



International  
**JavaScript**  
Conference

# One language to rule them all: TypeScript

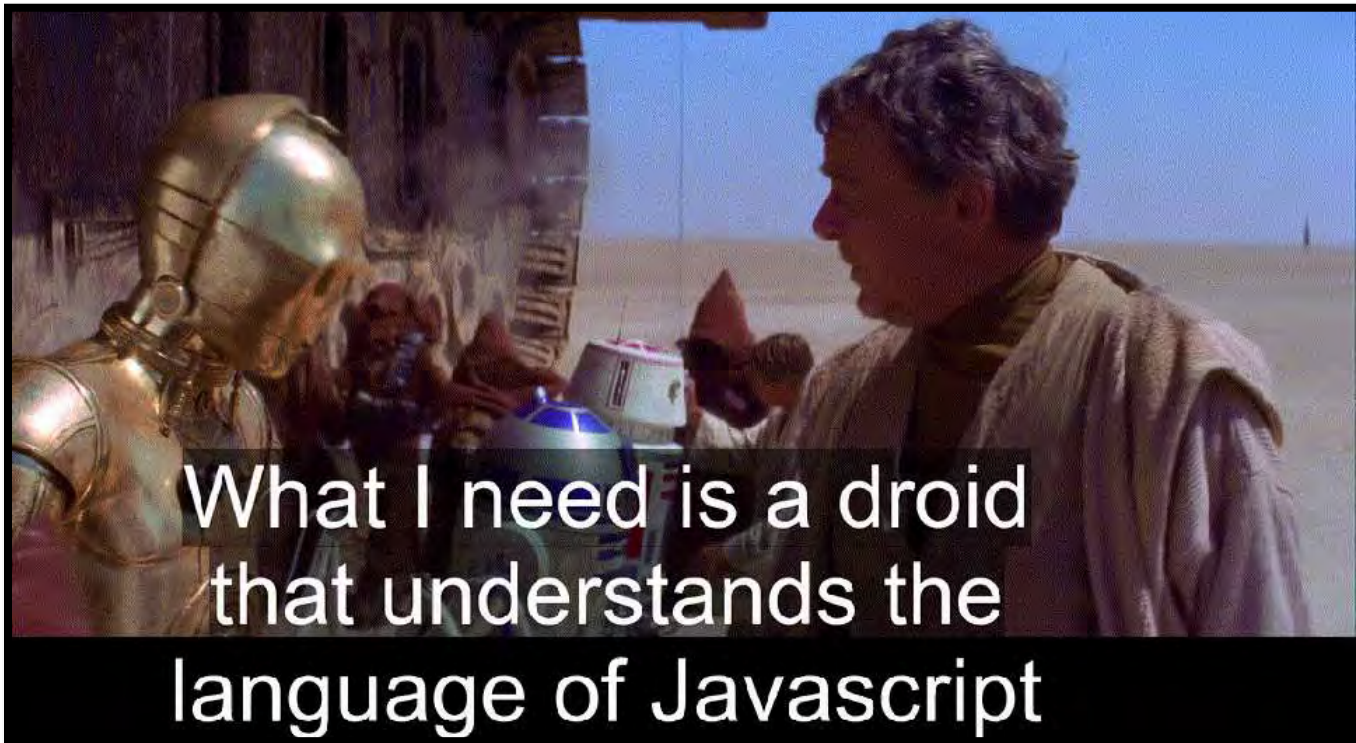
Gil Fink

sparXys CEO

@gilfink / [www.gilfink.net](http://www.gilfink.net)



# Why TypeScript?





# Interoperability

It's a JavaScript super set



# Features from the near future

Before they are part of ECMAScript spec



Static typing and type inference

Now that we know why  
TypeScript...



Let's get started



# About Me

- sparXys CEO and senior consultant
- Google Web Technologies GDE & Microsoft MVP
- Pro Single Page Application Development (Apress) co-author
- 4 Microsoft Official Courses (MOCs) co-author



sparXys



# Agenda

- TypeScript Everywhere
- TypeScript syntax and language features
- Building a simple end-to-end app with TypeScript

# TypeScript Everywhere

CROSS PLATFORM



---

TypeScript



Devices + IoT



Client



Server

ACROSS DEVICES

# Hello TypeScript and React

Demo

# Some TypeScript Key Features

**Support  
standard  
JavaScript code  
with static  
typing**

**Encapsulation  
through classes  
and modules**

**Support for  
constructors,  
properties and  
functions**

**Interfaces and  
enums support**

**Lambda and  
generics  
support**

**Intellisense and  
syntax checking**

# From TypeScript to JavaScript



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

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

The diagram shows the transformation of TypeScript code into JavaScript code. The TypeScript code on the left is a class named `Greeter` with a `greeting` property, a `constructor` that takes a `message` parameter and sets `this.greeting`, and a `greet` method that returns a string. The JavaScript code on the right is the compiled output, which uses a function factory pattern to create a `Greeter` function. The `greet` method is attached to the `Greeter` prototype. The transformation is labeled `tsc.js` in red.

# Building a Simple End-to-End App with TypeScript, React, Express and CouchDB

Demo

# Ideas to go from here

- Wrap in Electron (desktop app)
- Wrap in Cordova (make it a hybrid app)
- Add PWA capabilities
- Deploy to the cloud
- And more 😊



# Summary

- Open source language that compiles into JavaScript
- Key features:
  - Code encapsulation
  - Maintainable code
  - Tooling support
- Learn TypeScript today!

# Resources

- TypeScript – <http://www.typescriptlang.org>
- TypeScript Source Code - <https://github.com/Microsoft/TypeScript>
  
- My Website – <http://www.gilfink.net>
- Follow me on Twitter – @gilfink

Thank You!