

# Le développement FrontEnd avec JavaScript

## Module 5 - Les bases du TypeScript



1

Les bases du TypeScript

## Objectifs

- Découvrir les bases du TypeScript



2

## Curriculum vitae

- Depuis 2012
- <https://www.typescriptlang.org>
- TypeScript is a typed superset of JavaScript that compiles to plain JavaScript
- Any browser. Any host. Any OS. Open source



## Pourquoi TypeScript ?

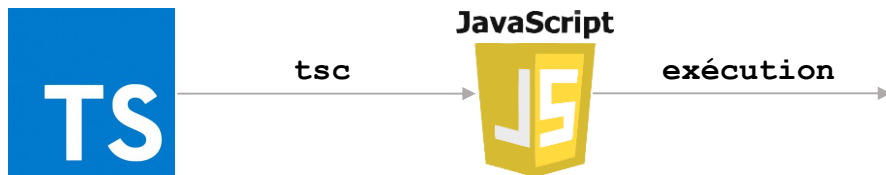


- Typage statique possible
- Utilisation des classes
- Utilisation des interfaces
- Utilisation des génériques
- Utilisation des énumérés
- Utilisation des modules



Les bases du TypeScript

## Exécution



Les bases du TypeScript

# Démonstration



## Les variables

```
//Déclaration de variables en TypeScript
var unBooleen : boolean = true;
var age : number = 25;
var profession : String = "Informaticien";

//La syntaxe JavaScript est toujours possible
var uneVariable = 123;

//Préférez la syntaxe suivante :
var uneAutreVariable : any = "dix";
uneAutreVariable = 10;
```



## Les énumérés

```
//Déclaration d'un énuméré
enum Etablissement{Maternelle, Primaire, College, Lycee, Faculte};

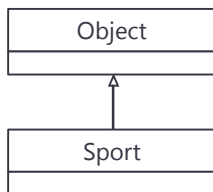
//Utilisation d'un énuméré
var etablisement : Etablissement = Etablissement.College;
```



# Démonstration



## Les classes

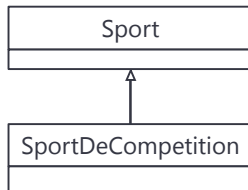


```
//Déclaration d'une classe en TypeScript  
class Sport  
{  
    public nom : string;  
    private prive : string;  
  
    constructor(nom : string, public description : string)  
    {  
        this.nom=nom;  
        this.prive="valeur cachée";  
    }  
  
    public afficher():void  
    {  
        console.log(this.nom+' '+this.description);  
    }  
}
```



## Les bases du TypeScript

# L'héritage



```
//Mise en place de l'héritage en TypeScript
class SportDeCompetition extends Sport
{
    constructor(nom : string,
                description : string,
                public niveau : string)
    {
        super(nom, description);
    }

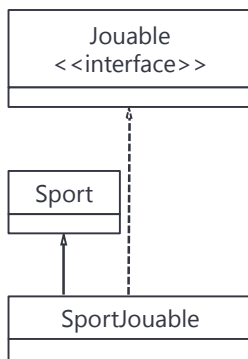
    public afficher():void
    {
        super.afficher();
        console.log("Niveau : " + this.niveau);
    }
}
```



11

## Les bases du TypeScript

# Les interfaces



```
//Interface
interface Jouable
{
    //Propriété optionnelle si ?
    score? : string;
    jouer(joueur1 : string, joueur2 : string) : string;
}

class SportJouable extends Sport
{
    jouer(joueur1 : string, joueur2 : string) : string
    {
        //...
    }
}
```



12

## Les interfaces de méthodes

```
interface jouer{  
    (joueur1 : string, joueur2 : string) : string;  
}  
  
var jouerAuBadminton : jouer = function(joueur1 : string, joueur2 : string)  
{  
    //...  
}
```



## Démonstration



## Les génériques

- Les tableaux

```
var tableauType : number[] = [1,2,3];  
var tableauGenerique : Array<number> = [9,8,7];
```

- Les classes

```
class Sandwich<V extends Viande, L extends Legume>  
{  
    constructor(public viande : V, public legume : L) {}  
  
    public afficher() : void  
    {  
        //...  
    }  
}
```



## Démonstration





## Les bases du TypeScript

# Les modules

- Création d'un module

```
//Création d'un module  
module BusinessObject  
{  
    //...  
    export class Sport{/*...*/}  
    //...  
}
```

- Utilisation d'un module

```
var sportModule : BusinessObject.Sport;  
  
import BO = BusinessObject;  
var sportAliasModule : BO.Sport;
```



## Les bases du TypeScript

# Démonstration



## Les expressions lambdas

```
//Déclaration d'une fonction classique
var donneMoiUnEntier = function()
{
    return Math.floor(Math.random()*10);
}
console.log(donneMoiUnEntier());

//Premier exemple d'expression lambda
var donneMoiUnEntierV2 = ()=>{return Math.floor(Math.random()*10);}
console.log(donneMoiUnEntierV2());

//Second exemple d'expression lambda
var donneMoiUnEntierV3 = ()=>Math.floor(Math.random()*10);
console.log(donneMoiUnEntierV3());

//Exemple d'expression lambda pour l'affichage des éléments d'un tableau
var tableau : Array<number> = [1,5,6,9];
tableau.forEach((value, index, tab) => console.log("tableau[%i]=%i", index, value));
```



## Démonstration



TP



## Conclusion

- Vous connaissez les bases de TypeScript pour commencer sereinement avec Angular

