

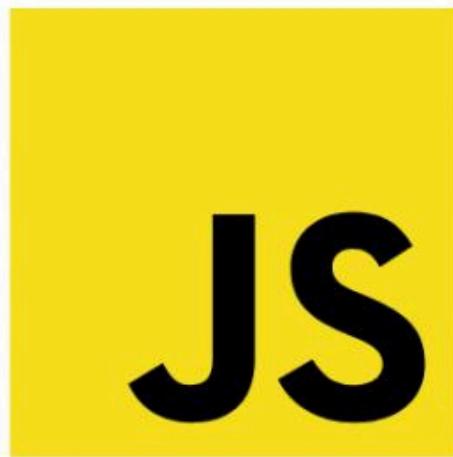
# TypeScript

Presentated by  
Table 4

TypeScript



# TypeScript ~JavaScript?



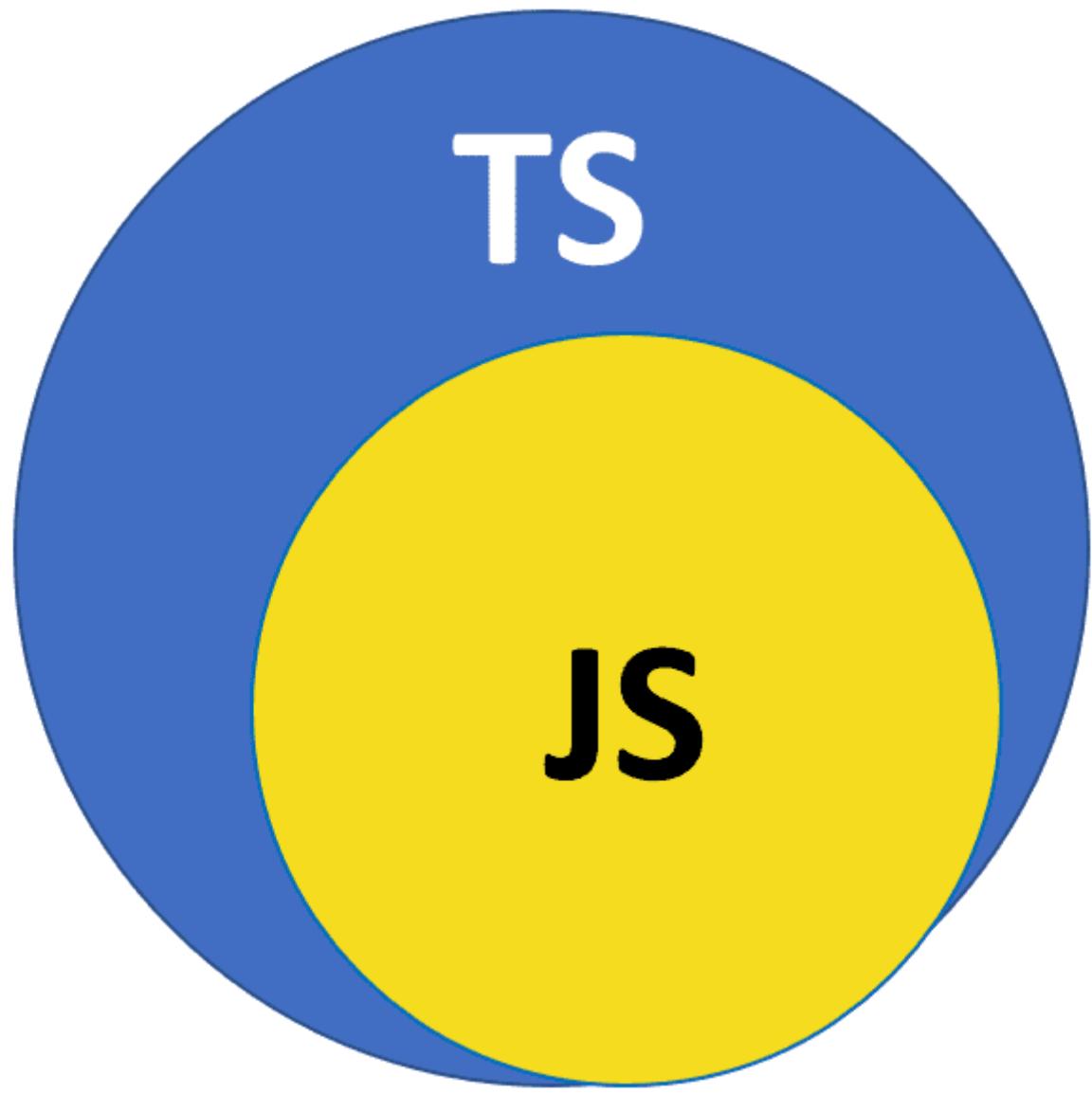
JAVASCRIPT



```
✖ Uncaught ReferenceError: variable index.js:1  
is not defined  
at index.js:1:1
```

```
✖ ► Uncaught TypeError: index.js:3  
variable.print is not a function  
at index.js:3:10
```

# TypeScript ?



# Installation

```
fatoumata@fatoumata-HP-ProBook-440-14-inch-G10-Notebook-PC:~$ sudo snap install node --classic
node (20/stable) 20.13.1 par OpenJS Foundation (iojs✓) installé
fatoumata@fatoumata-HP-ProBook-440-14-inch-G10-Notebook-PC:~$ node --version
v20.13.1
fatoumata@fatoumata-HP-ProBook-440-14-inch-G10-Notebook-PC:~$ npm --version
10.5.2
fatoumata@fatoumata-HP-ProBook-440-14-inch-G10-Notebook-PC:~$ █
fatoumata@fatoumata-HP-ProBook-440-14-inch-G10-Notebook-PC:~$ npm install typescript --save-dev
up to date, audited 2 packages in 706ms

found 0 vulnerabilities
fatoumata@fatoumata-HP-ProBook-440-14-inch-G10-Notebook-PC:~$ █
```

# Installation nodeJS

# Syntaxes de base

```
let variable;  
variable = "mot";  
variable = 123;  
variable = true;
```

```
let variable;
```

Codeium: Explain Problem

```
let variable: any
```

```
let a: any;  
let x: number;  
let y: string;  
let z: boolean;
```

```
a = 123;  
a = "Str";  
a = true;
```

```
y = 123;  
y = "Str";  
y = true;
```

```
z = 123;  
z = "Str";  
z = true;
```

## Variables

# Syntaxes de base

Tableaux

```
let friends: any[] = ["Alexis", "Justine", "Sophie"];
friends = [27, 25];
```

```
let identity: (number | string)[] = ["Alexis", 27];
identity = ["Justine", 25];
```

```
let user: { name: string; age: number; isMajeur: boolean } = {
  name: "Alexis",
  age: 27,
  isMajeur: true
};
```

Objet

## Structures de données

# Quelques méthodes

# indexOf

```
const index: number = livres.indexOf(livre);
const row = tbody.insertRow();
row.innerHTML =
<td>${livre.titre}</td>
<td>${livre.auteur}</td>
<td>${livre.année}</td>
<td></td>
<td> <button type="button" onclick="supprimer(${index})">
`;
```

- ▶ (2) [{-}, {-}]
- ▶ {titre: '*Incidunt ipsum quo*'}
- ▶ (3) [{-}, {-}, {-}]
- ▶ {titre: '*Enim fugiat sapient*'}
- ▶ (4) [{-}, {-}, {-}, {-}]

# TypeOf

```
/** Test type union **/

/** TypeOf **/


    // Utilisé pour vérifier le type d'une variable. Très utile dans le contrôle de flux pour s'assurer que les opérations sur les variables sont sûres.
let type :string|number|bigint... = typeof (referentiels);
let string :string = "typescript"
console.log(type); // résultat : Objet
console.log(typeof string); // résultat : string
```

# instanceof

```
/** instanceof */
//Utilisé pour tester si un objet est une instance d'une classe ou d'un constructeur spécifique.

Show usages
class Voiture {
    Show usages
    constructor(public mark: string, public model: string) {}
}

let myCar : Voiture = new Voiture( mark: "Toyota", model: "Corolla");

if (myCar instanceof Voiture) {
    console.log('myCar est un instance de voiture');
} else {
    console.log('myCar n\'est pas un instance de voiture');
}
```

# in

Effacer

```
/** in **/ //Vérifie si une propriété spécifique existe dans un objet. Show usages interface Game { title: string; developer: string; annee?: number; }

let game1: Game = {
  title: "The Witcher 3", developer: "CD Projekt Red"
};

let game2: Game = {
  title: "The Witcher 4", developer: "CD Projekt Red", annee:2022
};
```

```
if ('annee' in game1) {
  console.log('annee existe');
} else {
  console.log('annee not exist');

}

if ('annee' in game2) {
  console.log('annee existe');
} else {
  console.log('annee not exist');
```

# Foreach

```
livres.forEach(livre => {//Permet de parcourir le tableau livres
    const row = tbody.insertRow();
    row.innerHTML =
    <td>${livre.titre}</td>
    <td>${livre.auteur}</td>
    <td>${livre.année}</td>
    <td></td>
    <td> <button type="button" onclick=""> <i class='fas fa-trash'></i>
    </button> </td>
    `;
```

Titre	Auteur	Année	Statut	Action
Livre 2	Sidy	14/05/2024		<button>Supprimer</button>
Livre 1	Fatoumata	01/05/2024		<button>Supprimer</button>

# Every

TypeScript

```
arrayName.every(callbackFunction)
```

```
25
26 //===== Every() =====
27
28 |
29 const tableau: any[]=[6,3,4,10];
30
31 const reponse:boolean =tableau.every(nombre =>nombre > 2); //on vérifie si tout les éléments du tableaux sont > 2
32
33 console.log(reponse); // Sortie: true;
34
35
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- sidydiopbalde@sidydiopbalde-HP-ProBook-440-14-inch-G10-Notebook-PC:~/Typescript\$ tsc
- sidydiopbalde@sidydiopbalde-HP-ProBook-440-14-inch-G10-Notebook-PC:~/Typescript\$ node test.js  
true
- sidydiopbalde@sidydiopbalde-HP-ProBook-440-14-inch-G10-Notebook-PC:~/Typescript\$

# Every

## Exemple d'utilisation dans une fonction

```
34
35  function Verifier(element: number[],x:number): boolean {
36
37      const bool: boolean = element.every(el => el > x);
38
39      return bool;
40  }
41 //appel de la fonction dev
42 const tab2:number[]=[3,20,-5,32];
43
44 const answer:boolean = Verifier(tab2,1);
45 console.log(answer); //Sortie : false;
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```
• sidydiopbalde@sidydiopbalde-HP-ProBook-440-14-inch-G10-Notebook-PC:~/Typescript$ tsc
• sidydiopbalde@sidydiopbalde-HP-ProBook-440-14-inch-G10-Notebook-PC:~/Typescript$ node test.js
false
• sidydiopbalde@sidydiopbalde-HP-ProBook-440-14-inch-G10-Notebook-PC:~/Typescript$
```



# Unshift

TypeScript

```
arrayName.unshift(...elementsToAdd)|
```

```
function AjoutLivre(titre: string, auteur: string, année: string) {  
    const newLivre: livre = {titre, auteur, année};  
    //livres.push(newLivre);  
    livres.unshift(newLivre); //Permet d'ajouter un élément au début  
    console.log(newLivre);  
    console.log(livres);  
    AfficherLivres();  
}
```

## Gestionnaire de Bibliothèque

Titre du livre	Auteur	jj / mm / aaaa	Ajouter Livre
----------------	--------	----------------	---------------

Titre	Auteur	Année	Statut	Action
Livre 2	Sidy	14/05/2024		
Livre 1	Fatoumata	01/05/2024		

# push

Show usages

```
function ajoutLivre(titre: string, author: string, annee: string) : void {  
    const nouveauLivre : Livre = { titre , author , annee}  
    // PUSH PERMET D'AJOUTER UN ELEMENT DANS UN TABLEAU A LA DERNIER POSITION  
    livres.push(nouveauLivre)  
    affichageLivre()  
}
```

## Gestionnaire de Bibliothèque

livre 2

baba issa

Titre	Auteur	Année	Statut	Action
livre 1	cheikh	3/4/2024	Disponible	Supprimer
livre 2	baba issa	3/4/2024	Disponible	Supprimer

# sort

```
function trieTitre() {  
    livres.sort((a, b) => a.titre.localeCompare(b.titre));  
    affichageLivre(livres);  
}
```

Titre	Auteur	Année
Dolor molestiae dolo	Sint id cupidatat	1982/10/13
Est cupiditate dele	Reprehenderit et co	2012/08/26
Quaerat voluptas qui	Ipsa adipisci irure	2007/05/24
Ut eveniet eveniet	In rerum deserunt te	1997/07/08

tabnine: test | explain | document | ask | Codeium: Refactor | Explain | Generate JSDoc | ×

```
function trieAnnée() {  
    livres.sort((a, b) => a.année.localeCompare(b.année));  
    affichageLivre(livres);  
}
```

Titre	Auteur	Année
molestiae dolo	Sint id cupidatat	1982/10/13
niet eveniet	In rerum deserunt te	1997/07/08
it voluptas qui	Ipsa adipisci irure	2007/05/24
iditate dele	Reprehenderit et co	2012/08/26

# Splice

```
function spliceButton(index:number) :void {
    console.log(index)
    //splice() peut être utilisée pour supprimer des éléments sans laisser de "trous" dans le tableau.
    // livres.splice(index,1)
    //splice() peut également être utilisée pour ajouter des éléments à un tableau à un indice spécifique.
    // livres.splice(index, 0, {titre: "Fahrenheit 451", author: "Ray Bradbury", annee: "12/1/2023"});
    // Remplaçons un livre par un autre :
    livres.splice(index, deleteCount: 1, items: { titre: "Brave New World Revisited", author: "Aldous Huxley", annee: "1
    affichageLivre()
}
```

Titre	Auteur
Incididunt natus sin	Et esse assumenda qu
Recusandae Aut sit	Iste excepteur quas
Id ipsa provident	Numquam ad et sunt u

Titre	Auteur	Année
Brave New World Revisited	Aldous Huxley	12/1/2023
Recusandae Aut sit	Iste excepteur quas	0/10/1977
Id ipsa provident	Numquam ad et sunt u	6/6/1984

```
function supprimer(index: number) [
    console.log(index);
    livres.splice(index, 1); //Permet de supprimer l'élément à l'index 'index'
    AfficherLivres();
]
```

# Filter

```
function filterButton(element: string | number) {
  let filterList: Livre[] = livres.filter(livre => livre.auteur == element || livre.exemplaire == element);
  affichageLivre(filterList)
}
```

Dolores eum quis obc

Recherche

Titre	Auteur	Année	Stock	Action
Do voluptatum ex con	Dolores eum quis obc	1/3/1982	99	Supprimer

# Shift

Titre	Auteur
Tempora in voluptate	Molestiae sed conseq
Sed qui et dolor mol	Ut at autem modi fug
Velit rerum dolor q	Cumque molestiae omn

```
function shiftRemoveFirst() :void {
    // SHIFT: Supprime le 1er élément dans un tableau [0]
    livres.shift()
    affichageLivre()
}
```

```
function affichageLivre(): void
```

Titre	Auteur
Sed qui et dolor mol	Ut at autem modi fug
Velit rerum dolor q	Cumque molestiae omn

# Reduce

```
ts > ...
const array = [1, 2, 3, 4, 5];

// Addition des éléments du tableau en partant de la gauche
const sum = array.reduce((accumulator, currentValue) => accumulator + currentValue);

console.log(sum); // Résultat: 15
```



# join

```
const tabdate = [jourFormate, moisFormate, annee];
const date = tabdate.join("//On veut concaténer la date en une seule chaîne");
AjoutLivre(titre.value, auteur.value, date);
```

```
Show usages
function ajoutLivre(titre: string, author: string, annee: string) :void {
    const nouveauLivre : Livre = { titre , author , annee}
    // PUSH PERMET D'AJOUTER UN ELEMENT DANS UN TABLEAU A LA DERNIER POSITION
    livres.push(nouveauLivre)
    affichageLivre()
    date = tabdate.join ("//");
```

Année

14/05/2024

01/05/2024

# ReduceLeft~ReduceRight

```
st.ts > ...
  const array = [1, 2, 3, 4, 5];

  // Addition des éléments du tableau en partant de la droite
  const sum = array.reduceRight((accumulator, currentValue) => accumulator + currentValue);

  console.log(sum); // Résultat: 15
```



```
.ts > ...
  const array = [1, 2, 3, 4, 5];

  // Addition des éléments du tableau en partant de la gauche
  const sum = array.reduce((accumulator, currentValue) => accumulator + currentValue);

  console.log(sum); // Résultat: 15
```



# Concat

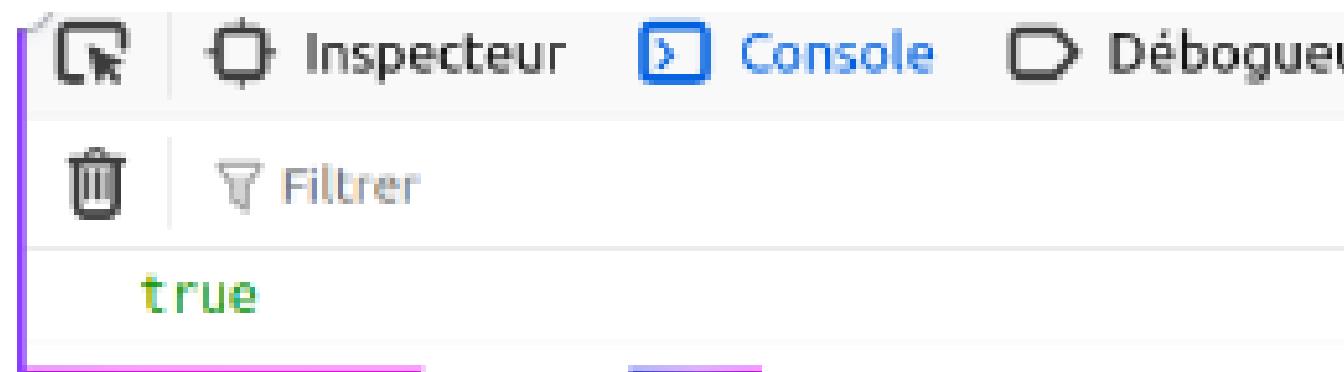
```
src.js / ...
const array1 = [1, 2];
const array2 = [3, 4];
const array3 = [5, 6];

// Fusion de plusieurs tableaux
const newArray = array1.concat(array2, array3);

console.log(newArray); // Résultat: [1, 2, 3, 4, 5, 6]
```

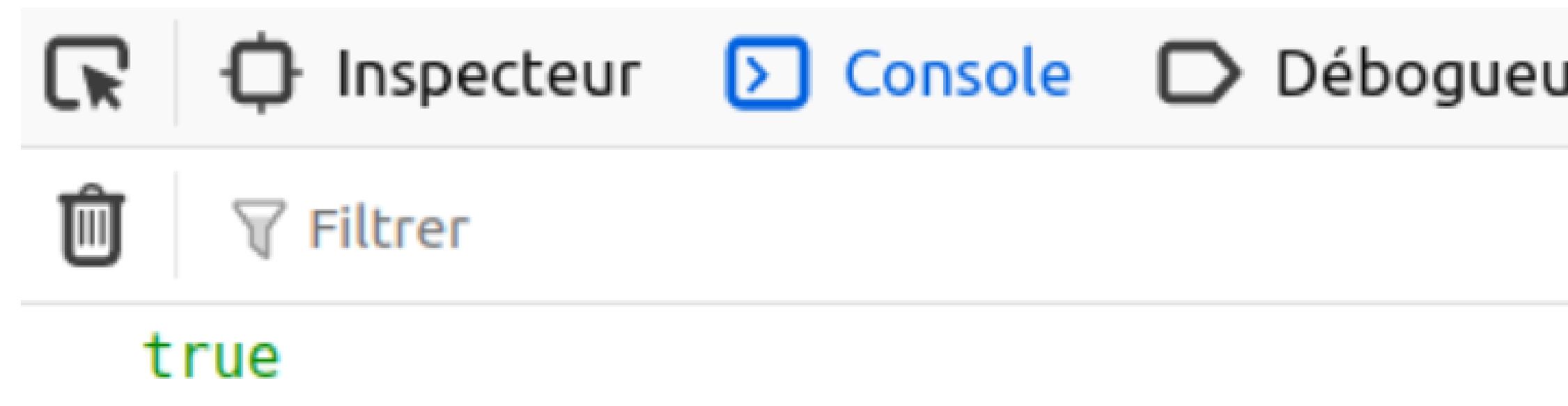
# Some

```
1 //exemple avec un tableau
2 var numbers = [1, 2, 3, 4, 5];
3 var hasEvenNumber = numbers.some(function (number) {
4   return number % 2 === 0;
5 });
6 console.log(hasEvenNumber);
7
```



# Some

```
//exemple avec un tableau d'objet
var users = [
  { name: 'Alice', isActive: false },
  { name: 'Bob', isActive: true },
  { name: 'Charlie', isActive: false }
];
var hasActiveUser = users.some(function (user) {
  return user.isActive;
});
console.log(hasActiveUser);
```



# Pop

Titre	Auteur
Sed qui et dolor mol	Ut at autem modi fug
Velit rerum dolor q	Cumque molestiae omn

```
src/app.ts :  
function p  
    // SHIFT: Supprime le dernier élément dans un tableau taille - 1  
    livres.pop()  
    affichageLivre()  
}
```

Titre	Auteur
Sed qui et dolor mol	Ut at autem modi fug



**Thank you**