TypeScript Basics & Best Practices

Kevin Van Houtte

TypeScript Backstory

Optionally typed

Transpiles to ES5/6/7

Prevents common bugs and errors

OO design

Declarations

Module system

Generics

Union and type guards

JavaScript

```
var Greeter = (function () {
    function Greeter(message) {
        this.greeting = message;
    }
    Greeter.prototype.greet =
function () {
        return "Hello, " + this.greeting;
    };
    return Greeter;
}());
```

TypeScript

```
class Greeter {
    greeting: string;
    constructor(message: string) {
        this.greeting = message;
    }
    greet() {
        return "Hello, " + this.greeting;
    }
}
```

Optionally typed

Transpiles to ES5/6/7

Prevents common bugs and errors

OO design

Declarations

Module system

Generics

Union and type guards

Optionally typed

Transpiles to ES5/6/7

Prevents common bugs and errors

OO design

Declarations

Module system

Generics

Union and type guards

JavaScript	TypeScript
let a = 123 a.trim() TypeError: undefined is not a function	let a: string = 123 a.trim() Cannot convert ‘number’ to ‘string’

Optionally typed

Transpiles to ES5/6/7

Prevents common bugs and errors

OO design

Declarations

Module system

Generics

Union and type guards

Declare It!

- Let
- Const

Old days

Why not just var?

- Var in scope block can be accessible out of the scope
- The same property can be declared twice (ex twice var i = 0)
- Etc...

Variable recognition

```
function f(input: boolean) {
    let a = 100;
    if (input) {
        let b = a + 1;
        return b;
    }
    // Error: 'b' doesn't exist here
    return b;
}
```

Double trouble

Not possible:

```
let x = 10;
let x = 20;
 // error: can't re-declare 'x' in the same scope
```

Order of declaration

```
a++
let a;
 // illegal to use 'a' before it's declared;
```

What is TypeScript?

Optionally typed

Transpiles to ES5/6/7

Prevents common bugs and errors

OO design

Declarations

Module system

Generics

Union and type guards

What is TypeScript?

Optionally typed

Transpiles to ES5/6/7

Prevents common bugs and errors

OO design  ?

Declarations

Module system

Generics

Union and type guards

- Can extend classes
- Can implement interfaces
- Members/methods (instance & static)
- Single constructor
- Access modifiers

```
class Snake extends Animal{}
```

- Can extend classes
- Can implement interfaces
- Members/methods (instance & static)
- Single constructor
- Access modifiers

```
class Snake implements Crawl{}
```

- Can extend classes
- Can implement interfaces
- Members/methods (instance & static)
- Single constructor
- Access modifiers

```
greeting:string;

greet(): string {
    return “Hello, “ + this.greeting;
}
```

- Can extend classes
- Can implement interfaces
- Members/methods (instance & static)
- Single constructor
- Access modifiers

```
constructor(message: string){
    this.greeting = message;
}
```

- Can extend classes
- Can implement interfaces
- Members/methods (instance & static)
- Single constructor
- Access modifiers

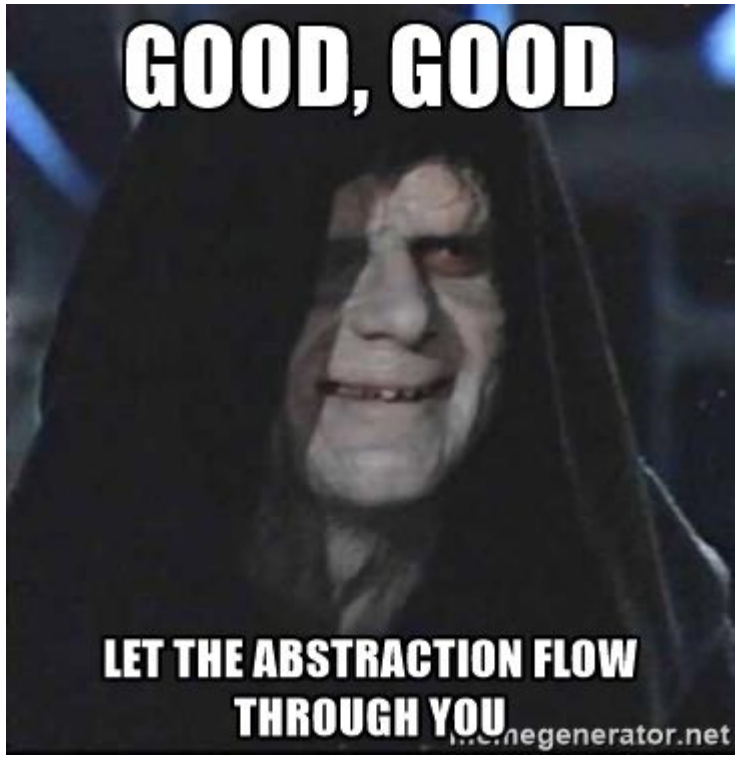
```
public write(){}

private secret: string;

protected constructor(name: string){}
```

Abstract Classes

- Prefix abstract
- Super class
- May not be initiated
- May contain implementation
- Abstract methods => must be implemented



Interfaces

- Prefix interface
- Enforcing that a class meets a particular contract
- intent of the class
- Extends (multiple) interfaces

What is TypeScript?

Optionally typed

Transpiles to ES5/6/7

Prevents common bugs and errors

OO design

Declarations

Module system

Generics

Union and type guards

Module System

Namespaces

- No name collisions
- Gets rid of the IIFE syntax
- No global variables
- Good for small applications

- Viable options to export and import are with:
  - variables
  - functions
  - classes
  - interfaces
  - type aliases

```
export interface StringValidator {}
export class Validator{}
export const regex= /^[0-9]+$/;
```

```
import { className } from "./fileName";
import * as validator from "./ZipCodeValidator";
let myValidator = new validator.ZipCodeValidator();
```

What is TypeScript?

Optionally typed

Transpiles to ES5/6/7

Prevents common bugs and errors

OO design

Declarations

Module system

Generics

Union and type guards

- Generic classes
- Generic functions

```
class GenericNumber<T> {
    zeroValue: T;
    add: (x: T, y: T) => T;
}

let myGenericNumber = new GenericNumber<number>();
myGenericNumber.zeroValue = 0;
myGenericNumber.add = function(x, y) { return x + y; };
```

- Generic classes
- Generic functions

```
function createInstance<A extends Animal>(c: new () => A): A {
    return new c();
}
```

Optionally typed

Transpiles to ES5/6/7

Prevents common bugs and errors

OO design

Declarations

Module system

Generics

Union types


ITS OVER
ITS FINALLY OVER
quickmeme.com

Practice

Fork: https://github.com/KevinVHoutte/Ordina-JWorks-TypeScript

Sources

https://www.typescriptlang.org/

https://github.com/Microsoft/TypeScript

https://github.com/Microsoft/TypeScript/wiki/Roadmap

https://github.com/KevinVHoutte/Ordina-JWorks-TypeScript