



---

# Bài 3

# Tổng quan về TypeScript

Module: Advanced Programming with JavaScript

- Trình bày được cú pháp khai báo biến
- Sử dụng được biến trong Typescript
- Sử dụng được kiểu dữ liệu trong Typescript
- Sử dụng được kiểu enum
- Trình bày được kiểu optional trong class
- Khai báo và sử dụng được hàm (có tham số) trong TypeScript
- Khai báo được kiểu liên hợp (Unions) trong TypeScript
- Sử dụng được từ khóa readonly trong TypeScript
- Khai báo và sử dụng được kiểu Tuple trong TypeScript



---

# Thảo luận

TypeScript là gì?

# Giới thiệu về TypeScript

---



- Mục tiêu của TypeScript là trở thành một trình kiểm soát “tĩnh” cho các chương trình JavaScript
- TypeScript bổ sung tùy chọn kiểu tĩnh và lớp hướng đối tượng mà điều này không có ở JavaScript
- TypeScript có thể sử dụng để phát triển các ứng dụng chạy ở client-side (Angular, ReactJS,...) và server-side (NodeJS)

# Điểm mạnh của TypeScript

---



- Dễ dàng mở rộng dự án
- Được sử dụng ở nhiều framework khác nhau
- Tài liệu nhiều, cộng đồng đông đảo
- Hỗ trợ các tính năng mới nhất của JavaScript
- Hỗ trợ OOP mạnh
  - Hầu hết các cú pháp hướng đối tượng đều được hỗ trợ bởi Typescript như kế thừa, đóng gói, constructor, abstract, interface, implement, override...v.v

# Kiểu động và kiểu tĩnh



- Trong JavaScript, ta làm việc với kiểu động (dynamic type). Còn với TypeScript, ta làm việc với kiểu tĩnh (static type).
- Kiểu tĩnh có những điểm mạnh như giúp giảm thiểu thời gian xử lý và cải thiện quá trình coding một cách hiệu quả hơn so với kiểu động.
- Biến có kiểu động là biến có thể chứa bất kì kiểu giá trị nào mà không bị giới hạn bởi một kiểu nhất định

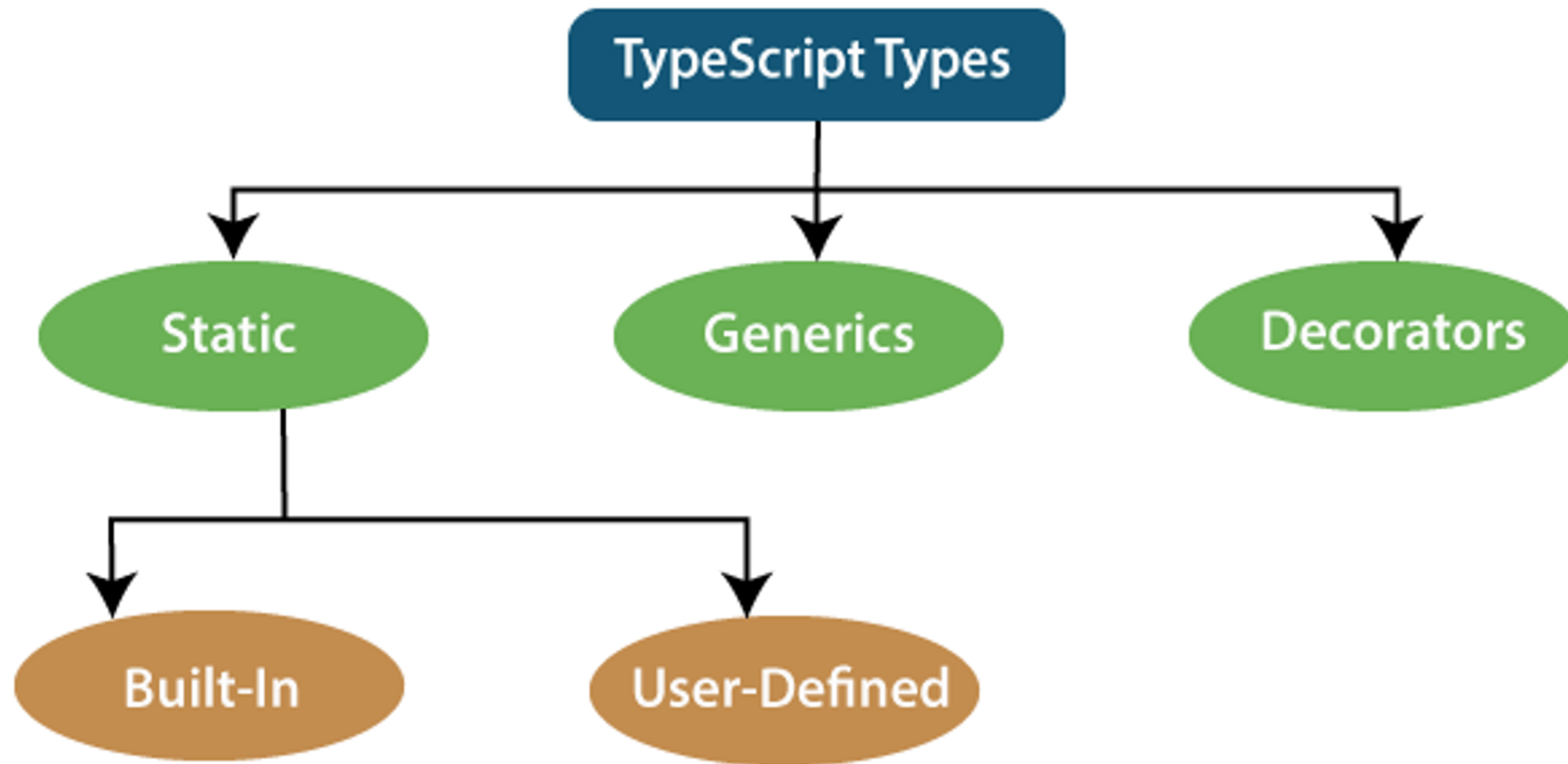
JavaScript

```
let a = 10;  
a = "Xin chào"
```

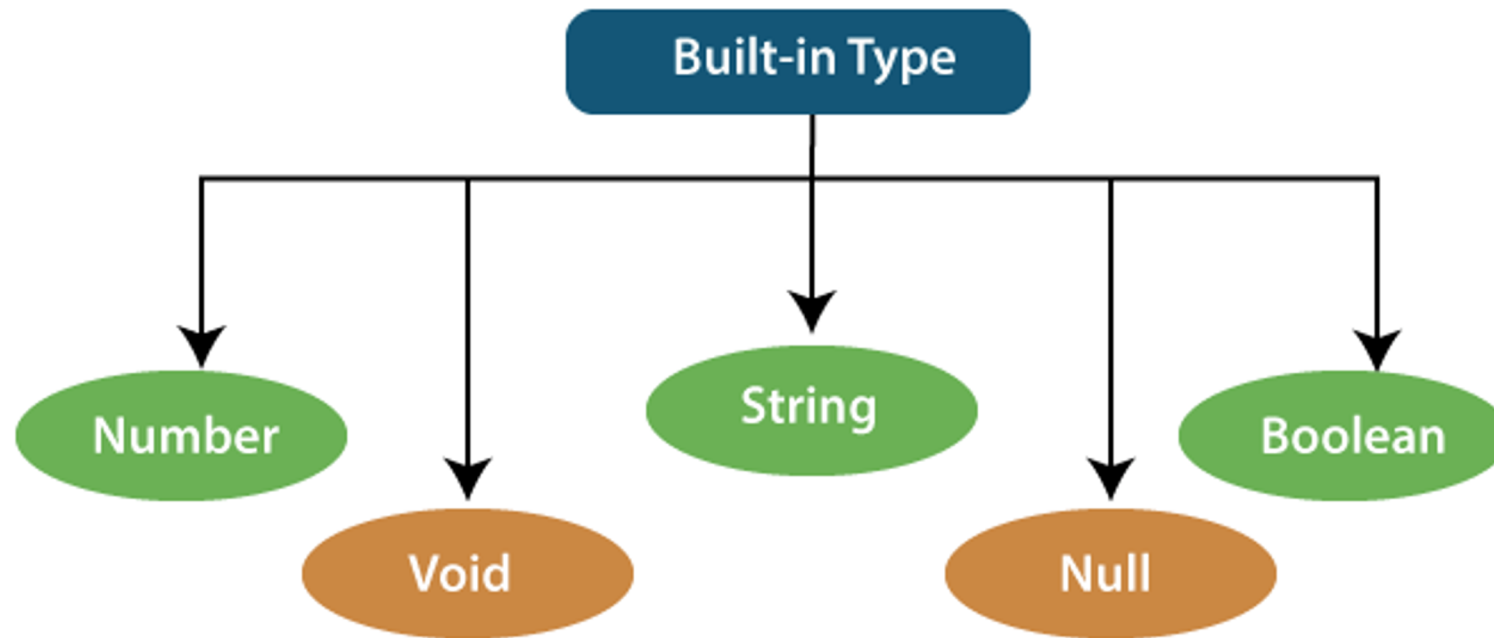
TypeScript

```
let a: number = 10;
```

# Kiểu dữ liệu trong TypeScript



# Kiểu dữ liệu nguyên thủy







---

# Demo

Các kiểu dữ liệu nguyên thủy

# Kiểu do người dùng định nghĩa

---



- Kiểu Array
- Kiểu Tuple
- Kiểu Enum
- Kiểu Interface
- Classes
- Functions
- Generic
- Decorators



---

# Demo

Kiểu do người dùng định nghĩa

# Kiểu liên hợp (Unions)



- Với một union, bạn có thể khai báo rằng một kiểu có thể là một trong nhiều kiểu.

```
type WindowStates = "open" | "closed" | "minimized";  
type LockStates = "locked" | "unlocked";  
type PositiveOddNumbersUnderTen = 1 | 3 | 5 | 7 | 9;
```

- Unions cung cấp một cách xử lý các loại khác nhau. Ví dụ:
  - Có thể có một hàm nhận một mảng hoặc một chuỗi

```
function getLength(obj: string | string[]) {  
    return obj.length;  
}
```



---

# Demo

Kiểu liên hợp (Unions)



# Từ khóa readonly

---

Trong TypeScript, chúng ta có thể khai báo các thuộc tính được đánh dấu là chỉ đọc với từ khóa `readonly`.

```
interface SomeType {  
    readonly prop: string;  
}
```

Sử dụng từ khóa `readonly` không có ngụ ý rằng một giá trị là hoàn toàn bất biến (immutable).

# Thuộc tính tùy chọn

---



Mỗi thuộc tính trong một kiểu đối tượng có thể chỉ định có phải là tùy chọn hay không. Ví dụ:

```
interface PaintOptions {  
  shape: Shape;  
  xPos?: number;  
  yPos?: number;  
}
```

Trong ví dụ trên, cả xPos và yPos đều được coi là tùy chọn.



---

# Demo

Từ khóa readonly và kiểu tùy chọn





---

# Cấu hình TypeScript

# Tập tin tsconfig.json

---



- Tập tin tsconfig.json trong một thư mục chỉ ra rằng thư mục đó là thư mục gốc của một dự án TypeScript.
- Tập tin tsconfig.json chỉ định các tập tin gốc và các tùy chọn biên dịch cần thiết để biên dịch dự án.
- Sử dụng tsconfig.json
  - Khi gọi lệnh tsc mà không có tập tin đầu vào, trình biên dịch sẽ tìm kiếm tập tin tsconfig.json bắt đầu trong thư mục hiện tại và tiếp tục lên các thư mục cấp trên.
  - Gọi lệnh **tsc** không có tập tin đầu vào nhưng có tùy chọn dòng lệnh **--project** (hoặc chỉ **-p**) chỉ định đường dẫn của thư mục chứa tập tin tsconfig.json (hoặc đường dẫn đến tập .json hợp lệ chứa cấu hình).



---

# Demo

Cấu hình TypeScript trong dự án

Qua bài học này, chúng ta đã tìm hiểu:

- TypeScript là gì
- Điểm mạnh của TypeScript
- Tìm hiểu cú pháp khai báo biến
- Sử dụng các kiểu dữ liệu trong Typescript
- Sử dụng được kiểu enum
- Sử dụng được kiểu optional (tùy chọn) trong class
- Khai báo được kiểu liên hợp (Unions) trong TypeScript
- Sử dụng được từ khóa readonly trong TypeScript