Building Next Generation Apps with TypeScript

Rachel Appel
Appel Consulting
http://RachelAppel.com
http://Twitter.com/RachelAppel
http://Linkedin.com/rachelappel

Agenda

- TypeScript Intro
- Tools
- Compilation
- Types
- Classes
- Inheritance
- Debugging
- Web and Apps

Get TypeScript

http://www.typescriptlang.org

TypeScript PREVIEW

TypeScript in the Wild

- Bing
- Monaco

TypeScript Features

Superset of JavaScript

ES5 / ES6

Type Checking

Visual Studio
Webstorm
Browser tools

Next-gen JavaScript for the enterprise

Syntactic Sugar

Generics & Arrow Functions

Classes

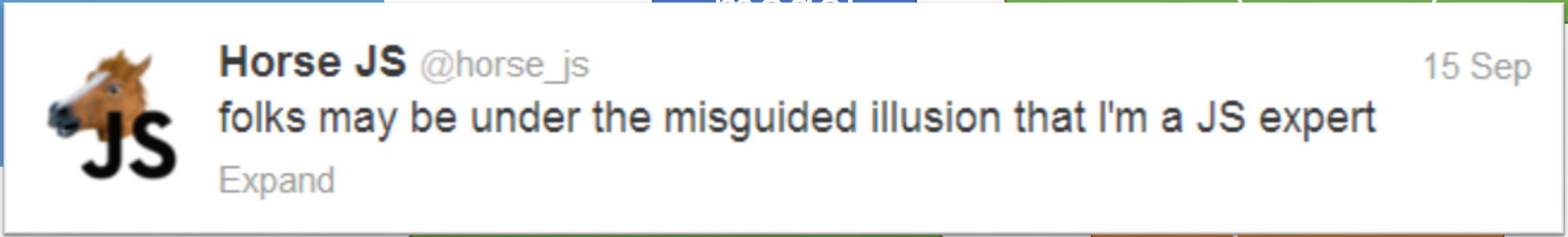
Dot notation for properties

Why use TypeScript?

JavaScript was designed for a

SPA App model

More and more server side logic using JavaScript (Node.JS)



Horse JS @horse_js 15 Sep

folks may be under the misguided illusion that I'm a JS expert

Expand

are becoming quite complex

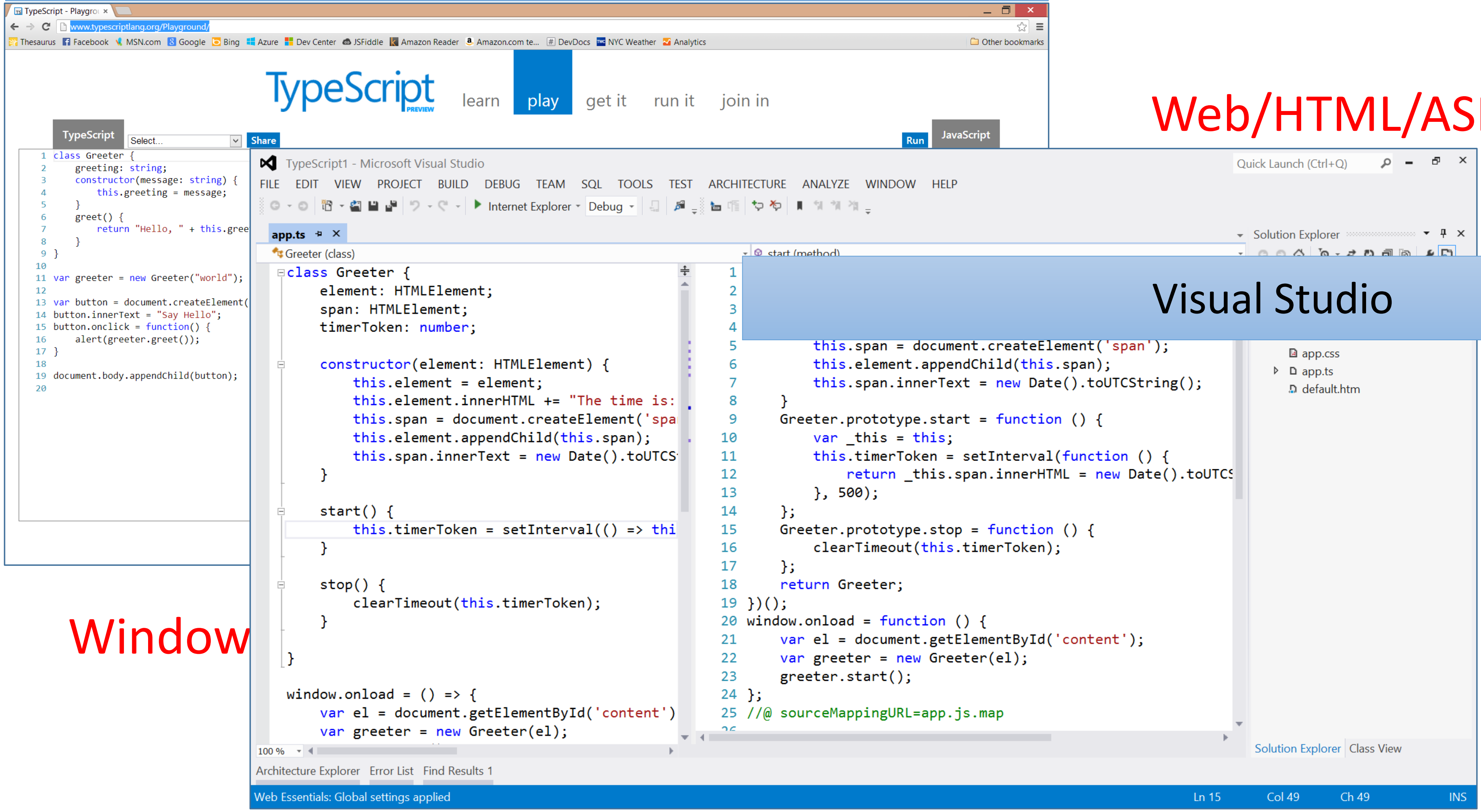
OOP is easier

C# coders forced into JavaScript

TypeScript PREVIEW

http://www.typescriptlang.org/playground

Web/HTML/ASP.NET



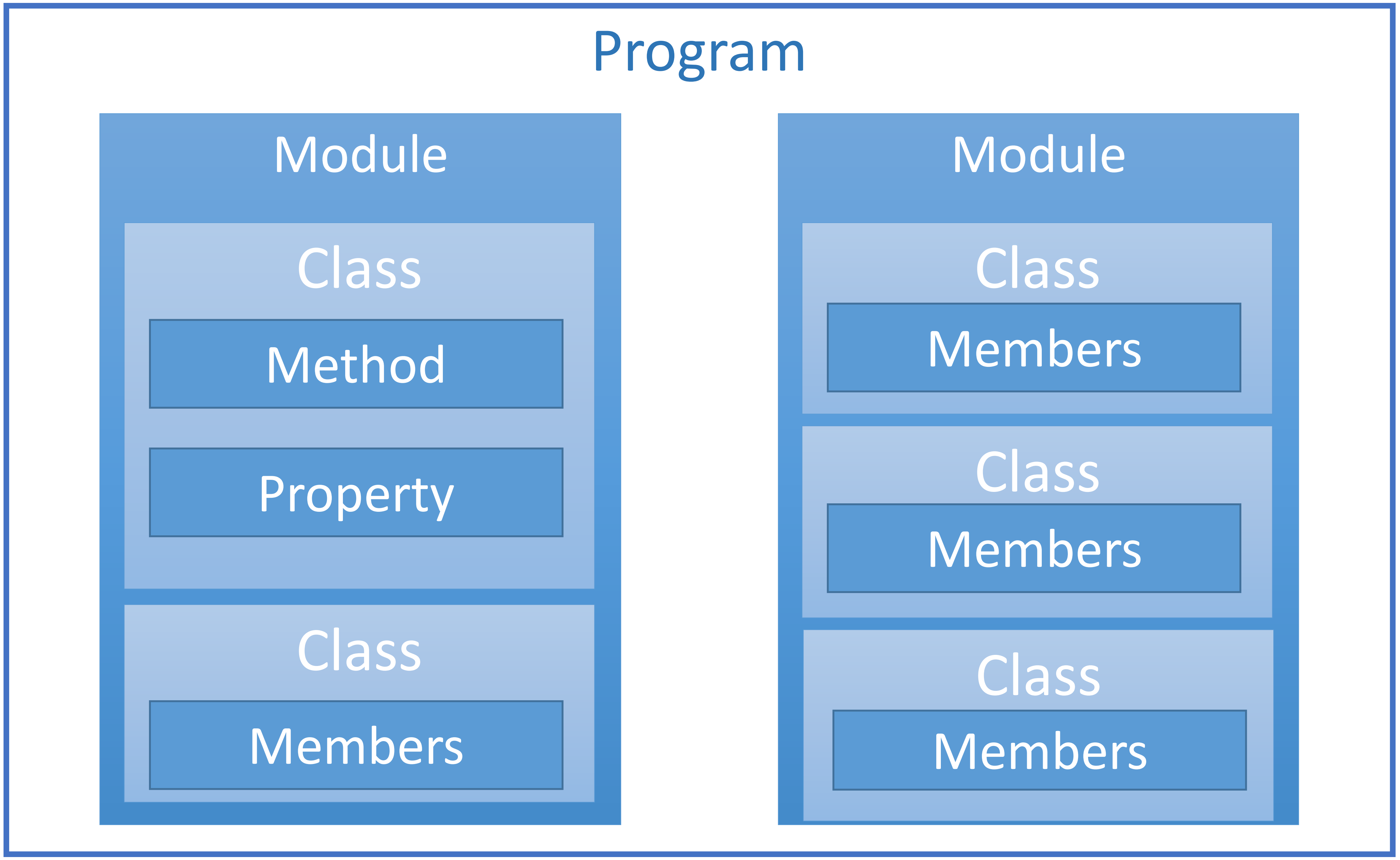
```
class Greeter {
    greeting: string;
    constructor(message: string) {
        this.greeting = message;
    }
    greet() {
        return "Hello, " + this.gree
    }
}

var greeter = new Greeter("world");

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

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

```
class Greeter {
    element: HTMLElement;
    span: HTMLElement;
    timerToken: number;

    constructor(element: HTMLElement) {
        this.element = element;
        this.element.innerHTML += "The time is:
        this.span = document.createElement('spa
        this.element.appendChild(this.span);
        this.span.innerText = new Date().toUTCS
    }

    start() {
        this.timerToken = setInterval(() => thi
    }

    stop() {
        clearTimeout(this.timerToken);
    }

}

window.onload = () => {
    var el = document.getElementById('content')
    var greeter = new Greeter(el);
```

```
        this.span = document.createElement('span');
        this.element.appendChild(this.span);
        this.span.innerText = new Date().toUTCString();
    }
    Greeter.prototype.start = function () {
        var _this = this;
        this.timerToken = setInterval(function () {
            return _this.span.innerHTML = new Date().toUTCS
        }, 500);
    };
    Greeter.prototype.stop = function () {
        clearTimeout(this.timerToken);
    };
    return Greeter;
})();
window.onload = function () {
    var el = document.getElementById('content');
    var greeter = new Greeter(el);
    greeter.start();
};
//@ sourceMappingURL=app.js.map
```

Visual Studio

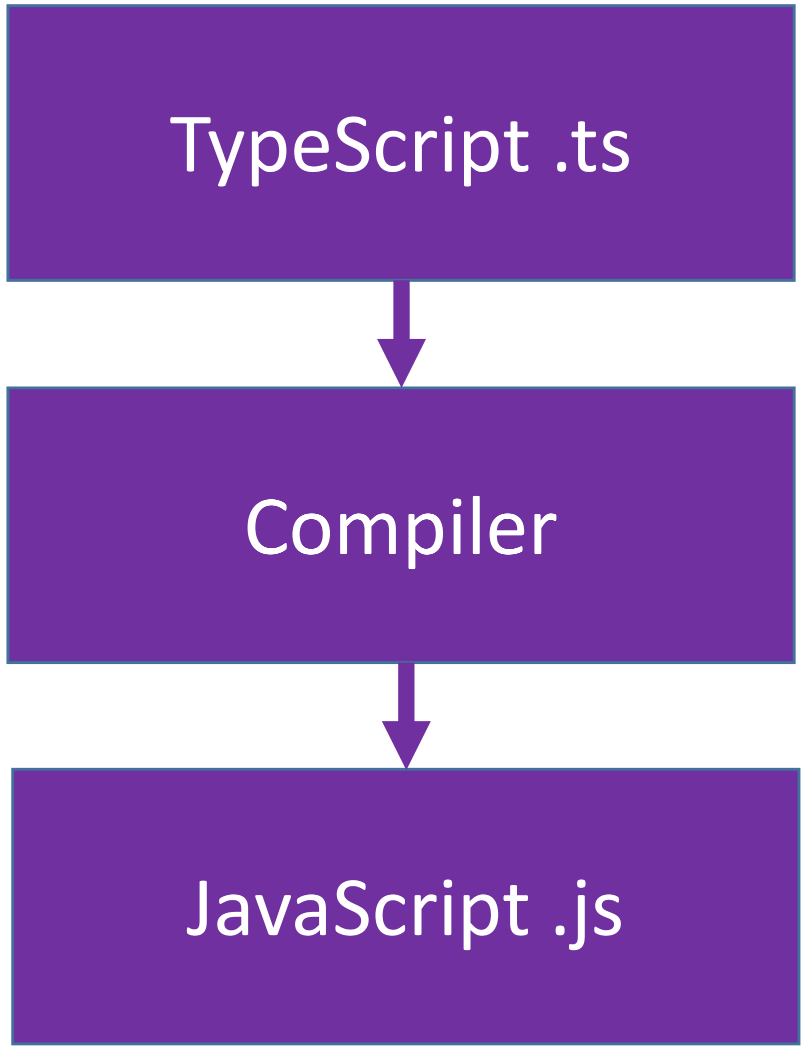
Window


Program
Module
Class
Method
Property
Class
Members
Module
Class
Members
Class
Members
Class
Members

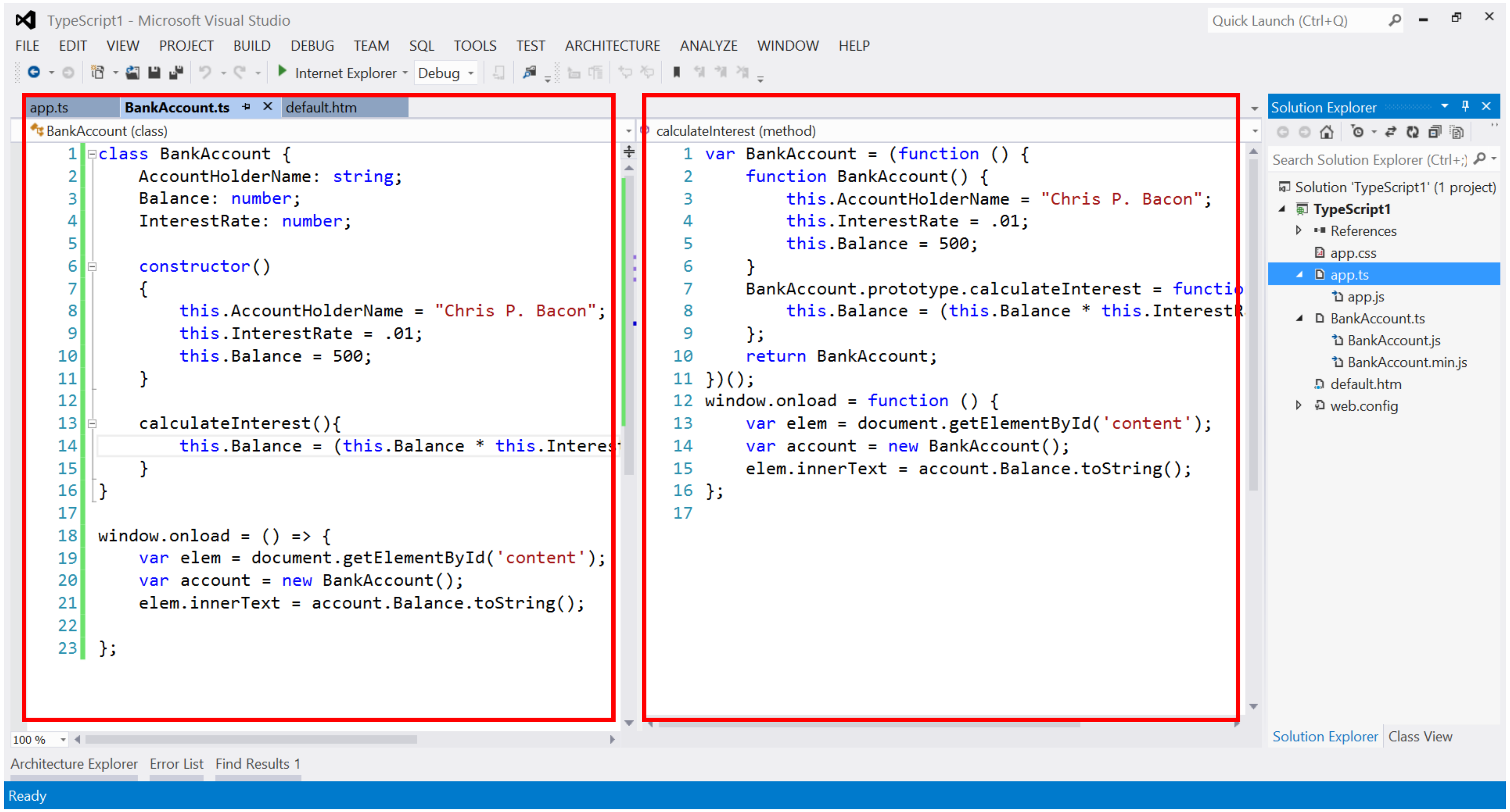
DEMO

- TypeScript programs in Visual Studio

TypeScript Compilation



TypeScript compilation



```
class BankAccount {
    AccountHolderName: string;
    Balance: number;
    InterestRate: number;

    constructor()
    {
        this.AccountHolderName = "Chris P. Bacon";
        this.InterestRate = .01;
        this.Balance = 500;
    }

    calculateInterest(){
        this.Balance = (this.Balance * this.Interes
    }
}

window.onload = () => {
    var elem = document.getElementById('content');
    var account = new BankAccount();
    elem.innerText = account.Balance.toString();

};
```

```
var BankAccount = (function () {
    function BankAccount() {
        this.AccountHolderName = "Chris P. Bacon";
        this.InterestRate = .01;
        this.Balance = 500;
    }
    BankAccount.prototype.calculateInterest = functio
        this.Balance = (this.Balance * this.InterestR
    };
    return BankAccount;
})();
window.onload = function () {
    var elem = document.getElementById('content');
    var account = new BankAccount();
    elem.innerText = account.Balance.toString();
};
```

Types

- Primitive and Object
- Any
- Number
- Boolean
- String
- Null *
- Undefined *
- Object
- Void *
- HTMLElement
- Functions
- Enum

Type annotations

- Argument types
- Return types
- Type inference

DEMO

- Types and annotations

Classes

```
class BankAccount {}
```

TS

```
var BankAccount = (function () {
    function BankAccount() { }
    return BankAccount;
})();
```

JS

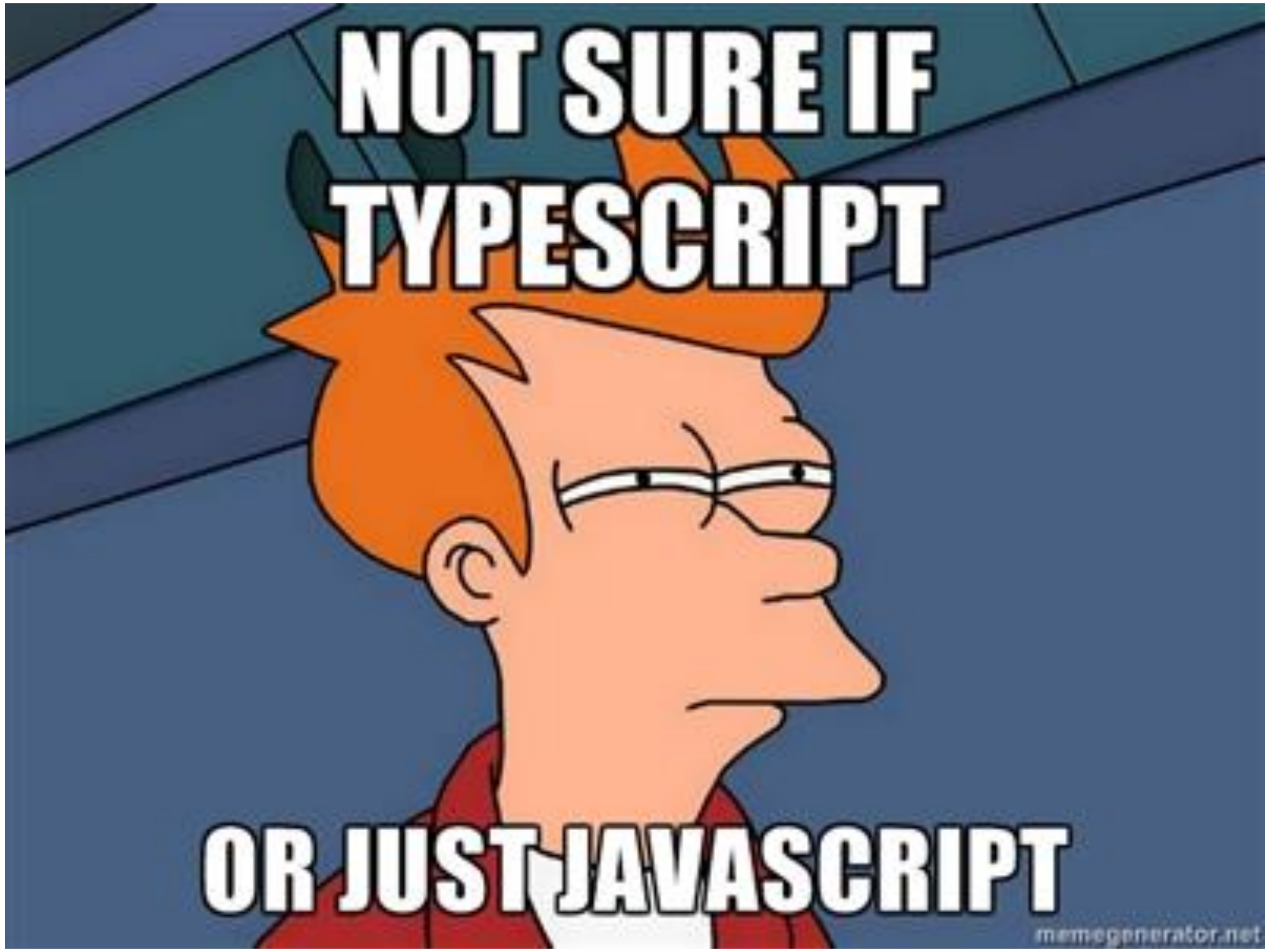
Classes: Constructors

```
constructor()
{
    this.AccountHolderName = "Chris P. Bacon";
    this.InterestRate = .01;
    this.Balance = 500;
}
```

TS

```
function BankAccount() {
    this.AccountHolderName = "Chris P. Bacon";
    this.InterestRate = .01;
    this.Balance = 500;
}
```

JS


NOT SURE IF TYPESCRIPT
OR JUST JAVASCRIPT
memegenerator.net

Classes: Members

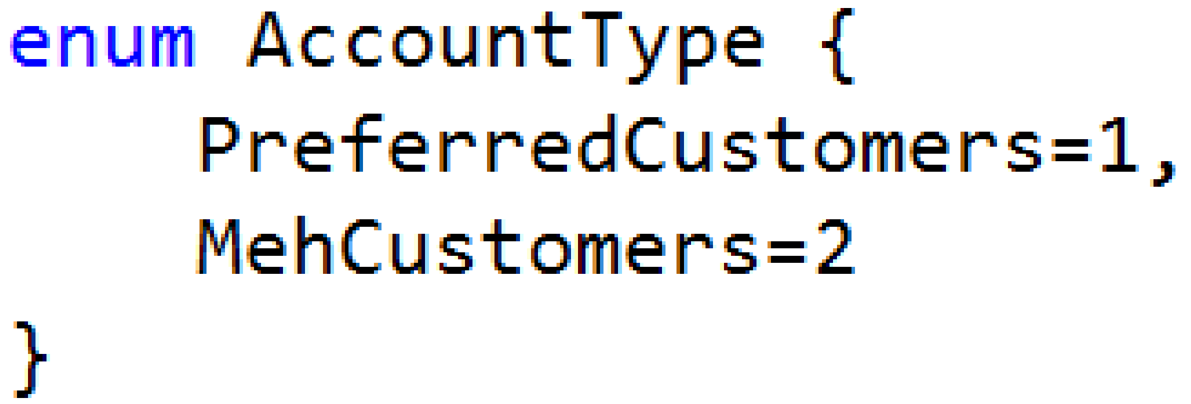
```
deposit(amount: number) {
    this.Balance += amount;
}

calculateInterest() : number {
    this.Balance = (this.Balance * this.InterestRate);
    return this.Balance;
}
```

TS

```
BankAccount.prototype.deposit = function (amount) {
    this.Balance += amount;
};
BankAccount.prototype.calculateInterest = function () {
    this.Balance = (this.Balance * this.InterestRate);
    return this.Balance;
};
```

JS

Enums



```
enum AccountType {
    PreferredCustomers=1,
    MehCustomers=2
}
```

Classes: Access modifiers

```
class BankAccount {
    public AccountHolderName: string;
    public Balance: number;
    private InterestRate: number;

    public deposit(amount: number) {
        this.Balance = this.Balance + amount;
    }

    public calculateInterest() : number {
        this.Balance = (this.Balance * this.InterestRate);
        return this.Balance;
    }
}
```

Accessing a Class and its members

```
window.onload = () => {
    var elem = document.getElementById('content');
    var account = new BankAccount();
    account.deposit(500);
    elem.innerText = account.Balance.toString();
};
```

DEMO

- Creating and using a class

Inheritance

```
class CheckingAccount extends BankAccount {}
```

TS

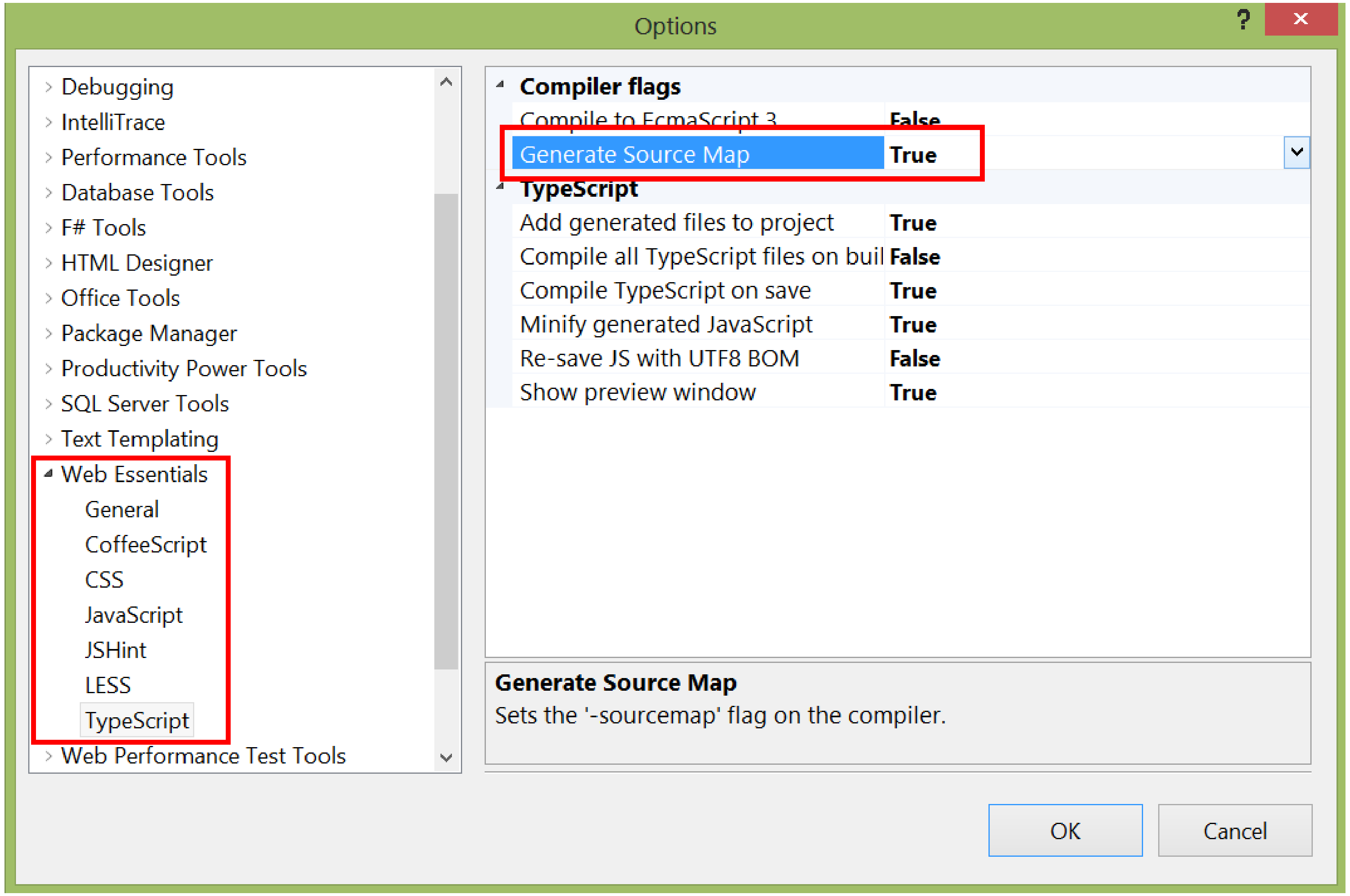
```
var CheckingAccount = (function (_super) {
    __extends(CheckingAccount, _super);
    function CheckingAccount() {
        _super.apply(this, arguments);

    }
    return CheckingAccount;
})(BankAccount);
```

JS

DEMO

- Inheritance

Debugging TypeScript



Compiler flags	
Compile to EcmaScript 3	False
Generate Source Map	True
TypeScript	
Add generated files to project	True
Compile all TypeScript files on buil	False
Compile TypeScript on save	True
Minify generated JavaScript	True
Re-save JS with UTF8 BOM	False
Show preview window	True

Generate Source Map
Sets the '-sourcemap' flag on the compiler.

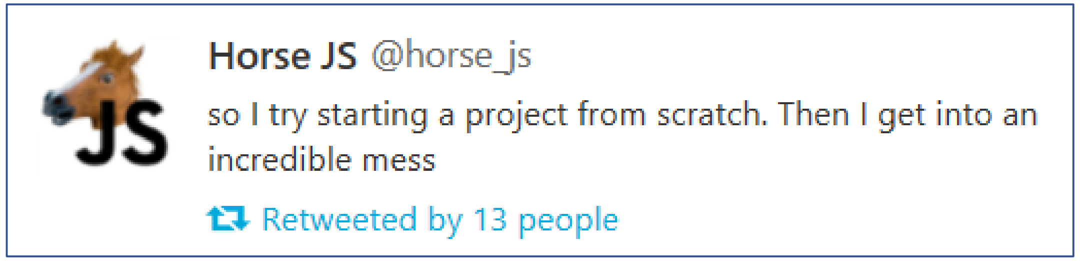
DEMO

- Debugging

Web and Apps

- HTML
- ASP.NET WebForms and ASP.NET MVC
- Windows Store JavaScript apps

http://bit.ly/15iOfFt



Horse JS @horse_js

so I try starting a project from scratch. Then I get into an incredible mess

Retweeted by 13 people

Thank You!

Rachel Appel

rachel@rachelappel.com

http://RachelAppel.com

http://Twitter.com/RachelAppel

http://LinkedIn.com/RachelAppel

http://bit.ly/RachNOW Use code APPEL-13 for a 14 day free trial!