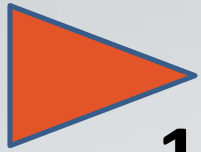




Introduction à TypeScript



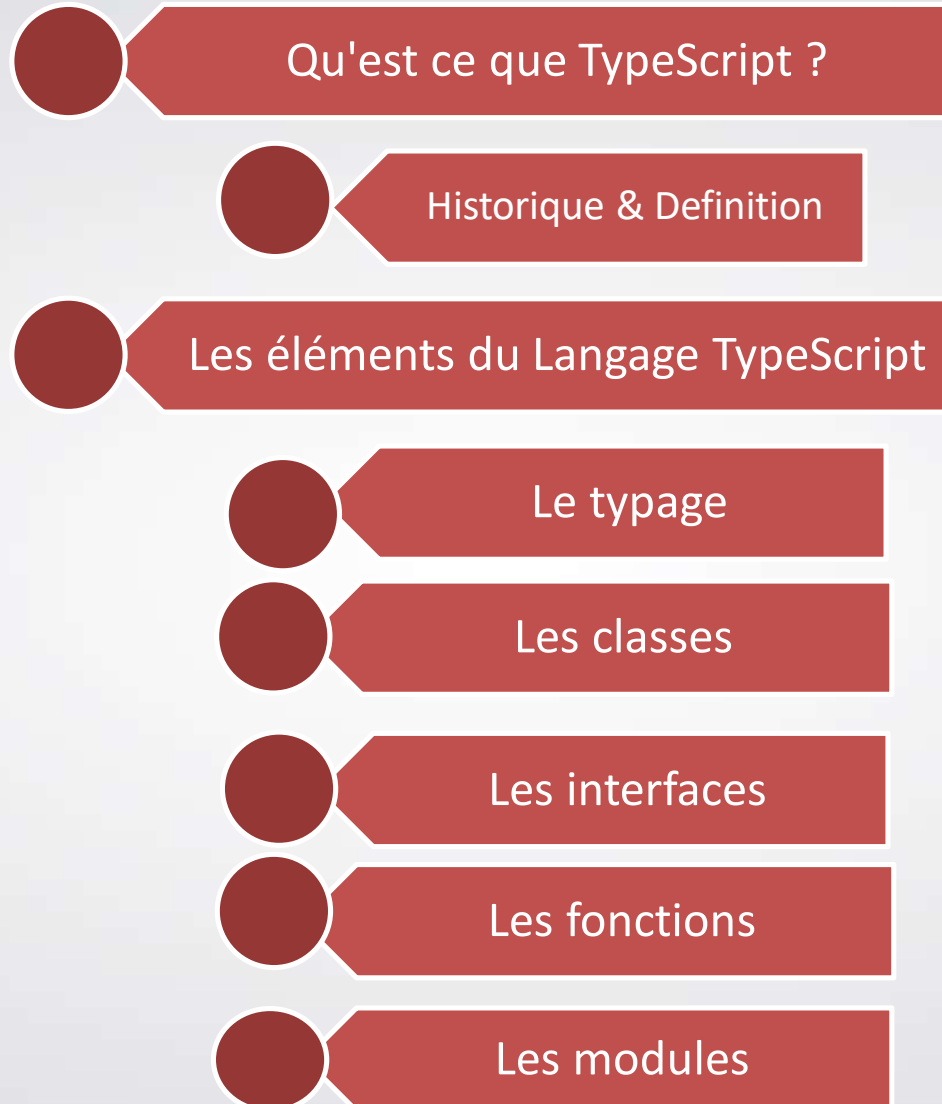
1. Prérequis:

- Connaissances P.O.O , Javascript.

2. Objectifs pédagogiques:

- Maîtriser le langage Typescript
- Appliquer les notions Orienté Objets
- Se familiariser avec les types et la déclaration des variables typescript
- Appliquer TS pour le développement des applications Angular

Plan:





Qu'est ce que TypeScript ?



- Preview depuis Octobre 2012
- Avril 2014 : Release 1.0
- Langage transcompilé en JavaScript.
- Support d'EcmaScript 3 et 5 et ES6.
- Utilisation de polyfills.
- Mis en place et suivi par Microsoft.
- Grande équipe de contributeur .
- Opensource, source disponible sur github.

Les éléments du Langage TypeScript

1. Le typage

1.1 Les types:

Les types de bases sont les même qu'en Javascript:

- number (entier, float, etc...)
- string
- boolean
- any (qui est le type 'objet' JS)

Les éléments du Langage TypeScript

1. Le typage

1.1 Les variables:

- variable de type nombre: `var maVariable :number;`
- variable de type chaine de caractères:
`var maVariable :string;`
- Les variables peuvent être typé implicitement à la déclaration: `var maVariable=1;`

▶ Les éléments du Langage TypeScript

Syntaxe:

Lambda expression

- Fonction 'compacte' qui gère la portée de **this**.

TypeScript

```
var myFunc = (h: number, w: number) => h * w;
```

JavaScript

```
var myFunc = function (h, w) { return h * w; };
```

Les éléments du Langage TypeScript

2. Les classes

- Comme dans javascript ES6 il y a la notion de class dans typescript.

```
class Salutation {  
    saluer: string;  
  
    Constructor(message: string) {  
        this.saluer = message;  
    }  
    greet() {  
        return "Hello, " + this.saluer;  
    }  
}  
  
let Salut = new Salutation("world");
```


Les éléments du Langage TypeScript

2. Les classes (héritage)

```
class Etudiant extends Personne{ // class Etudiant herite de Personne
  classe: string; // attribut

  constructor(name: string, classe: string) { // constructeur
    super(name); // appel constructeur de la class mère
    this.classe= classe;
  }

  initiation(): string { // une methode
    return "Bonjour!"
  }
}

//instanciation
var MonEtudiant = new Etudiant(Mohamed',5GL');

//appel d'un methode
MonEtudiant.initiation();
```

▶ Les éléments du Langage TypeScript

3. Les interfaces

- La nuance avec la class est qu'on ne définit l'intérieur des méthodes:

```
interface IVoiture {  
    moteur: string;  
    couleur: string;  
}  
  
class Voiture implements IVoiture {  
    constructor(public moteur: string,  
                public couleur: string) {  
  
        // ...  
    }  
}
```

Les éléments du Langage TypeScript

4. Les fonctions

Une fonction qui retourne une valeur numérique.

```
function carre(x: number): number {  
    Return x * x ;  
}
```

Si la fonction ne renvoie pas de valeur on lui donne le type **void** :

```
function affiche(texte: string): void {  
    alert(texte);  
}
```

Les éléments du Langage TypeScript

4. Les fonctions

4.1 Paramètre optionnel

TypeScript permet de rendre des paramètres optionnels:

```
function afficheNom(nom: string, prenom?: string): void {  
  let texte = nom;  
  if (prenom) {  
    texte += ' ' + prenom;  
  }  
  alert(texte);  
}  
afficheNom('Durand');  
afficheNom('Dupont', 'Marcel');
```

Les éléments du Langage TypeScript

4. Les fonctions

4.2 Paramètres du reste

Des fois on ne sait pas de combien de paramètres on a besoin. On utilise alors les paramètres du reste:

```
function ajouter(base, ...elements) {  
  for (var i = 0; i < elements.length; i++) {  
    base += elements[i];  
  }  
  return base;  
}  
var resultat = ajouter(10, 1, 2);  
alert(resultat);
```

on ajoute au premier paramètre tous les autres transmis, quel qu'en soit le nombre.

▶ Les éléments du Langage TypeScript

4. Les modules

4.1 les modules internes

```
module Formes {  
  class Rectangle {  
    constructor (height: number, width:  
number) { }  
  }  
  // OK!  
  var rect1 = new Rectangle(10, 4);  
}  
// NOT OK!!!!  
var rect2 = Formes...
```

4.2 les modules externes

```
module Formes{  
  export var Cercle= {Point: 1,9};  
}  
console.log(Formes.Cercle.Point); // Affiche  
'1,9'
```