



- 🎤 #1: Dan Vanderkam will talk about type declarations and @types.
- 🎤 #2: Orta Therox will explain ALL the compiler flags.*
- 🎤 #3: Lightning talks. This could be you!

WIFI: GoogleGuest

*not all, the amount is too damn high



*Thanks @erindepew
for the logo*



Your organizers: Dan, Jason, Kirill and Orta

Meetup Format

Three 10–20 minute talks: Beginner, Intermediate, Advanced

Short Q&A

Socialize afterwards at Brass Monkey (55 Little W 12th St, New York, NY)

Please use #tsnyc on Twitter, Instagram, etc.

We follow the JSConf Code of Conduct



Talk: type declarations and @types

Dan Vanderkam



Break



Talk: compiler flags

Orta Therox



@ORTA

DANGER

The image shows a gallery installation of Richard Serra's artwork 'Betwixt the Torus and the Sphere'. It consists of several large, curved, rusted metal plates standing upright on a light-colored floor. The plates are arranged in a way that they create a sense of depth and perspective, with some plates in the foreground and others receding into the background. The lighting is bright, highlighting the texture and color of the metal. The ceiling features a white truss structure with recessed lighting.

Make art as popular as music

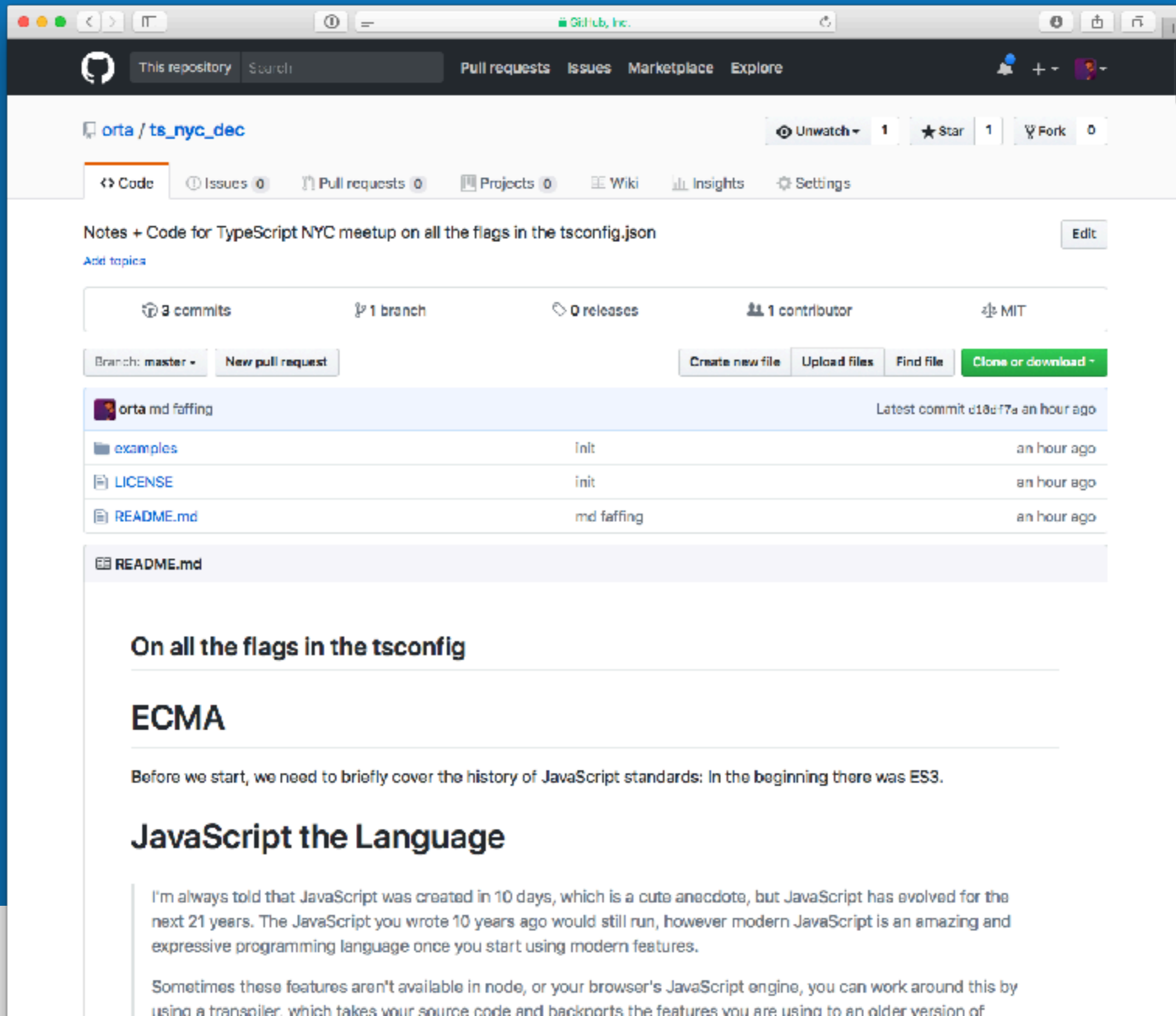


Bring all of the world's art online

What is inside a TSConfig?

```
{
  "compilerOptions": {
    "target": "es5",
    "module": "commonjs",
    "lib": [],
    "allowJs": true,
    "checkJs": true,
    "jsx": "preserve",
    "declaration": true,
    "sourceMap": true,
    "outFile": "./",
    "outDir": "./dist",
    "rootDir": "./",
    "removeComments": true,
    "noEmit": true,
    "importHelpers": true,
    "downlevelIteration": true,
    "isolatedModules": true,
    "strict": true,
    "noImplicitAny": true,
    "strictNullChecks": true,
    "strictFunctionTypes": true,
    "noImplicitThis": true,
    "alwaysStrict": true,
    "noUnusedLocals": true,
    "noUnusedParameters": true,
    "noImplicitReturns": true,
    "noFallthroughCasesInSwitch": true,
    "moduleResolution": "node",
    "baseUrl": "/"
  },
  /* Specify ECMAScript target version: 'ES3' (default), 'ES5', 'ES2015', 'ES2016', 'ES2017',
  /* Specify module code generation: 'none', 'commonjs', 'amd', 'system', 'umd', 'es2015', or
  /* Specify library files to be included in the compilation: */
  /* Allow javascript files to be compiled. */
  /* Report errors in .js files. */
  /* Specify JSX code generation: 'preserve', 'react-native', or 'react'. */
  /* Generates corresponding '.d.ts' file. */
  /* Generates corresponding '.map' file. */
  /* Concatenate and emit output to single file. */
  /* Redirect output structure to the directory. */
  /* Specify the root directory of input files. Use to control the output directory structure
  /* Do not emit comments to output. */
  /* Do not emit outputs. */
  /* Import emit helpers from 'tslib'. */
  /* Provide full support for iterables in 'for-of', spread, and destructuring when targeting
  /* Transpile each file as a separate module (similar to 'ts.transpileModule'). */
  /* Enable all strict type-checking options. */
  /* Raise error on expressions and declarations with an implied 'any' type. */
  /* Enable strict null checks. */
  /* Enable strict checking of function types. */
  /* Raise error on 'this' expressions with an implied 'any' type. */
  /* Parse in strict mode and emit "use strict" for each source file. */
  /* Report errors on unused locals. */
  /* Report errors on unused parameters. */
  /* Report error when not all code paths in function return a value. */
  /* Report errors for fallthrough cases in switch statement. */
  /* Specify module resolution strategy: 'node' (Node.js) or 'classic' (TypeScript pre-1.6).
  /* Base directory to resolve non-absolute module names. */
}
```

orta/ts_nyc_dec



This repository Search Pull requests Issues Marketplace Explore

orta / ts_nyc_dec Unwatch 1 Star 1 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Notes + Code for TypeScript NYC meetup on all the flags in the tsconfig.json Edit

Add topics

3 commits 1 branch 0 releases 1 contributor MIT

Branch: master - New pull request Create new file Upload files Find file Clone or download -

File	Commit	Time
examples	init	an hour ago
LICENSE	init	an hour ago
README.md	md faffing	an hour ago

README.md

On all the flags in the tsconfig

ECMA

Before we start, we need to briefly cover the history of JavaScript standards: In the beginning there was ES3.

JavaScript the Language

I'm always told that JavaScript was created in 10 days, which is a cute anecdote, but JavaScript has evolved for the next 21 years. The JavaScript you wrote 10 years ago would still run, however modern JavaScript is an amazing and expressive programming language once you start using modern features.

Sometimes these features aren't available in node, or your browser's JavaScript engine, you can work around this by using a transpiler, which takes your source code and backports the features you are using to an older version of

Need to know: ECMA

Need to know: ECMA

*ES3

Need to know: ECMA

*ES3

*ES6

Need to know: ECMA

*ES3

*ES6

*ES2015

Need to know: ECMA

*ES3

*ES6

*ES2015

*Stages

configs

keys

```
target
module
lib
allowJs
checkJs
jsx
declaration
sourceMap
outFile
outDir
rootDir
removeComments
noEmit
importHelpers
downlevelIteration
isolatedModules
strict
noImplicitAny
strictNullChecks

strictFunctionTypes
noImplicitThis
alwaysStrict
noUnusedLocals
noUnusedParameters
noImplicitReturns
noFallthroughCasesInSwitch
moduleResolution
baseUrl
paths
rootDirs
typeRoots
types
allowSyntheticDefaultImports
preserveSymlinks
sourceRoot
mapRoot
inlineSourceMap
inlineSources

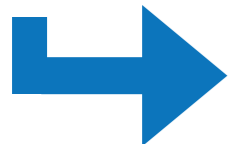
experimentalDecorators
emitDecoratorMetadata
```

Environment

Target

TypeScript:

```
export const helloWorld = "Hi"
```



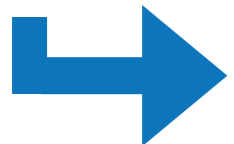
Exported JavaScript with "commonjs"

```
"use strict"  
exports.__esModule = true  
exports.helloWorld = "Hi"
```

Target

TypeScript:

```
export const helloWorld = "Hi"
```



Exported JavaScript with "commonjs"

```
"use strict"  
exports.__esModule = true  
exports.helloWorld = "Hi"
```

Exported JavaScript with "es6"

```
export var helloWorld = "Hi"
```

Lib

What is in your runtime environment

ES5, ES6, ES2015, ES7, ES2016, ES2017, ESNEXT, DOM, DOM.Iterable, WebWorker, ScriptHost, ES2015.Core, ES2015.Collection, ES2015.Generator, ES2015.Iterable, ES2015.Promise, ES2015.Proxy, ES2015.Reflect, ES2015.Symbol, ES2015.Symbol.WellKnown, ES2016.Array.Include, ES2017.object, ES2017.SharedMemory, ES2017.TypedArrays, esnext.asynciterable.

Lib

What is in your runtime environment?

ES5, ES6, ES2015, ES7, ES2016, ES2017, ESNNext, DOM, DOM.Iterable, WebWorker, ScriptHost, ES2015.Core, ES2015.Collection, ES2015.Generator, ES2015.Iterable, ES2015.Promise, ES2015.Proxy, ES2015.Reflect, ES2015.Symbol, ES2015.Symbol.WellKnown, ES2016.Array.Include, ES2017.object, ES2017.SharedMemory, ES2017.TypedArrays, esnext.asynciterable.

checkJs

```
def.js
```

```
export const pi = parseFloat(3.124)
```

```
// parseFloat(value:string)
```


checkJs

def.js

```
export const pi = parseFloat(3.124)
```

```
// parseFloat(value:string)
```

TypeScript - ex.ts

```
import { pi } from "./def"  
  
console.log(pi)
```

checkJs

def.js

```
export const pi = parseFloat(3.124)
```

```
// parseFloat(value:string)
```

TypeScript - ex.ts

```
import { pi } from "./def"  
  
console.log(pi)
```

With: Raises an error

```
def.js(1,30): error TS2345: Argument of type '3.124' is  
not assignable to parameter of type 'string'.
```

jsx

How to handle JSX files?

`preserve`, `react`, `react-native`

declaration

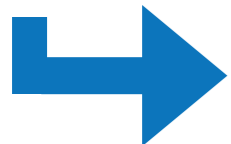
How to handle JSX files?

`preserve, react, react-native`

declaration

TypeScript: ex.ts

```
export const helloWorld = "Hi"
```



JavaScript: ex.js

```
"use strict"  
exports.__esModule = true  
exports.helloWorld = "Hi"
```

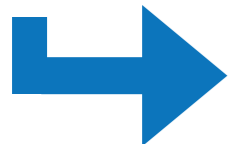
External Definition: ex.d.ts

```
export declare const helloWorld = "hi";
```

sourceMap

TypeScript: ex.ts

```
export const helloWorld = "Hi"
```



JavaScript: ex.js

```
"use strict"  
exports.__esModule = true  
exports.helloWorld = "Hi"  
//# sourceMappingURL=ex.js.map
```

External Definition: ex.js.map

```
{"version":3,"file":"ex.js","sourceRoot":"","sources":["..
```

File Management

outFile

Pack all the TypeScript into one file

ourDir

The folder to put the js in

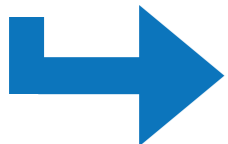
rootDir

The root of your TypeScript files

removeComments

TypeScript: ex.ts

```
/** Used to show the user a hello message */  
export const helloWorld = "hi";
```



JavaScript: with true

```
"use strict";  
Object.defineProperty(exports, "__esModule", { value: true });  
exports.helloWorld = "hi";
```

JavaScript: with false

```
"use strict";  
Object.defineProperty(exports, "__esModule", { value: true });  
/** Used to show the user a hello message */  
exports.helloWorld = "hi";
```

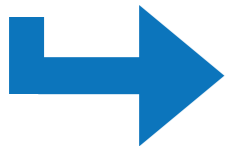
noEmit

Don't save to disk

importHelpers

TypeScript: ex.ts

```
export const helloWorld = { ... { hello: "world" } }
```



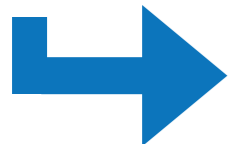
JavaScript: off (default)

```
"use strict";
var __assign = (this && this.__assign) || Object.assign || function(t) {
  for (var s, i = 1, n = arguments.length; i < n; i++) {
    s = arguments[i];
    for (var p in s) if (Object.prototype.hasOwnProperty.call(s, p))
      t[p] = s[p];
  }
  return t;
};
Object.defineProperty(exports, "__esModule", { value: true });
exports.helloWorld = __assign({ hello: "world" });
```

importHelpers

TypeScript: ex.ts

```
export const helloWorld = { ... { hello: "world" } }
```



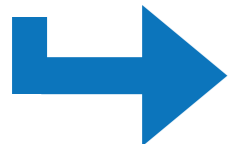
JavaScript: on

```
"use strict";  
Object.defineProperty(exports, "__esModule", { value: true });  
var tslib_1 = require("tslib");  
exports.helloWorld = tslib_1.__assign({ hello: "world" });
```

downlevelIteration

TypeScript: ex.ts

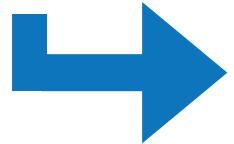
```
const helloWorld = () => {  
  for (const char of "Hello World") {  
    console.log(char)  
  }  
}
```



JavaScript: off (default)

```
var helloWorld = function () {  
  for (var _i = 0, _a = "Hello World"; _i < _a.length; _i++) {  
    var char = _a[_i];  
    console.log(char);  
  }  
};
```

downlevelIteration



JavaScript: on

```
"use strict";
var __values = (this && this.__values) || function (o) {
  var m = typeof Symbol === "function" && o[Symbol.iterator], i = 0;
  if (m) return m.call(o);
  return {
    next: function () {
      if (o && i ≥ o.length) o = void 0;
      return { value: o && o[i++], done: !o };
    }
  };
};
var helloWorld = function () {
  try {
    for (var _a = __values("Hello World"), _b = _a.next(); !_b.done; _b = _a.next()) {
      var char = _b.value;
      console.log(char);
    }
  }
  catch (e_1_1) { e_1 = { error: e_1_1 }; }
  finally {
    try {
      if (_b && !_b.done && (_c = _a.return)) _c.call(_a);
    }
    finally { if (e_1) throw e_1.error; }
  }
  var e_1, _c;
};
```


isolatedModules

Just ship my code damnit

isolatedModules

Just ship my code damnit

TSC knows best

noImplicitAny

Make my code sing

noImplicitAny

TypeScript

```
const myFunc = value => value * 2
```

Without: no errors

With: Raises an error

```
ex.ts(1,16): error TS7006: Parameter 'value' implicitly  
has an 'any' type.
```

strictNullChecks

TypeScript

```
const getUserAge = () : string | null => "32"  
  
// getUserAge could return null - but  
// parseInt only takes a string  
parseInt(getUserAge())
```

Without: no errors

With: Raises an error

```
ex.ts(5,10): error TS2345: Argument of type 'string | null' is not assignable to  
parameter of type 'string'.  
  Type 'null' is not assignable to type 'string'.
```

strictNullChecks

TypeScript

```
const getUserAge = () : string | null => "32"  
  
// getUserAge could return null - but  
// parseInt only takes a string  
parseInt(getUserAge())
```

Without: no errors

With: Raises an error

```
ex.ts(5,10): error TS2345: Argument of type 'string | null' is not assignable to  
parameter of type 'string'.  
  Type 'null' is not assignable to type 'string'.
```

strictFunctionTypes

Makes function params line up

noImplicitThis

TypeScript

```
let o = {  
  n: 101,  
  explicitThis: function (m: number) {  
    return m + this.n.length; // error, 'length' does not exist on 'number'  
  },  
};
```

Without: no errors

With: Raises an error

```
ex.ts(4,25): error TS2339: Property 'length' does not  
exist on type 'number'.
```

noImplicitThis

TypeScript

```
let o = {  
  n: 101,  
  explicitThis: function (m: number) {  
    return m + this.n.length; // error, 'length' does not exist on 'number'  
  },  
};
```

Without: no errors

With: Raises an error

```
ex.ts(4,25): error TS2339: Property 'length' does not  
exist on type 'number'.
```

alwaysStrict

Only work in strict mode

noUnusedLocals

TypeScript

```
const myFunc = () => {  
  const onething = 1  
  return "Hello"  
}
```

Without: no errors

With: Raises an error

```
ex.ts(2,9): error TS6133: 'onething' is declared but its  
value is never read.
```

noUnusedParameters

TypeScript

```
const myFunc = value => "Hi"
```

Without: no errors

With: Raises an error

```
ex.ts(1,16): error TS6133: 'value' is declared but its value is never read.
```

Experimental

experimentalDecorators

TypeScript

```
const track = (target: Object, propertyKey: string,  
descriptor: TypedPropertyDescriptor<any>)  
⇒ {  
  console.log("Analytics event")  
  return descriptor  
}  
  
class MyApp {  
  @track  
  method() { return "hello world" }  
}
```

emitDecoratorMetadata

TypeScript - off

```
var MyApp = /** @class */ (function () {  
    function MyApp() {  
    }  
    MyApp.prototype.method = function () { return "hello world"; };  
    __decorate([  
        track  
    ], MyApp.prototype, "method", null);  
    return MyApp;  
})();
```


emitDecoratorMetadata

TypeScript - on

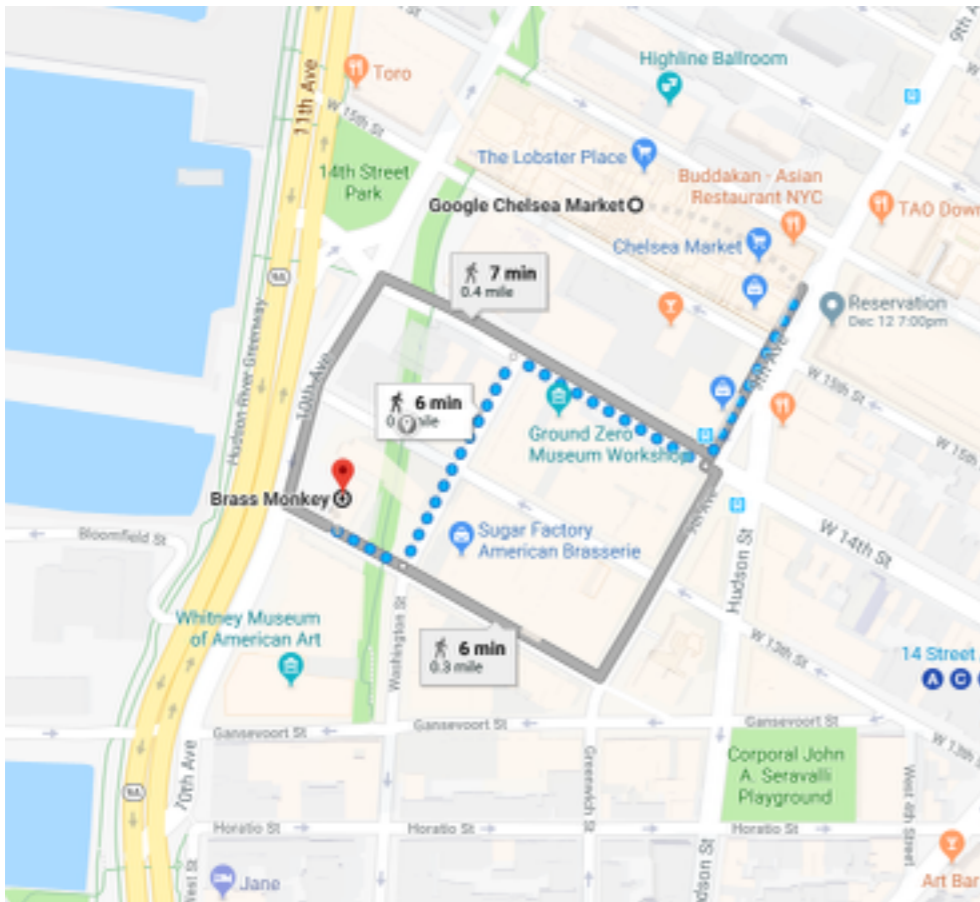
```
var MyApp = /** @class */ (function () {  
    function MyApp() {  
    }  
    MyApp.prototype.method = function () { return "hello world"; };  
    __decorate([  
        track,  
        __metadata("design:type", Function),  
        __metadata("design:paramtypes", []),  
        __metadata("design:returntype", void 0)  
    ], MyApp.prototype, "method", null);  
    return MyApp;  
})();
```



Lightning



Drinks & Discussions



Brass Monkey

55 Little W 12th St, New York, NY