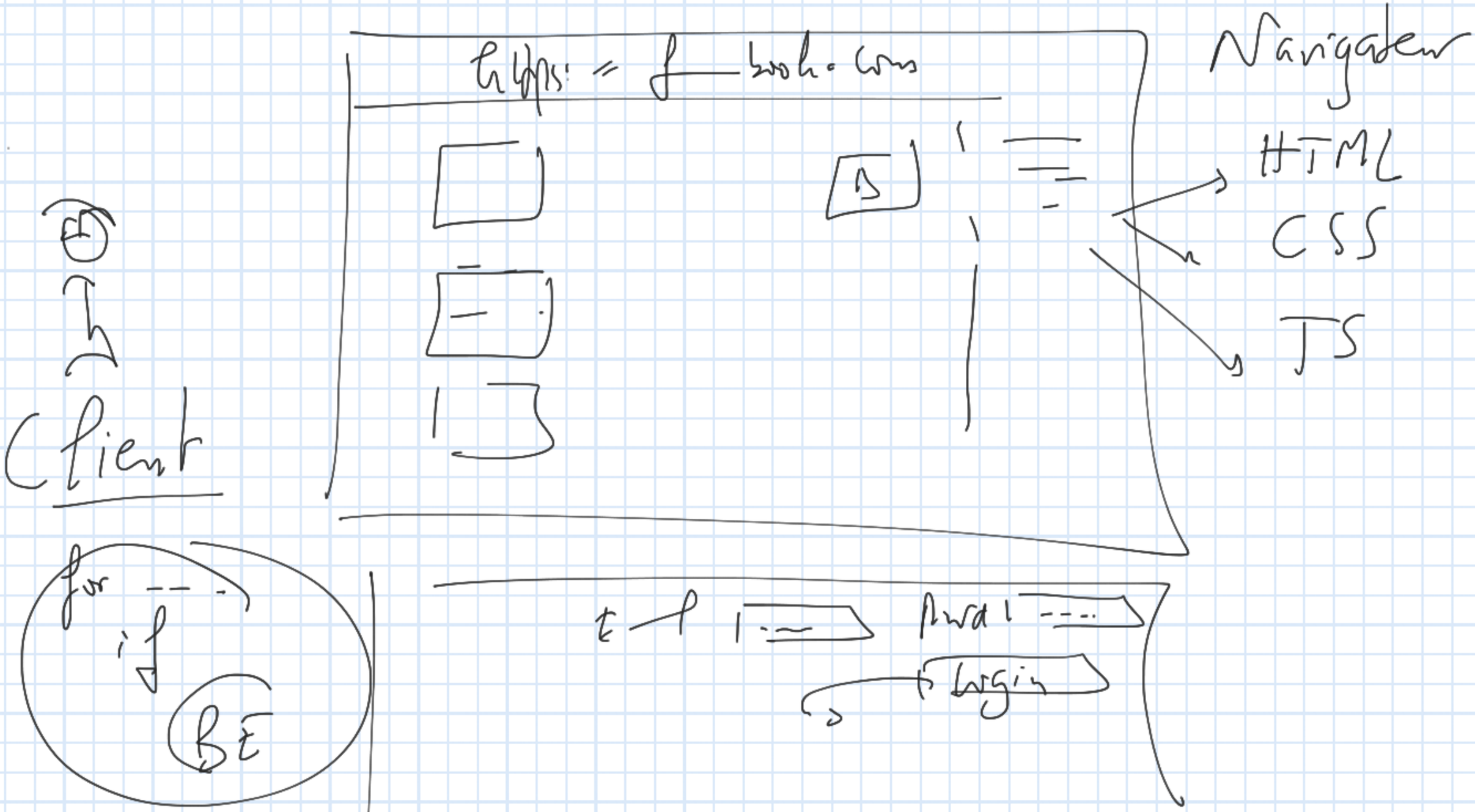
d'attributs ✓

ngStyle, ngClass

↓

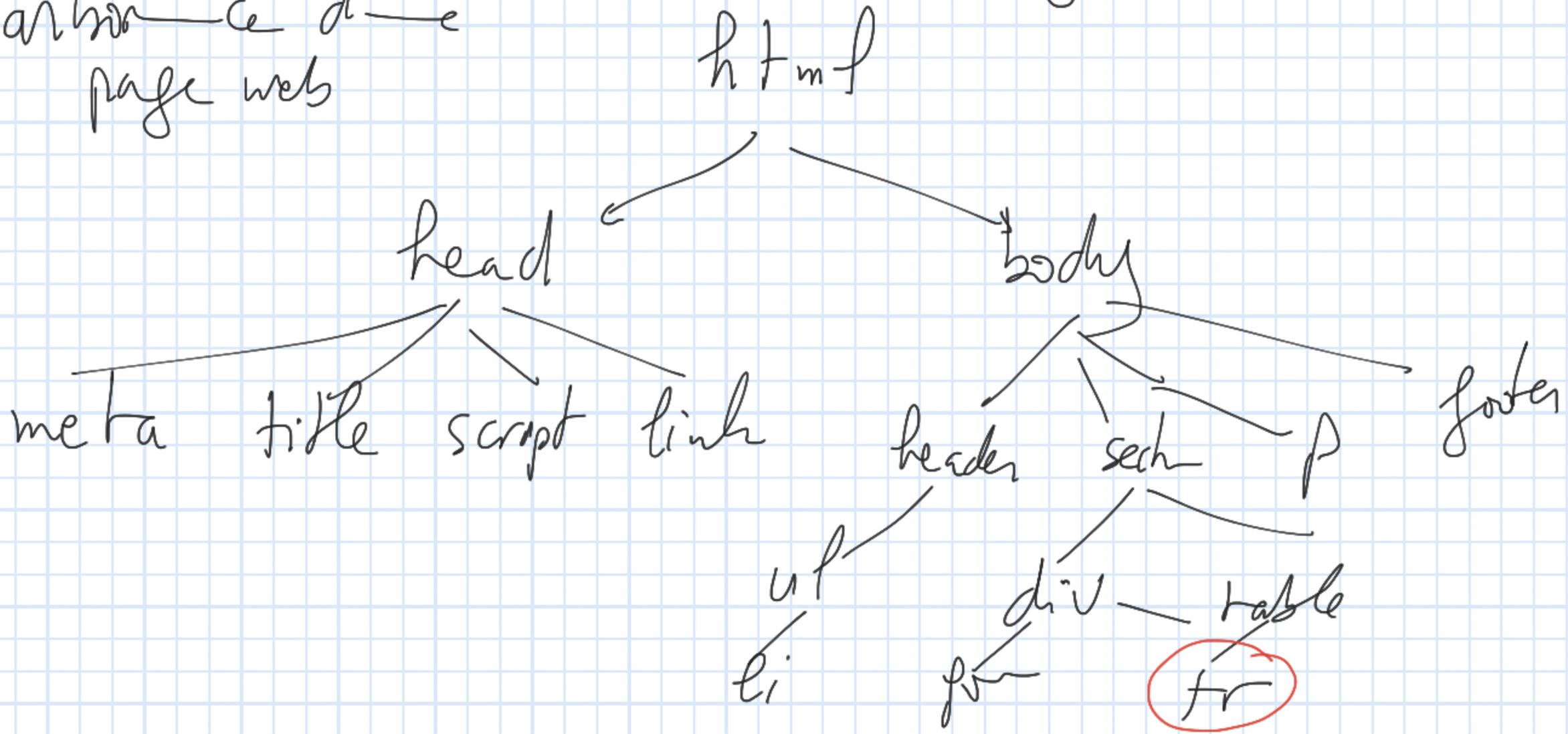
Apparence

des élts du DOM (Salise)


https://book.com
Navigateur
HTML
CSS
JS
Client
for
if
BE
Axios
login

DOM: Document Object Model

arborescence d'une page web



 Contenu

<b0 style="property : value"> Contenu <bf>

---

ng Style : Affichage Dynamique

<p style="color: red;">

Content

<p>

<p class="red">

<p>

.css

.red {
color: red;
}

#green { color:

X ( ) {

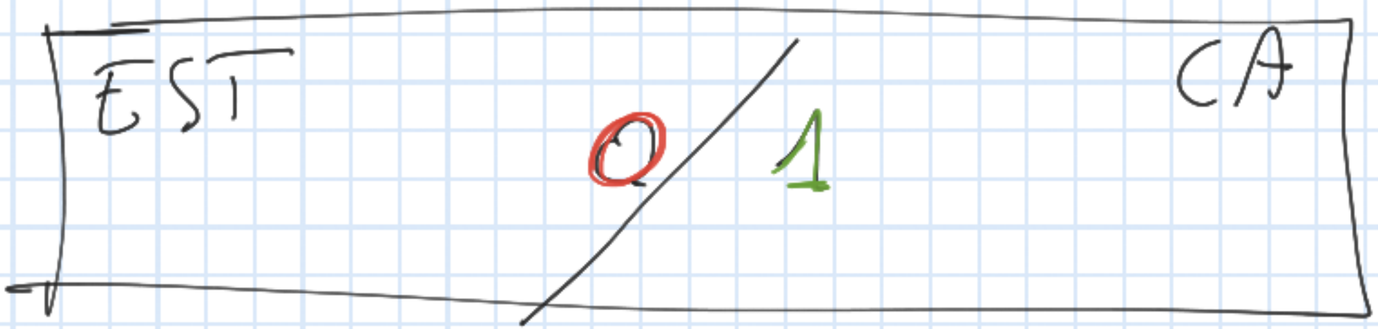
if — " green ",

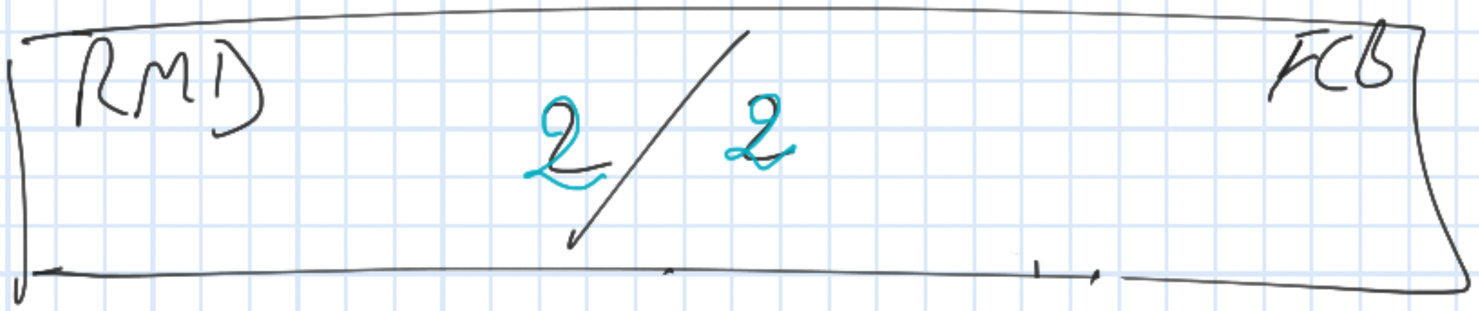
else if

re — " red ";

re — " blue "

}





< BD [myStyle] = " { " property " : value Dynamique } " >

Contenu

< /BF >

```
X(s1, s2) {
  if (s1 > s2) {
    rel = "green";
  } else if (s1 < s2) {
    rel = "red";
  } else { rel = "blue"; }
}
```

```
<p [ngStyle] = "
  { 'color': X(0,4) }">
  Contenu
</p>
```

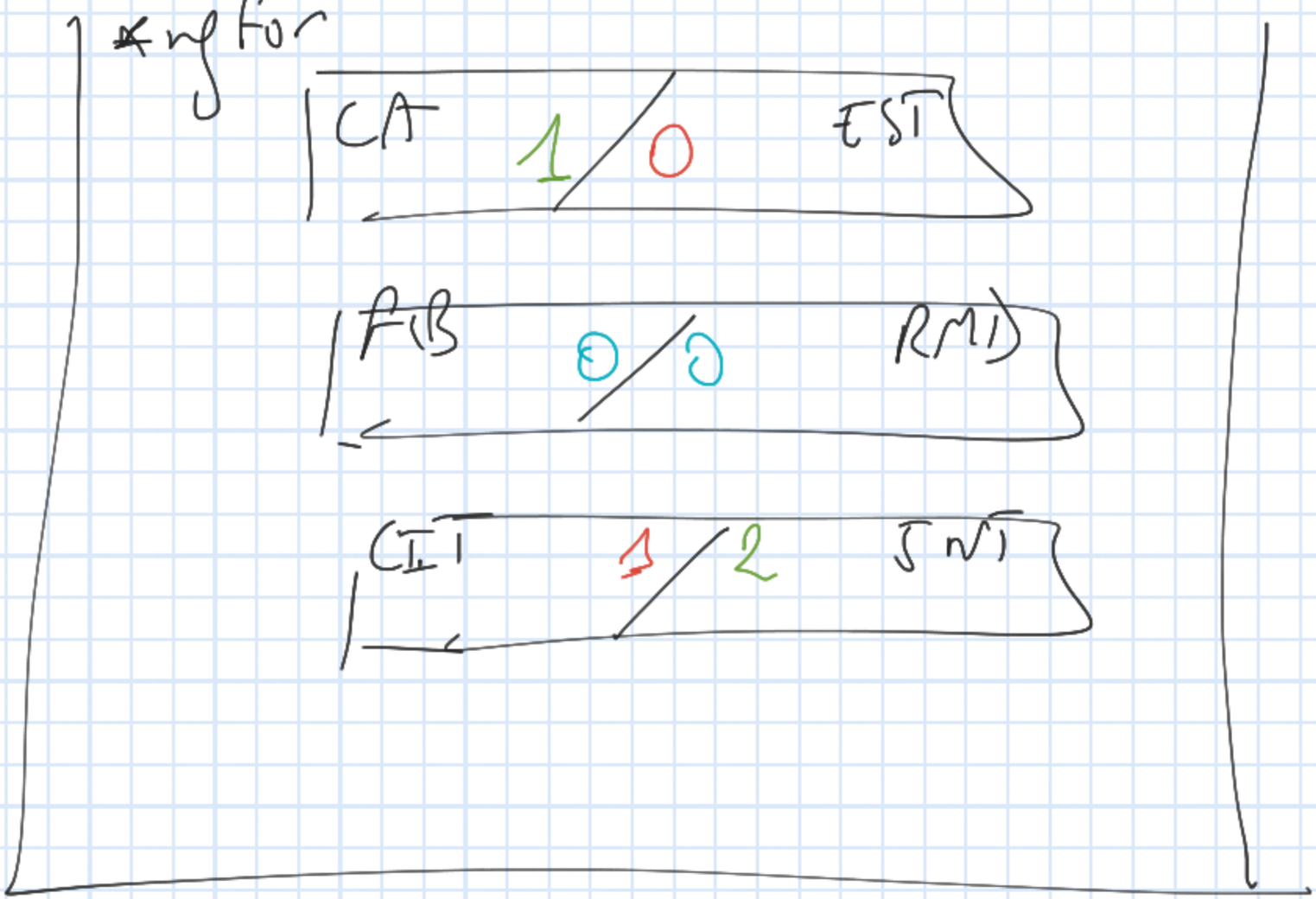
X(1,0)

matches

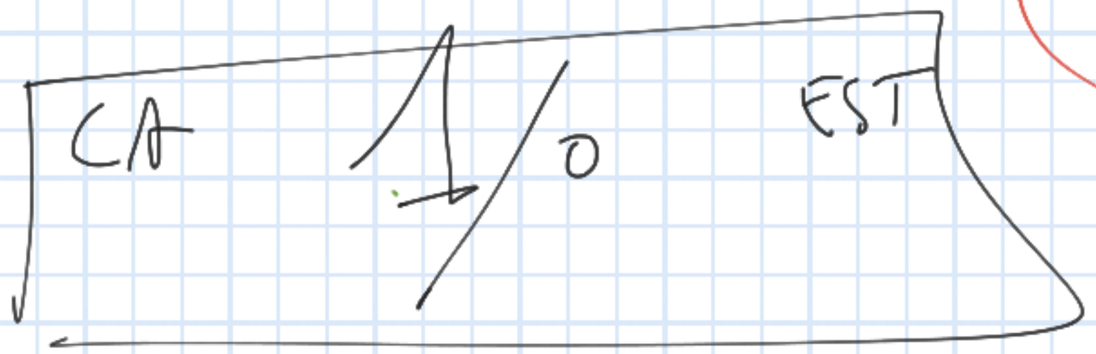
↓

fast fpf

<app-root>



scoreColor(s1, s2, s3) {



font-size: 50px
20px

}

scoreColor(X0...,
X1...,
X2...)

<BO Class = " Ns_Classe " >

Contenu

</BO>

1) Déclarat° des classes CSS

<BO [ng Class] = " { 'Ns_Class1' : Condit 1 ? True / False

'Ns_Class2' : Cond_2 ? True / False

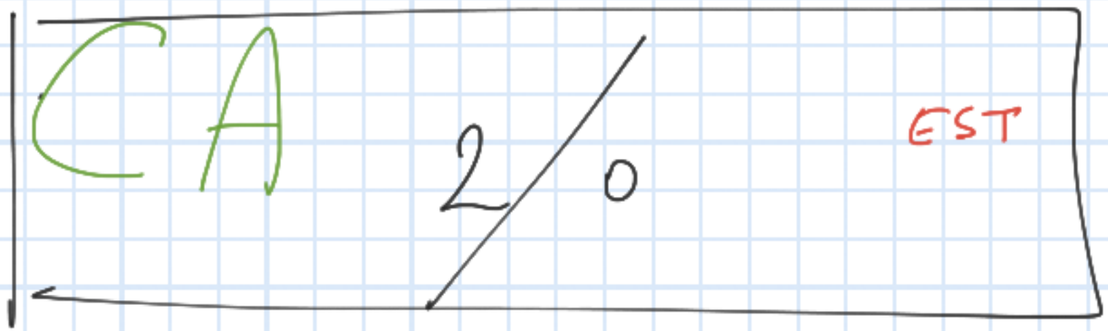
'Ns_ClassN' : Cond_N } " >

Les noms des classes (a, b, c) doivent être déclarées dans le fichier CSS

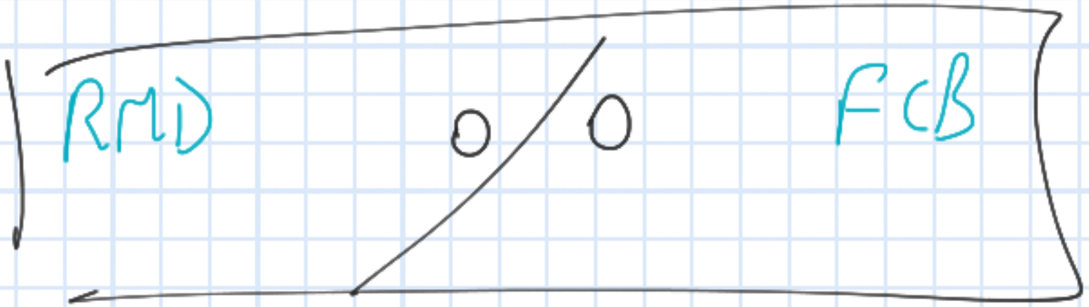
Contenu

</BF>

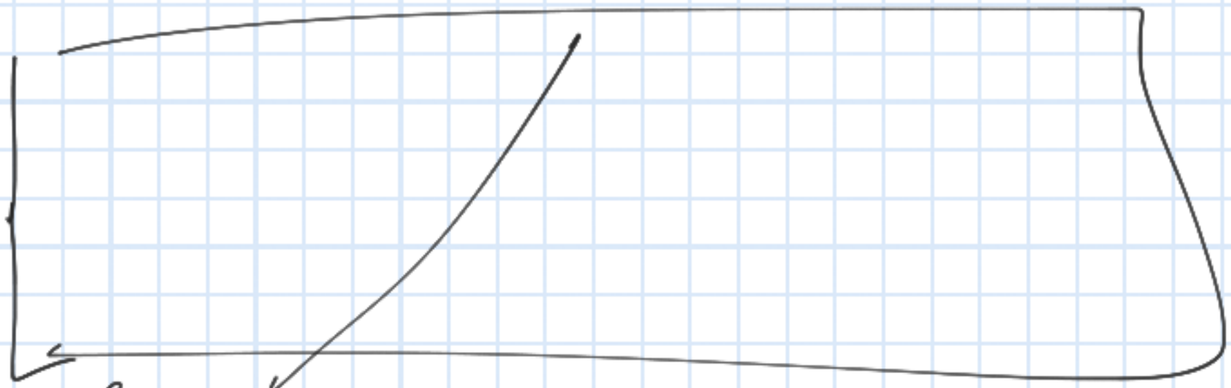
- win {
  color: green;
  font-size: 20px;
}



2) team One features



- loss {
  color: red;
  f-size: 20px
}



- draw { color: blue; f-size: 30px }