Angular:❤️Day1

Framework	Vue	Angular	React
Owner	Evan You	Google	Facebook
Year	2019	2009	2013

Note

ReactNative→Facebook 2015

GraphQl→Facebook 2015 " Server Side

Angular

1)FrontEnd-Framework:

FrontEnd:

- HTMl <Template>
- CSS <Style>
- Logic <JS/TS>
    - ❖ ex:Call Api
    - ❖ Do Function
    - ❖ Anything Js Can Do It
    - ❖ Validation
        - ➔ Format val <Client Side>
        - ➔ Data Val <Server side>

Framework:

اى كلاس ناخده برا البروجكت ونعملة export وعشان استخدمة بعملة import يبقى اسمة Module

شوية Modules مع بعض اسمها Library

شويه Library مع بعض اسمها Framework

Note

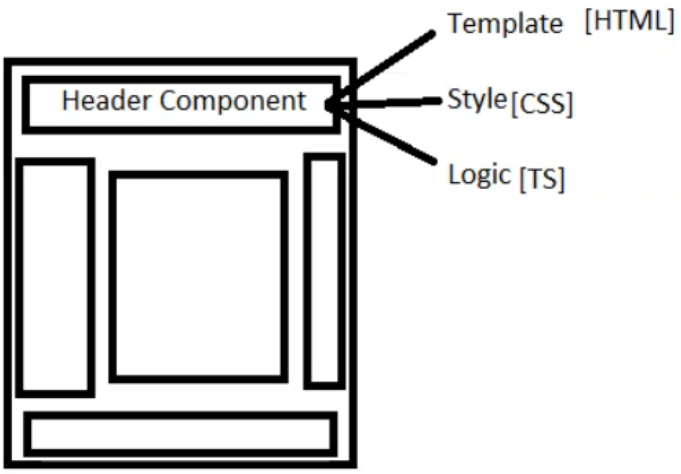
- React contains one library
- Angular based on Node js

2)Component Based

Each component contains 3 files (html,css,js)

Adv:

1. Reusability
2. Maintenance
3. Testing
4. SPA (Single Page Application)



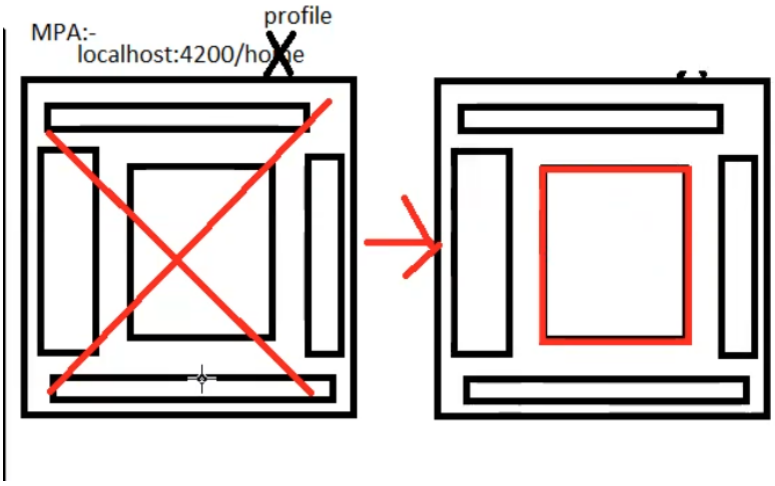
MPA→ Multi Page Application

الفكرة هنا ان مثلا لو كنت طالب صفحة ال home في ال url بعدين عايز اروح على صفحة ال profile في ال MPA بيعمل Reload for whole page رغم ان كان ممكن يغير بس حتة ال home by profile ويسيب ال header w footer زى ما هما

Dis Adv:

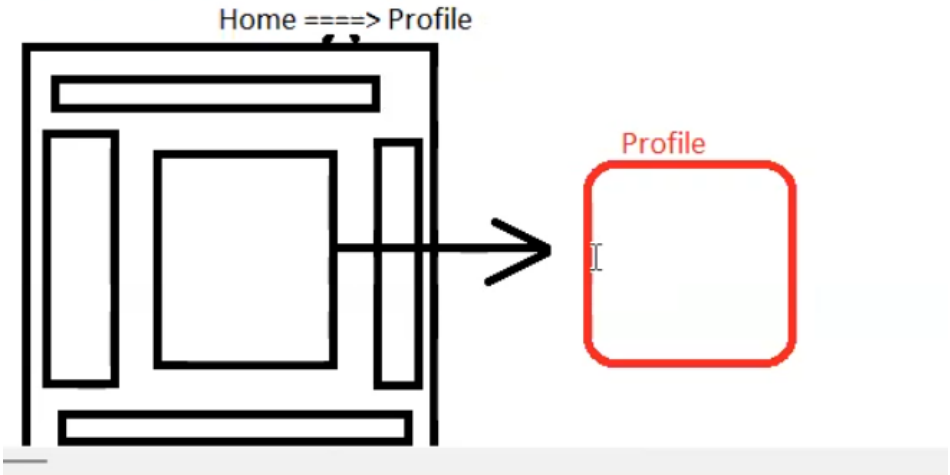
More internet

Slow while reloading



SPA→ SinglePage Application

هنا العكس بقا هيغير بس حتة ال Profil by home من غير ما يحصل reload



Angular Docs→https://angular.io/docs

Steps To Start Your Project:

1. Install Node Js, Npm
2. Install TypeScript
    - npm i typescript -g (Tsc→type script compiler)
3. Install Angular CLI
    - ❖ npm i @angular/cli -g (ng"اختصار كلمة angular")
4. To Create new project
    - Ng new projectname
5. To Run Project
    - ng serve (by default not open in the browser)
    - ng serve -o (to open directly in browser)
6. To coles server
    - ctrl+c

To convert to new version for ecma script use this command

(tsc filename.ts - -target es6)

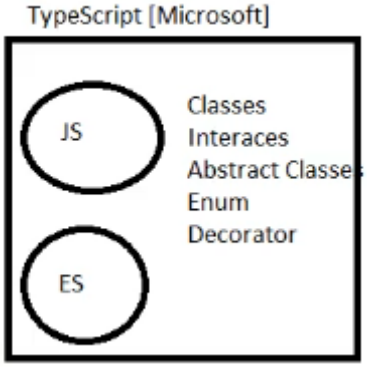
To make js file uptodate with ts file use this command

tsc filename.ts - -target es6 - -watch

Typescript (Microsoft)"إذن هتلاقى جواها كل الكونسبت بتاعة ال oop"

superset from javascript with all updates for ES

+classes/Interfaces/Abstract classes/Enum/Decorator



Extension for typescript files (ts)

```
//to declare var
//datatype =initialization
var x:number=20;
```

Note

In js →loosely coupled "يعنى الفاريابل ممكن بعد كده اغير قيمته عادي وكمان ال datatype"

In ts →tightly coupled "بمجرد الفاريابل ما خد قيمة مينفعش تتغير من رقم لسترينج"

```
//to declare var
//datatype =initialization
var x:number=20;
x=45;//valid
x="Rehab";//error
```

In js →can declare variable without initialization (Console.log=undefined)

In ts → can't do that (Console.log=error)

```
var y:string;
```

```
console.log(y);//error
```

Type Inference "حتى لو محددتش الداتا تايب بس بمجرد ما خد مثلا رقم عمرة ما هيقبل غير أرقام "

```
var c=10; //type inference
c="hdg";//error
```

Ways to deal with variable like javascript

```
//Way 1
var v:any;
v=10;
v="rehab";
v=true;
v={};
v=[];

//Way 2
var b;
b=10;
b="Mahmoud";
b=true;
b={};
b=[];
```

Arrays (Reference type)

```
//take only one number
var arr:[number];
arr=[10];

//take 2 numbers
var arr2:[number,number];
arr2=[1,2];

//take any numbers
var arr3:number[];
arr3=[1,2,3,45];
```

Custom data type (accept only integer and string)

```
var num_string: number|string;
num_string=45;
num_string="Rehab";

var arr4:string|number[];
arr4=[1];
arr4=[1,2];
arr4=[1,33,45];
arr4=[""];//error --> accept array of numbers or string
arr4="rehab";//de tamam
```

To create array of numbers or strings

```
var arr4:string[]|number[];
```

```
var arr4:(string|number)[];
```

Objects(Reference type)

```
var obj:{};
obj={};
obj={name:"Rehab",age:56};
```

Custom object

```
var obj1:{name,age};//just declaration
obj1={name:"cskn",age:45};//initialization
```

Custom data type for property in object

```
var obj2:{name:string,age:number,address:string};
obj2={name:"Rehab",age:45,address:"Helwan"};
```

```
//Can Make optional property
var obj3:{name:string,age:number,address?:string};
obj3={name:"Rehab",age:45};
```

Array of objects

```
var arr_j:{name:string,age:number,address?:string}[];
arr_j=[{name:"Reem",age:20,address:"cairo"},{name:"Rehab",age:45}]
```

Note:

Typescript compiler (TSC)

Engine can't understand typescript so we need something understand typescript then convert it to javascript

→To run ts file

- tsc filename.ts (will create new file with the same name but .js)
- In html file connect with js file not ts

Functions:

```
function xyz():number{
    return 0;
}
//smart enugh to know return type
function abc(){

    return "abc";
}
```

Classes:

```
class Person{
    name:string;
    age:number;
    constructor(name="Person Name",age=45){
        this.name=name;
        this.age=age;
    }

}
var p=new Person();//name=Person Name,age=45
var p1=new Person("rehab",4);//name=rehab,age=4
```

Syntax sugar

```
class Person{
    constructor(public name="person name",public age=0){}
    getName(){
```

```
        return this.name;
    }

}
```

Getter and setter property

```
class Person{
    private address="132 street";
    constructor(public name="person name",public age=0){}
    getName(){
        return this.name;
    }
    //getter property
    //call-->p.Address
    get Address(){
        return this.address;
    }
    set Name(value)
    {
        this.name=value;
    }
}
```

Static variable:

```
    static counter=0;
    constructor
(public name="person name",public age=0){Person.counter++;}
```

Inheritance "extends"

```
class Employee extends Person{

}
```

In typescript can call static member from base class🥲

```
//Default
class Employee extends Person{
    constructor(){
        super();//to chain in parent
    }
}
```

To declare property→set before it any access modifier else it will be parameter

Abstract Vs Interface

```
abstract class Aperson{
    name:string;
    age:number;
    getName(){
        return this.name;
    }
    abstract getAge();

}
class Emp extends Aperson{
    name: string;
    age: number;

    getAge() {
        //throw new Error("Method not implemented.");
    }
}
```

```
interface Iperson{
    name: string;
    age: number;
    getName();
    getAge();
}

class Emppp implements Iperson{
    name: string;
    age: number;
    getName() {
        throw new Error("Method not implemented.");
    }
    getAge() {
        throw new Error("Method not implemented.");
    }

}
```