TypeScript

will finally bring peace to your troubled soul

Tim Perry
@pimterry
Tech Lead & Open-Source Champion at Softwire


softwire

TypeScript

JavaScript + types

Chris Martin
@chris__martin

"Dynamic typing": The belief that you can't explain to a computer why your code works, but you can keep track of it all in your head.

1:15 AM - 10 Aug 2015 · San Mateo, CA, United States

925 742

```
navigator.geolocation.getCurrentPosition(function onSuccess(position) {
  var lat = position.latitude;
  var long = position.longitude;

  var lastUpdated = Date.parse(position.timestamp);
  var now = new Date();
  var positionIsCurrent = now.getYear() === lastUpdated.getYear();

  if (positionIsCurrent) {
    var div = document.createElement("div");
    div.class = "message";
    div.style = "width: 100%; height: 100px; background-color: red;";
    div.text = "Up to date position: " + lat + ", " + long;
    document.body.append(div);
  } else {
    var messageDivs = document.querySelectorAll("div.message");
    messageDivs.forEach(function (message) {
      message.style.display = false;
    });
  }
}, { enableHighAccuracy: "never" });
```

Type Inference

Type Annotation

```
var y: number;
y = 1; // compiles
y = "hello"; // won't compile: "Type 'string' is not assignable to type number"
```

```
function sayHelloTo(name: string): string {
  return "hello " + name;
}

var message = sayHelloTo("Tim"); // message inferred as string
var result: number = sayHelloTo("everybody"); // won't compile
message = sayHelloTo({name: "Tim"}); // won't compile
```

Type Annotation

```
var callback: (e: Event) => boolean;
```

```
var person: { name: string, age: number };
```

```
var lotsOfStrings: string[];
```

```
var stringOrNumber: string|number;
```

```
var noIdea: any;
```

Interfaces

```
var greeter: { hello: (name: string) => string, bye: (name: string) => string, language: string };
```

```
interface Greeter {
  hello(name: string): string;
  bye(name: string): string;

  language: string;
}

var greeter: Greeter;
```

Classes

```
class Greeter {
  hello(name: string): string {
    return "hi " + name;
  }

  bye(name: string): string {
    return "bye " + name;
  }

  language = "English";
}

var greeter: Greeter = new Greeter();
```

```
var Greeter = (function () {
  function Greeter() {
    this.language = "English";
  }
  Greeter.prototype.hello = function (name) {
    return "hi " + name;
  };
  Greeter.prototype.bye = function (name) {
    return "bye " + name;
  };
  return Greeter;
})();
var greeter = new Greeter();
```

Optional Typing

ES2015 (ES6)

Arrow functions `(x) => x * 2`

Destructuring `var {a, b} = {a:1, b:2}`

Spread operator `var xs = [1, 2, ...iterable, 5, 6];`

For/of `for (var x of [1, 2, 3}) { ... }`

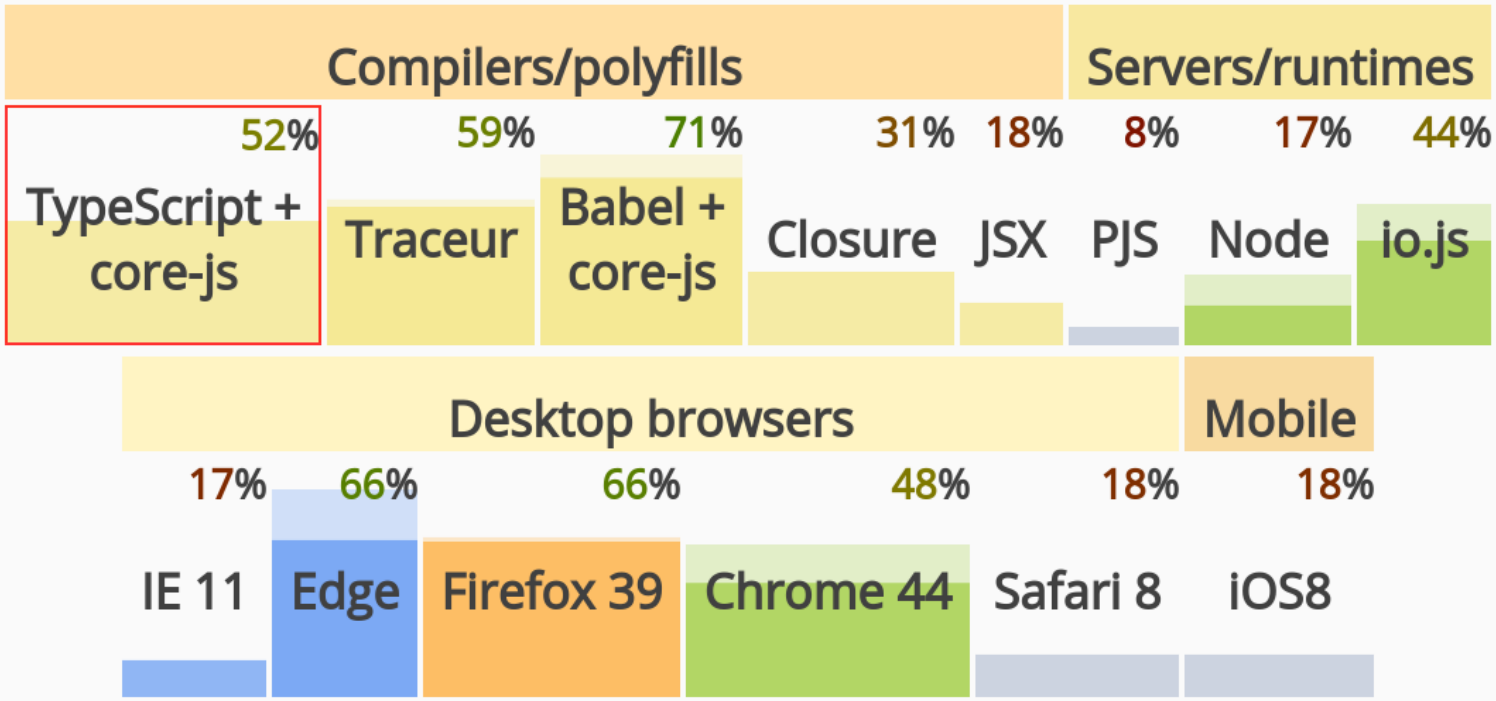
Symbols `var s = Symbol("mySymbol")`

Let/const `let x = 1; const y = "hi"`

Template strings `y = `hello ${myVar}``

And ES2015 modules, classes, promises, default parameters, unicode support, ES7 decorators, and more...

ES2015 Compatibility



(See kangax.github.io/compat-table/es6/ for full details)

ES2015 Backward compatibility

```
var f = (x, y) => x * y

var {a, b} = {a:1, b:2}

var iterable = [3, 4];
var abc = [1, 2, ...iterable, 5, 6];

for (var i of abc) {
  f(i, 10);
}


let x = 1;
const y = "hi"

var z = `hello ${y}`
```

```
var f = function (x, y) { return x * y; };

var _a = { a: 1, b: 2 }, a = _a.a, b = _a.b;

var iterable = [3, 4];
var abc = [1, 2].concat(iterable, [5, 6]);

for (var _i = 0; _i < abc.length; _i++) {
  var i = abc[_i];
  f(i, 10);
}

var x = 1;
var y = "hi";

var z = "hello " + y;
```

TypeScript Roadmap

ES2015 Generators

More powerful types: local types, intersection types, generic aliases

More powerful classes: 'extends' expressions, abstract classes

JSX support

Async/await

Built-in module bundling

And more...

(See github.com/Microsoft/TypeScript/wiki/Roadmap for full details)

Downsides

Compile step

Occasional type wrangling

Could be more ambitious

Other options

Closure compiler

Flow

Babel

TypeScript

will finally bring peace to your troubled soul

Tim Perry
@pimterry
Tech Lead & Open-Source Champion at Softwire