Agenda

- What isTypeScript
- Language Introduction / Live Coding
- TypeScript and Angular
- Conclusion

Always fun being a SOLE JavaScript developer

Team Development Turns into war

Hmmmm!!! Issues with Dynamically Typed Languages

TypeScript to the RESCUE

What is TypeScript

TypeScript

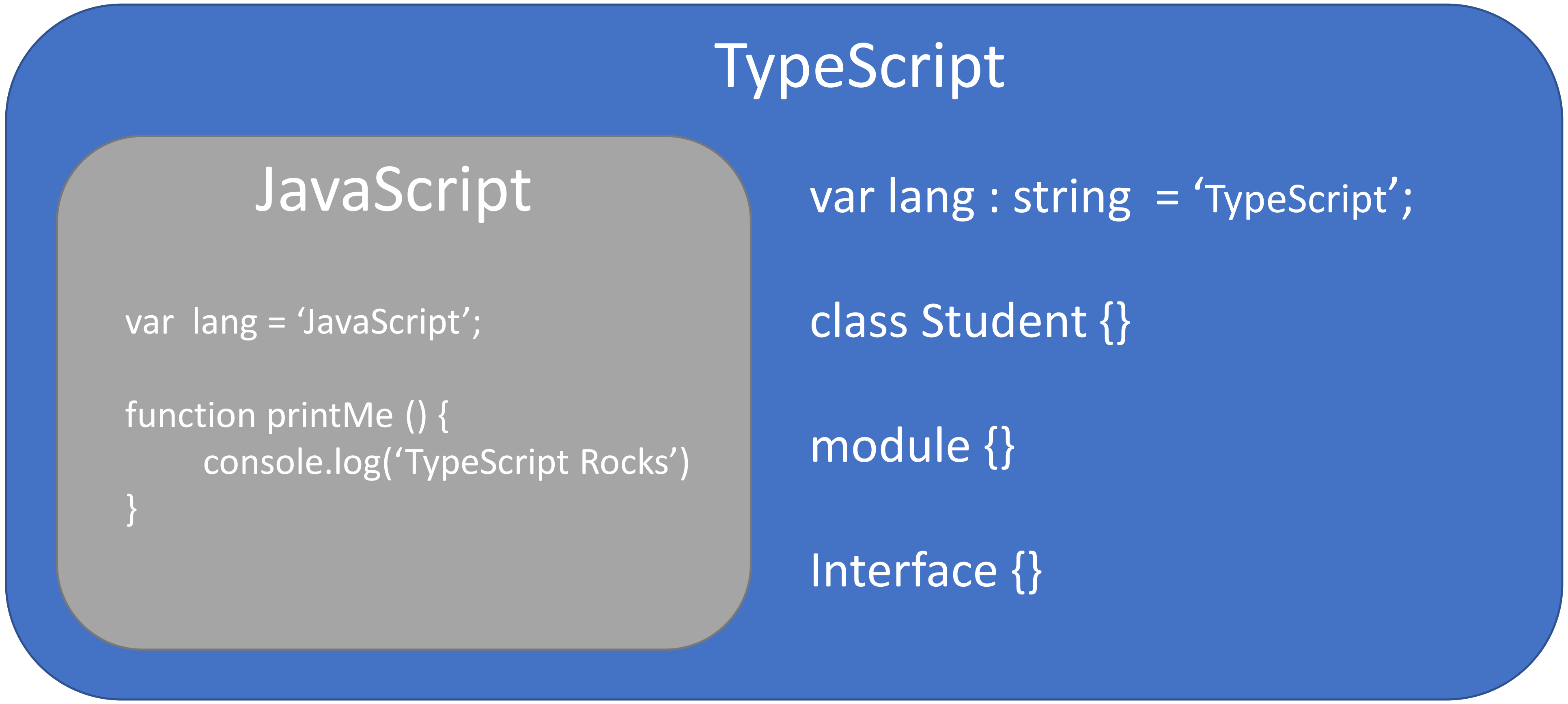
- ❖Superset of JavaScript
- ❖Extend JavaScript to facilitate writing large applications
- ❖Compile to plain JavaScript

What is TypeScript

TypeScript

- ❖Strongly Typed
- ❖Class-based Object-Orientation
- ❖Development Tooling
- ❖ TIP: If you are a C# or Java developer, then it feels like home

Superset of JavaScript



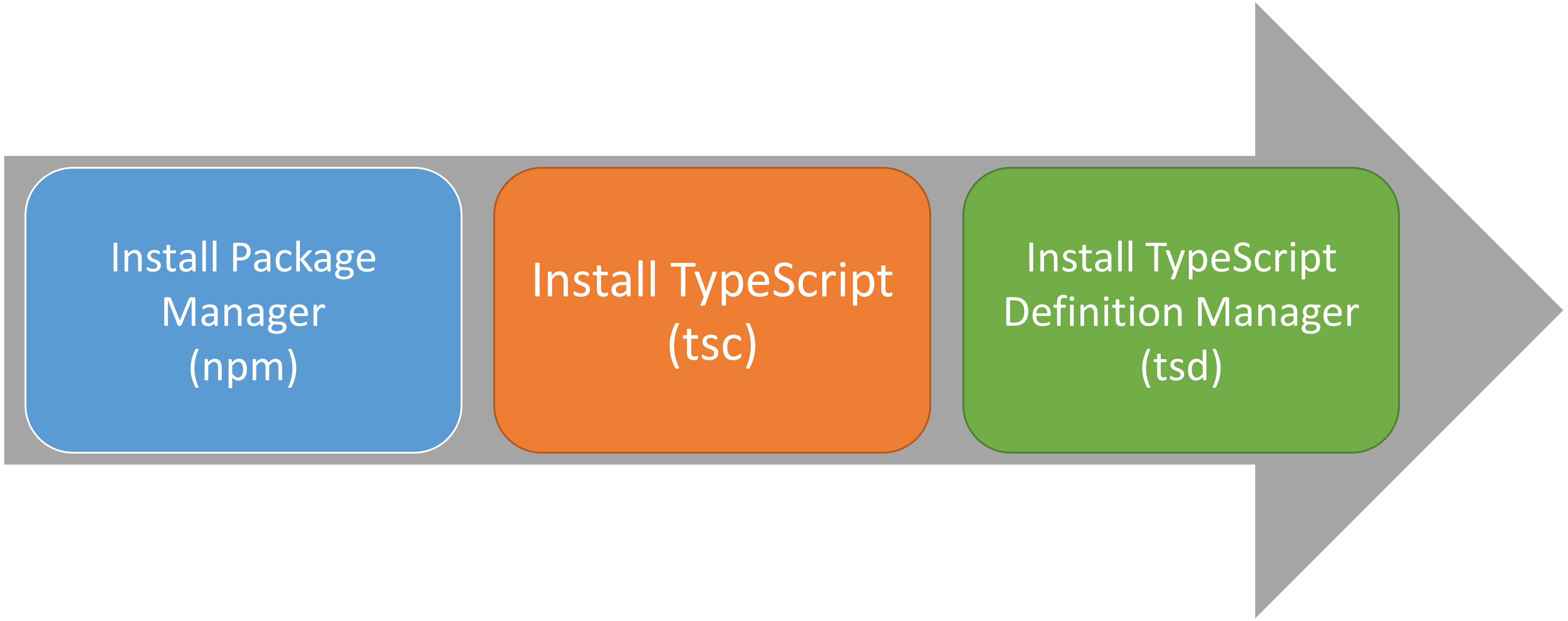
```
var  lang = 'JavaScript';

function printMe () {
        console.log('TypeScript Rocks')
}
```

```
var lang : string  = 'TypeScript';

class Student {}

module {}

Interface {}
```

TypeScript Transpiles to JavaScript

$$\begin{tikzcd}
\text{app.cs} \arrow[r, "\text{CSC}"] & \text{app.dll} \\
\text{app.js} \arrow[r, "\text{tsc}"] & \text{app.js (es*)}
\end{tikzcd}$$

Installing TypeScript


Install Package Manager (npm)
Install TypeScript (tsc)
Install TypeScript Definition Manager (tsd)

Installing TypeScript

```
>> npm install -g typescript

>> tsc helloworld.ts
```

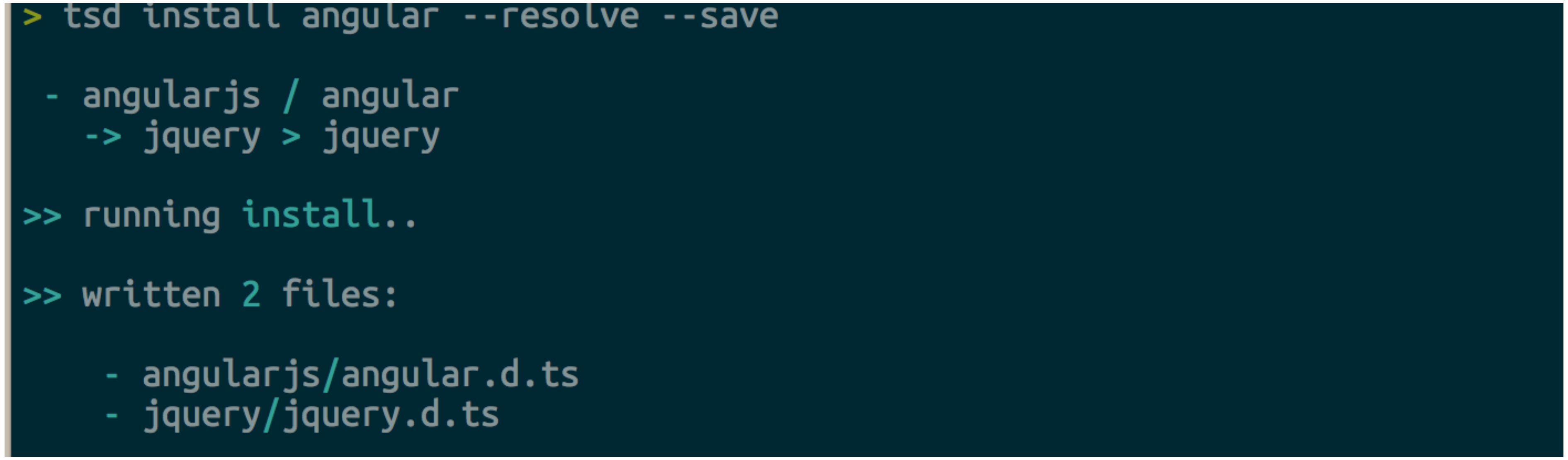
TypeScript Definition Files

- Describes the types defined in external libraries
- Suffixed .d.ts
- TypeScript Definition Manager (tsd)
  - Specialized package manager
  - Finds and installs TypeScript Definition files
  - Packages are found in DefinitelyTyped repository

TypeScript Definition Files

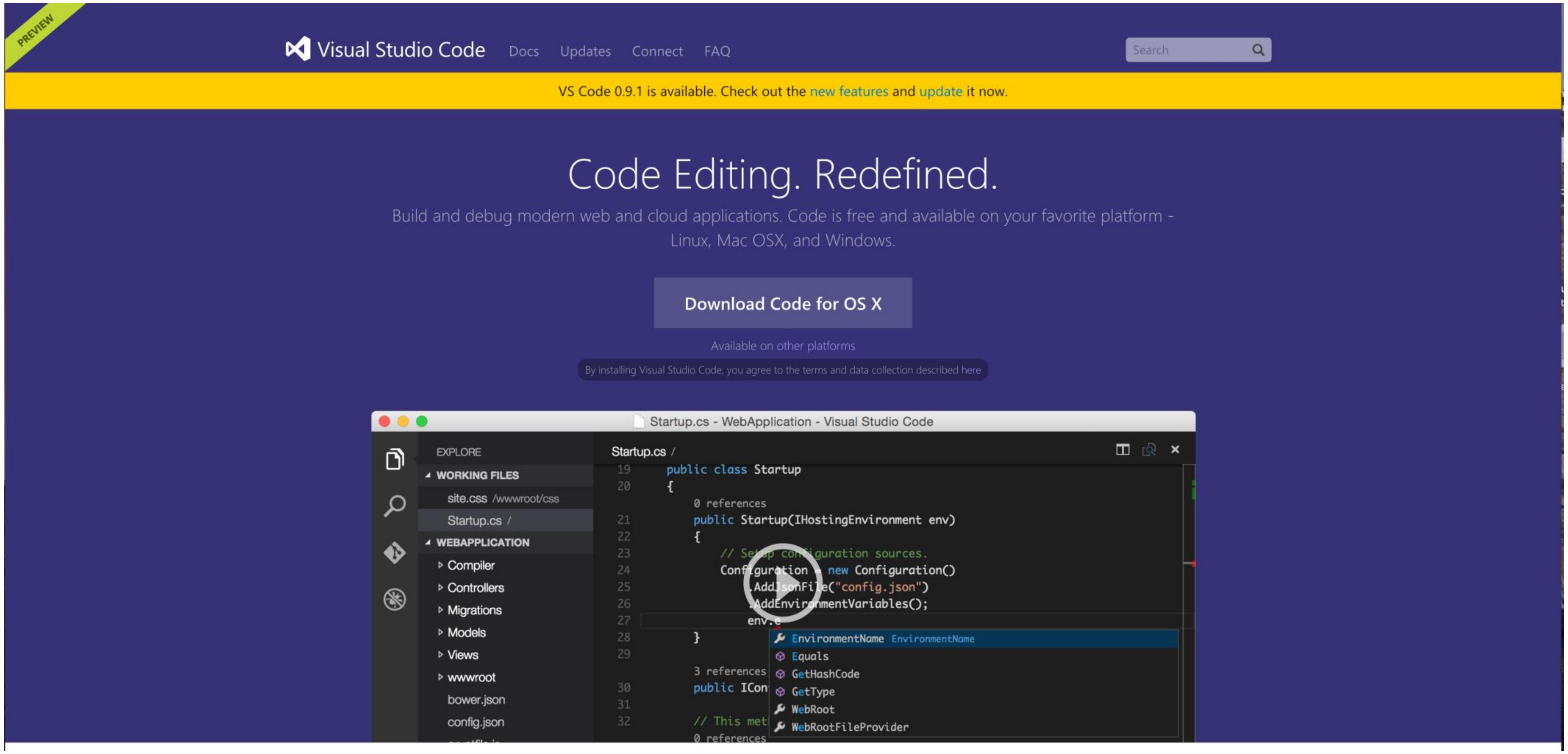
>> npm install tsd -g

>> tsd install angular –resolve –save



```
> tsd install angular --resolve --save

 - angularjs / angular
   -> jquery > jquery

>> running install..

>> written 2 files:

    - angularjs/angular.d.ts
    - jquery/jquery.d.ts
```

Code Editor of Choice



VS Code 0.9.1 is available. Check out the new features and update it now.

Code Editing. Redefined.

Build and debug modern web and cloud applications. Code is free and available on your favorite platform - Linux, Mac OSX, and Windows.

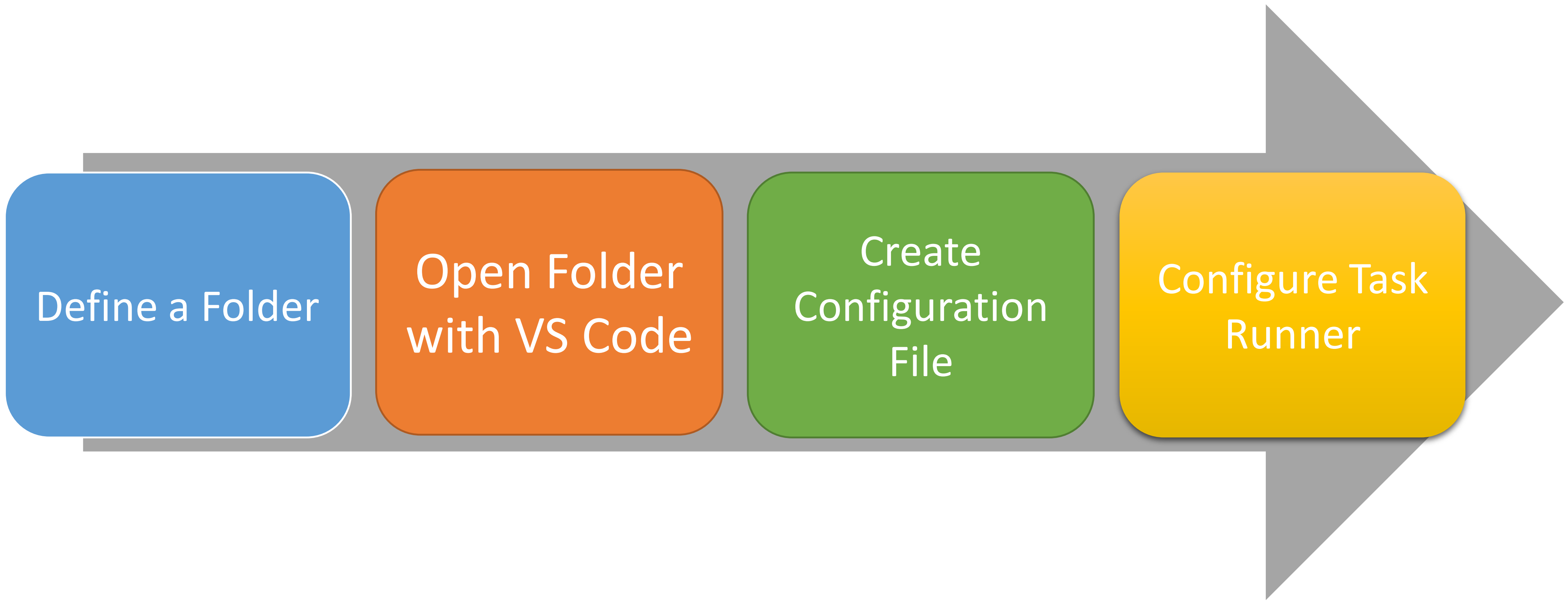
By installing Visual Studio Code, you agree to the terms and data collection described here

Startup.cs - WebApplication - Visual Studio Code

```
    public class Startup
    {
        0 references
        public Startup(IHostingEnvironment env)
        {
            // Setup configuration sources.
            Configuration = new Configuration()
                .AddJsonFile("config.json")
                .AddEnvironmentVariables();
                env.e
        }

        3 references
        public ICon

        // This met
        0 references
```

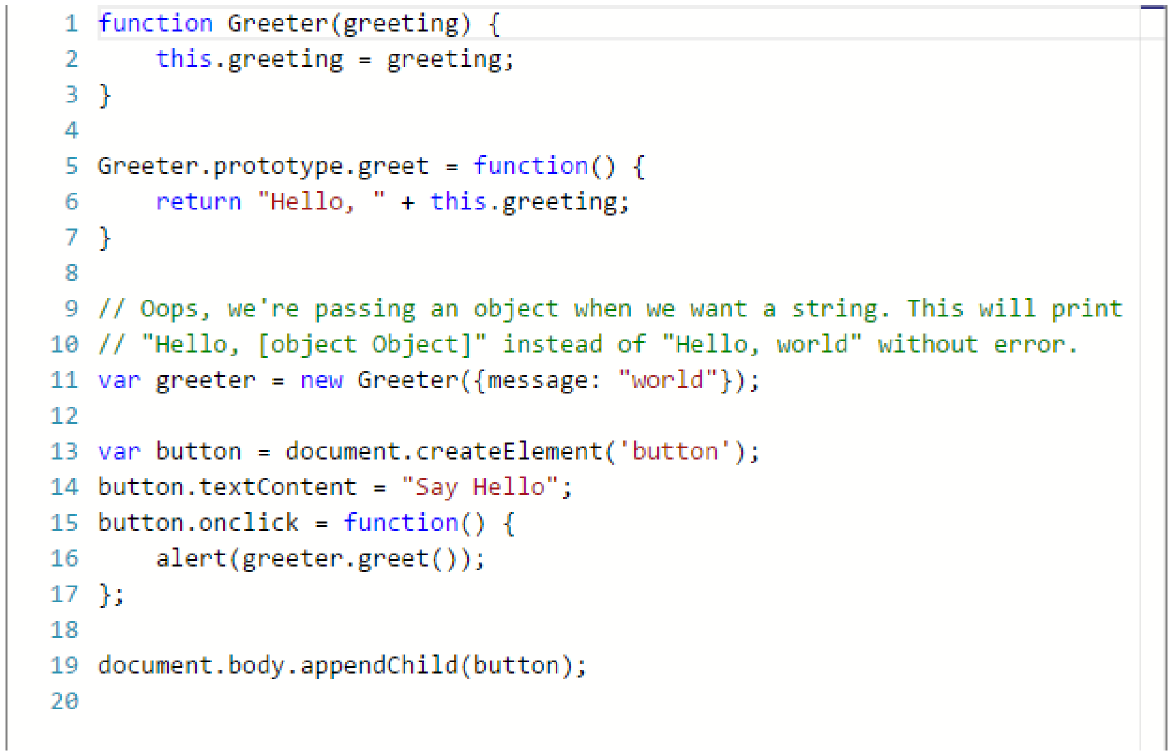
```
EnvironmentName  EnvironmentName
Equals
GetHashCode
GetType
WebRoot
WebRootFileProvider
```

Setting Up VS Code For TypeScript

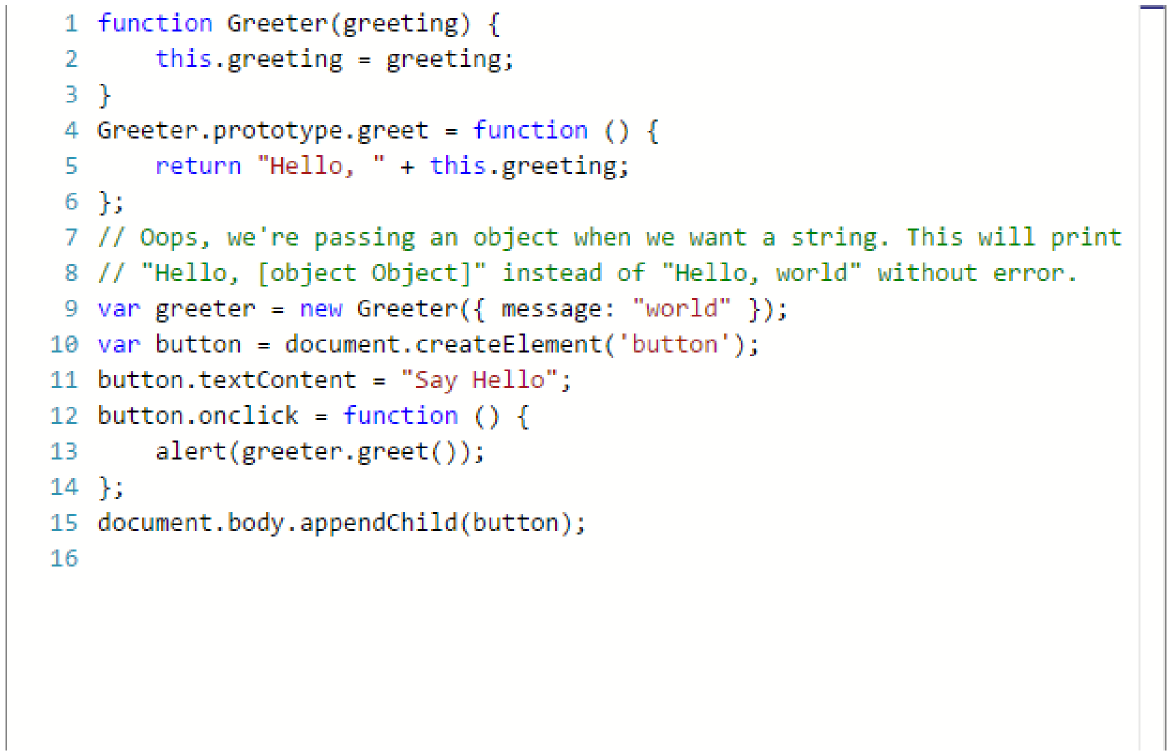


Code samples from the TypeScript Site

Can you Identify the problem HERE!!!

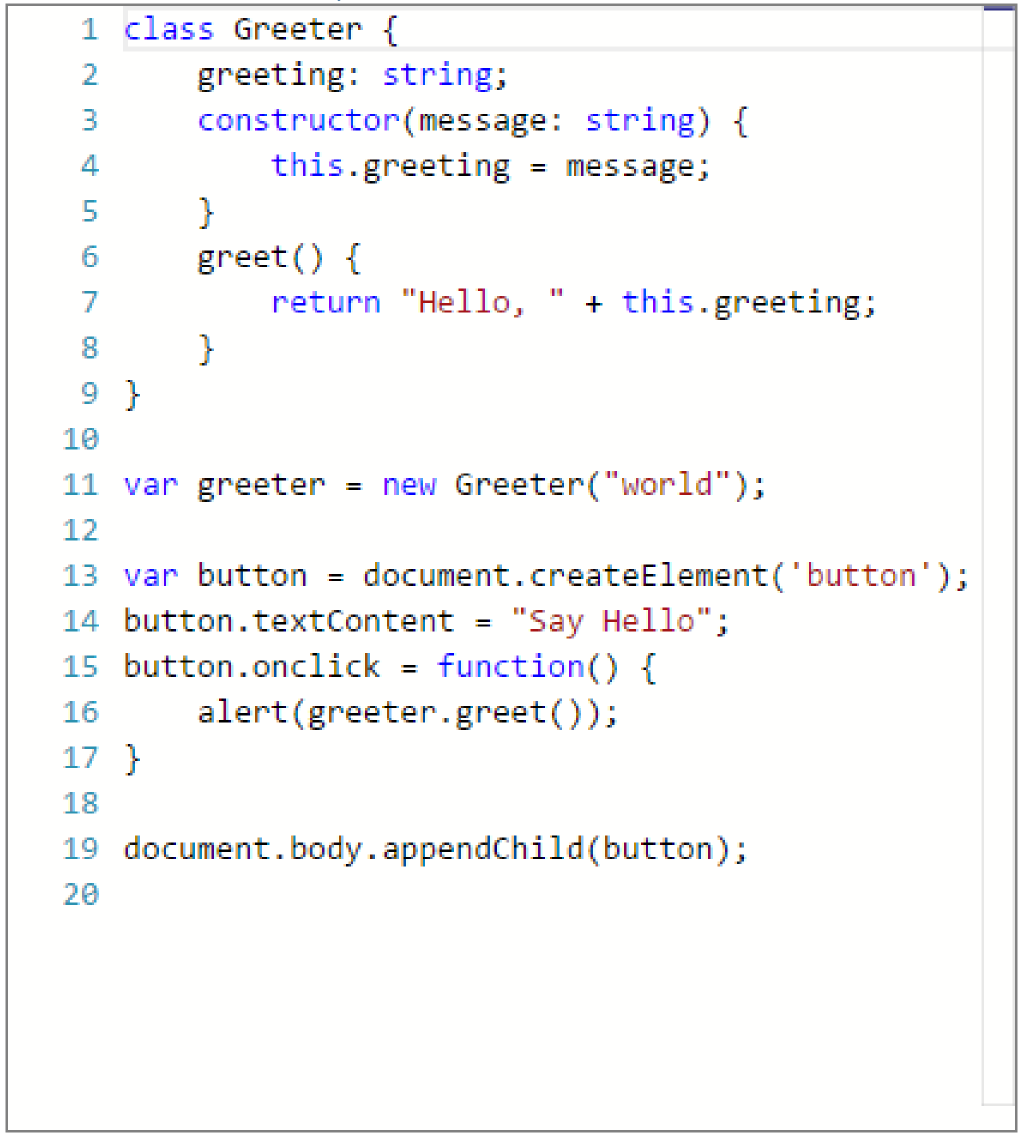


```
function Greeter(greeting) {
    this.greeting = greeting;
}

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

// Oops, we're passing an object when we want a string. This will print
// "Hello, [object Object]" instead of "Hello, world" without error.
var greeter = new Greeter({message: "world"});

var button = document.createElement('button');
button.textContent = "Say Hello";
button.onclick = function() {
    alert(greeter.greet());
};

document.body.appendChild(button);
```



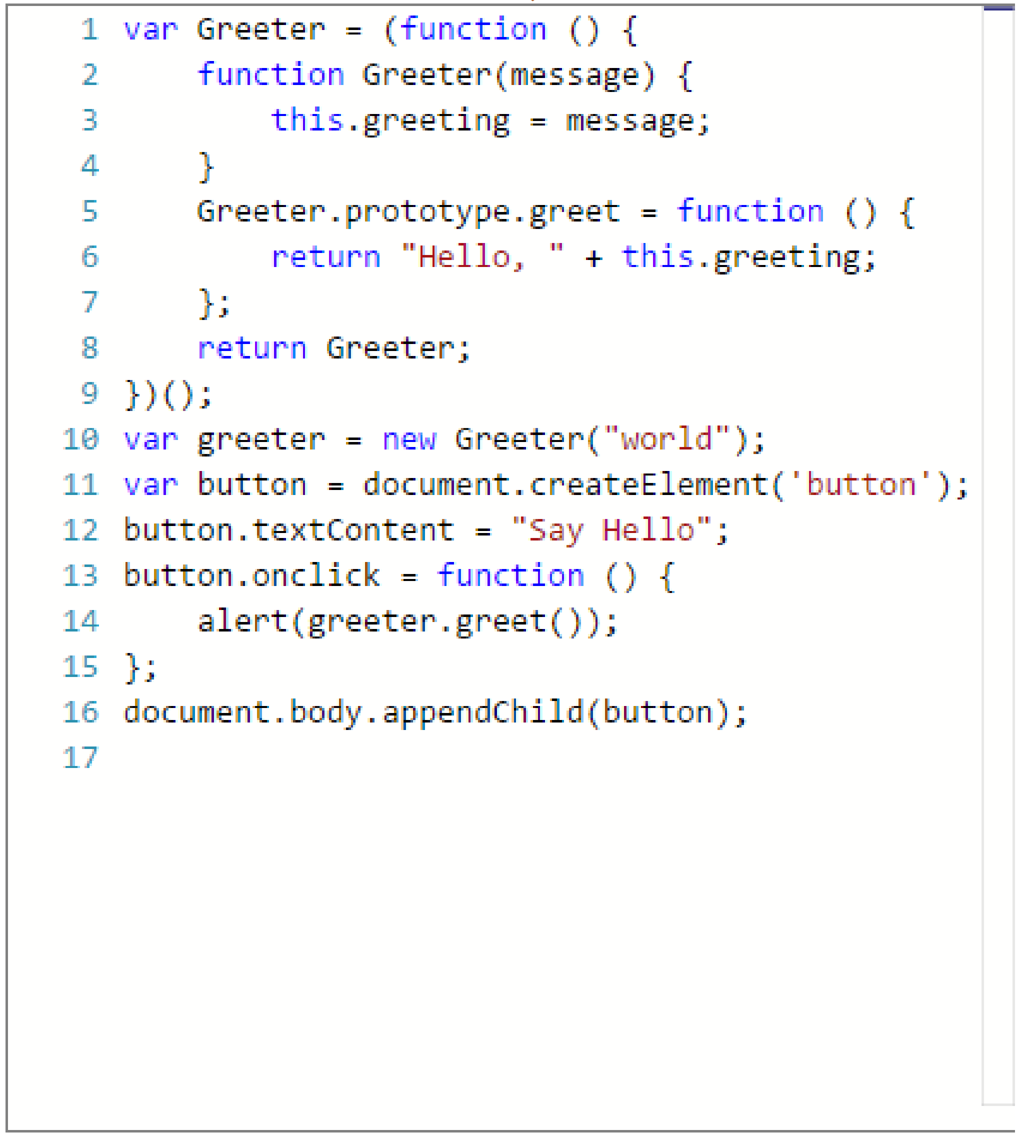
```
function Greeter(greeting) {
    this.greeting = greeting;
}
Greeter.prototype.greet = function () {
    return "Hello, " + this.greeting;
};
// Oops, we're passing an object when we want a string. This will print
// "Hello, [object Object]" instead of "Hello, world" without error.
var greeter = new Greeter({ message: "world" });
var button = document.createElement('button');
button.textContent = "Say Hello";
button.onclick = function () {
    alert(greeter.greet());
};
document.body.appendChild(button);
```

JavaScript



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

var greeter = new Greeter("world");

var button = document.createElement('button');
button.textContent = "Say Hello";
button.onclick = function() {
    alert(greeter.greet());
}

document.body.appendChild(button);
```

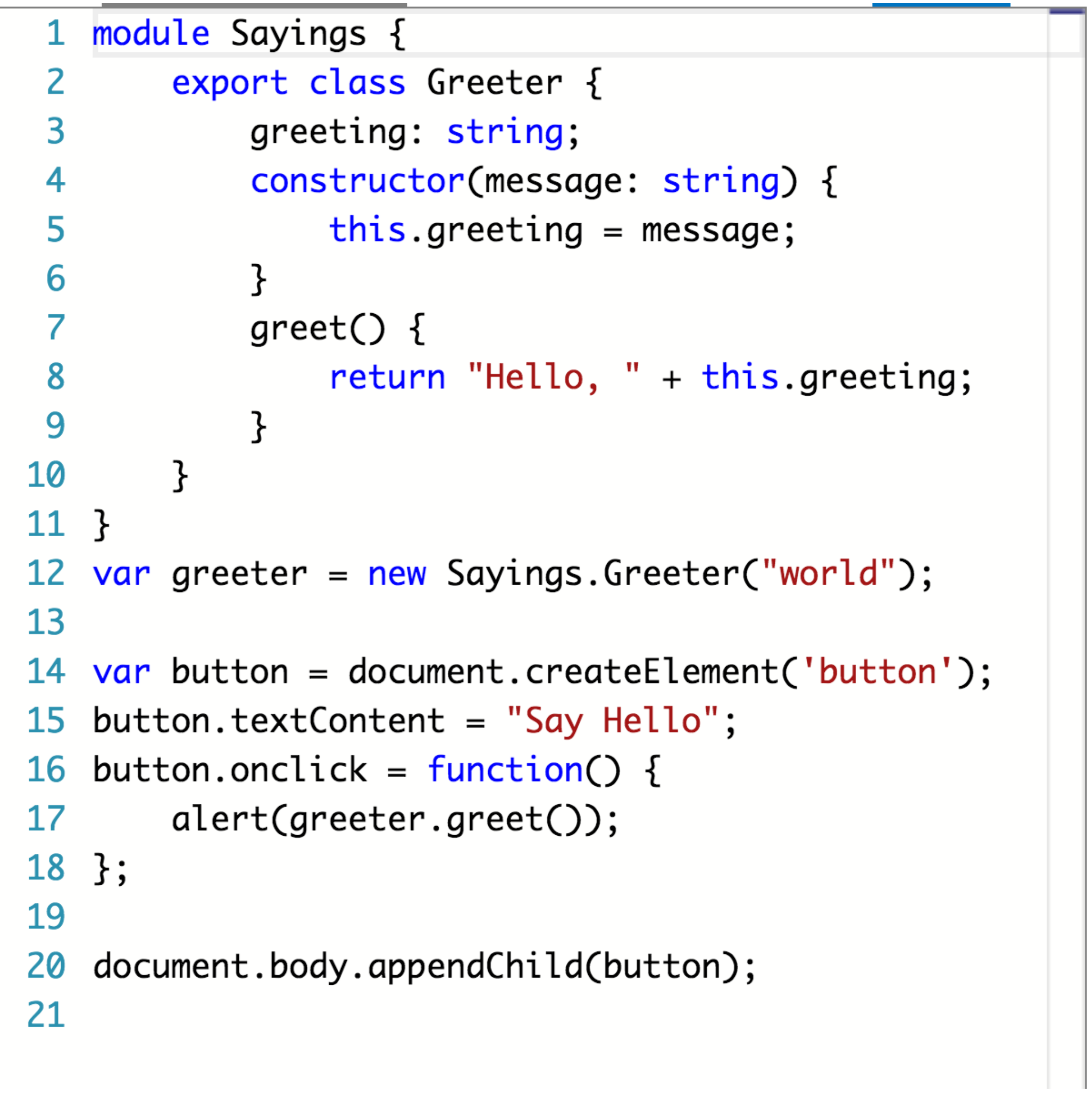
TypeScript



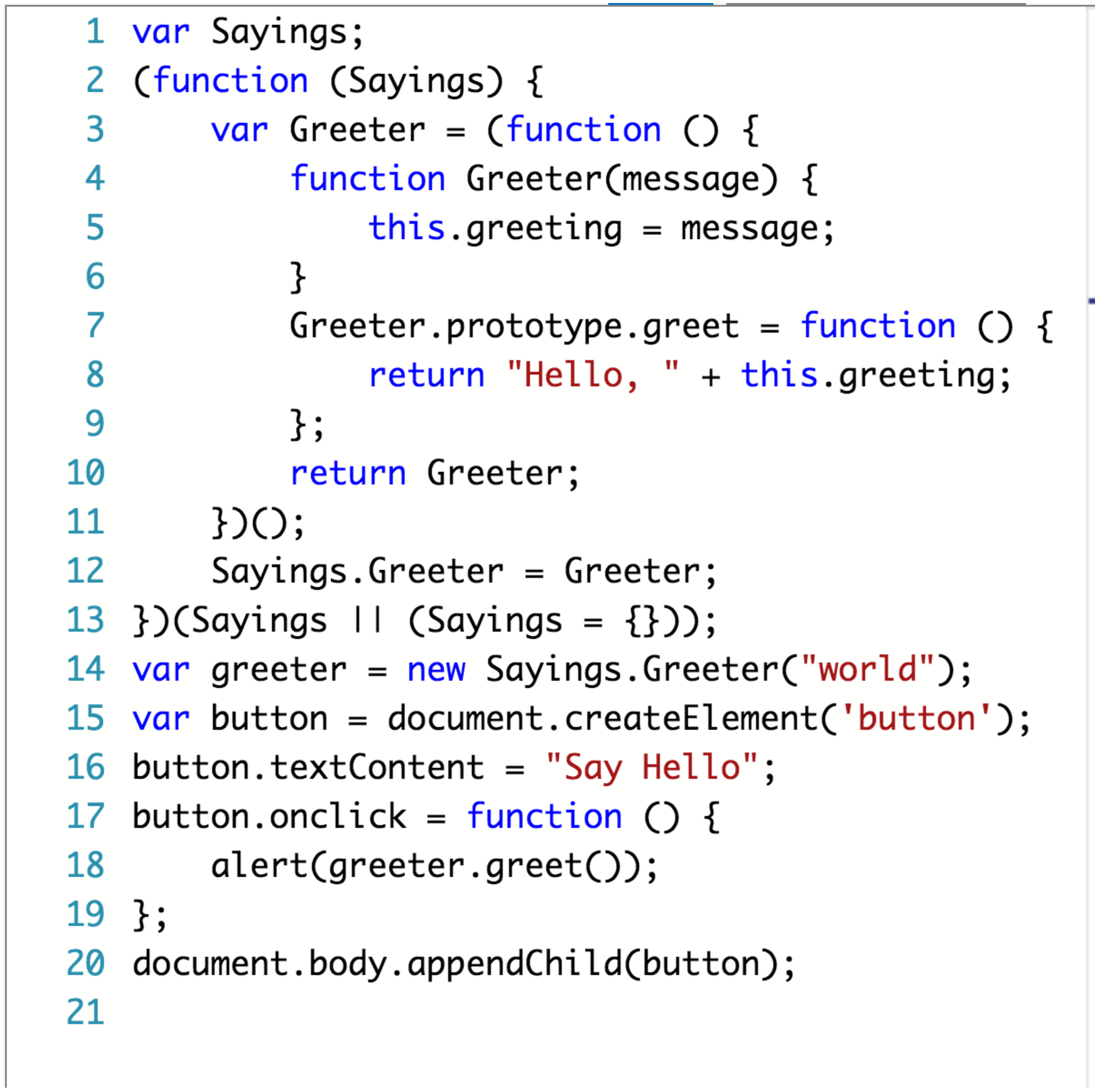
```
var Greeter = (function () {
    function Greeter(message) {
        this.greeting = message;
    }
    Greeter.prototype.greet = function () {
        return "Hello, " + this.greeting;
    };
    return Greeter;
})();
var greeter = new Greeter("world");
var button = document.createElement('button');
button.textContent = "Say Hello";
button.onclick = function () {
    alert(greeter.greet());
};
document.body.appendChild(button);
```

```typescript
class Animal {
    constructor(public name: string) { }
    move(meters: number) {
        alert(this.name + " moved " + meters + "m.");
    }
}

class Snake extends Animal {
    constructor(name: string) { super(name); }
    move() {
        alert("Slithering...");
        super.move(5);
    }
}

class Horse extends Animal {
    constructor(name: string) { super(name); }
    move() {
        alert("Galloping...");
        super.move(45);
    }
}

var sam = new Snake("Sammy the Python");
var tom: Animal = new Horse("Tommy the Palomino");

sam.move();
tom.move(34);

```

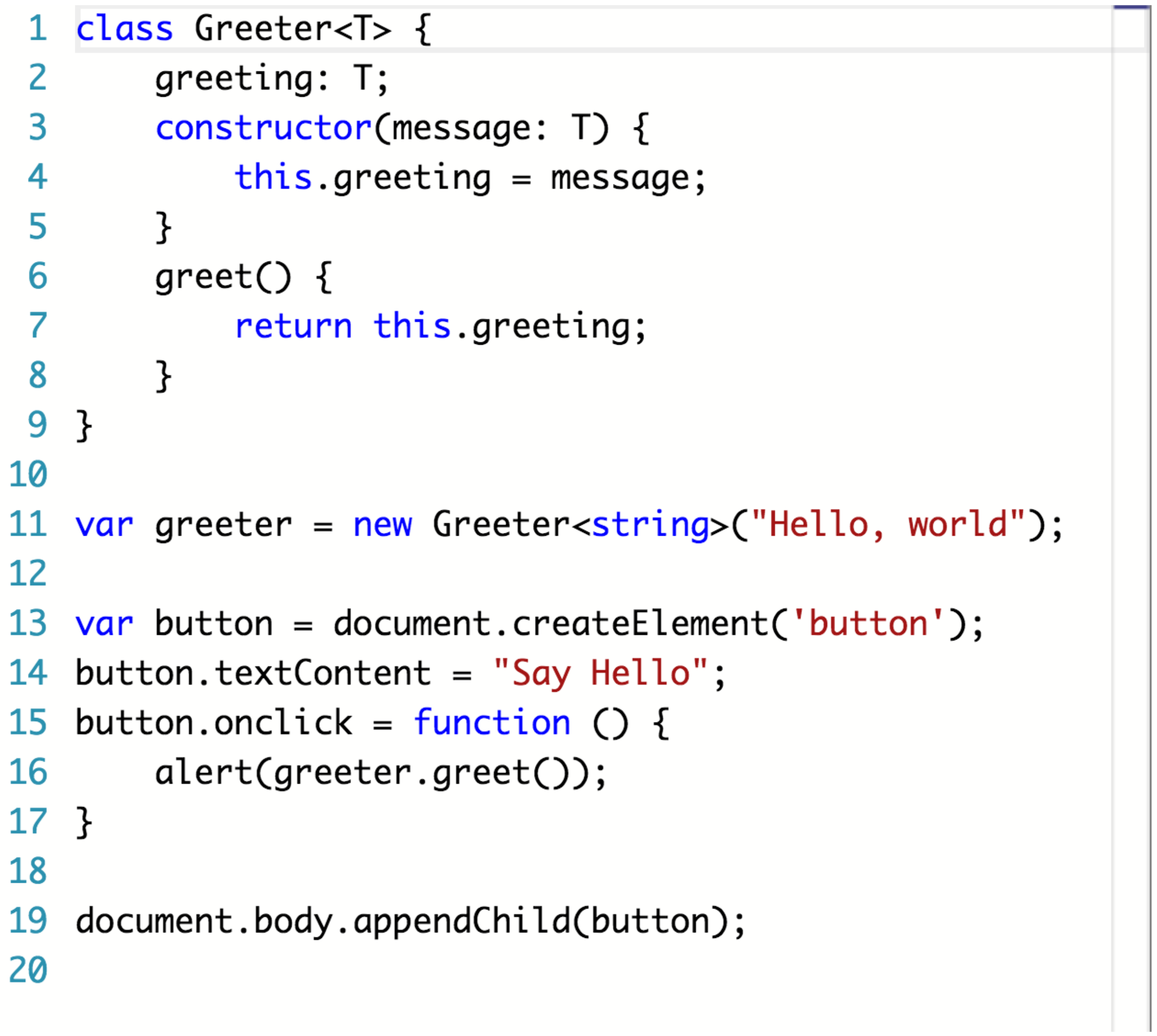
```javascript
var __extends = (this && this.__extends) || function (d, b) {
    for (var p in b) if (b.hasOwnProperty(p)) d[p] = b[p];
    function __() { this.constructor = d; }
    d.prototype = b === null ? Object.create(b) : (__.prototype = b.prototype, new __());
};
var Animal = (function () {
    function Animal(name) {
        this.name = name;
    }
    Animal.prototype.move = function (meters) {
        alert(this.name + " moved " + meters + "m.");
    };
    return Animal;
})();
var Snake = (function (_super) {
    __extends(Snake, _super);
    function Snake(name) {
        _super.call(this, name);
    }
    Snake.prototype.move = function () {
        alert("Slithering...");
        _super.prototype.move.call(this, 5);
    };
    return Snake;
})(Animal);
var Horse = (function (_super) {
    __extends(Horse, _super);
    function Horse(name) {
        _super.call(this, name);
    }
    Horse.prototype.move = function () {
        alert("Galloping...");
        _super.prototype.move.call(this, 45);
    };
    return Horse;
})(Animal);
var sam = new Snake("Sammy the Python");
var tom = new Horse("Tommy the Palomino");
sam.move();
tom.move(34);

```



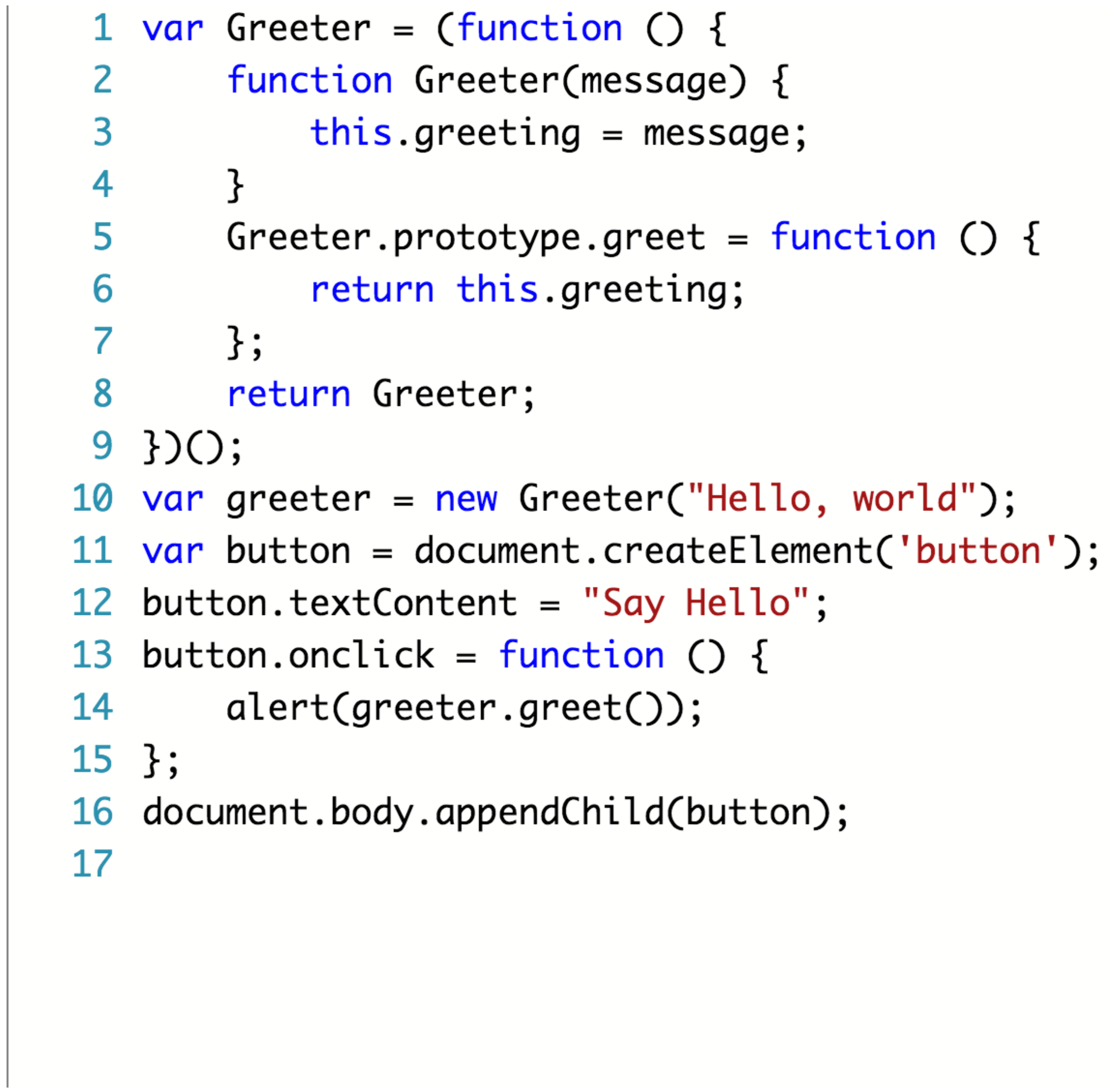
```typescript
module Sayings {
    export class Greeter {
        greeting: string;
        constructor(message: string) {
            this.greeting = message;
        }
        greet() {
            return "Hello, " + this.greeting;
        }
    }
}
var greeter = new Sayings.Greeter("world");

var button = document.createElement('button');
button.textContent = "Say Hello";
button.onclick = function() {
    alert(greeter.greet());
};

document.body.appendChild(button);

```



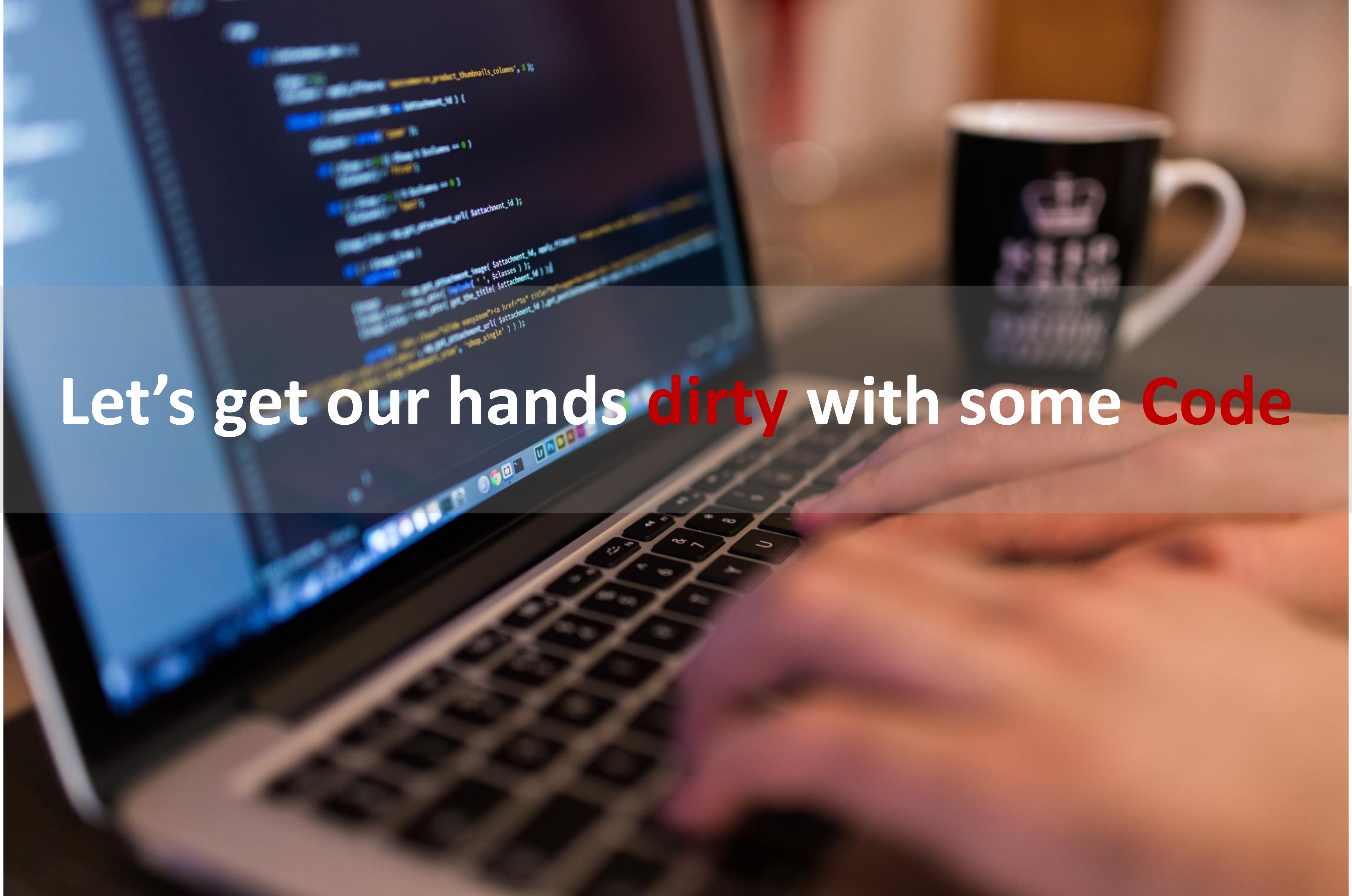
```javascript
var Sayings;
(function (Sayings) {
    var Greeter = (function () {
        function Greeter(message) {
            this.greeting = message;
        }
        Greeter.prototype.greet = function () {
            return "Hello, " + this.greeting;
        };
        return Greeter;
    })();
    Sayings.Greeter = Greeter;
})(Sayings || (Sayings = {}));
var greeter = new Sayings.Greeter("world");
var button = document.createElement('button');
button.textContent = "Say Hello";
button.onclick = function () {
    alert(greeter.greet());
};
document.body.appendChild(button);

```



```
class Greeter<T> {
    greeting: T;
    constructor(message: T) {
        this.greeting = message;
    }
    greet() {
        return this.greeting;
    }
}

var greeter = new Greeter<string>("Hello, world");

var button = document.createElement('button');
button.textContent = "Say Hello";
button.onclick = function () {
    alert(greeter.greet());
}

document.body.appendChild(button);

```



```
var Greeter = (function () {
    function Greeter(message) {
        this.greeting = message;
    }
    Greeter.prototype.greet = function () {
        return this.greeting;
    };
    return Greeter;
})();
var greeter = new Greeter("Hello, world");
var button = document.createElement('button');
button.textContent = "Say Hello";
button.onclick = function () {
    alert(greeter.greet());
};
document.body.appendChild(button);

```



Let’s get our hands dirty with some Code