

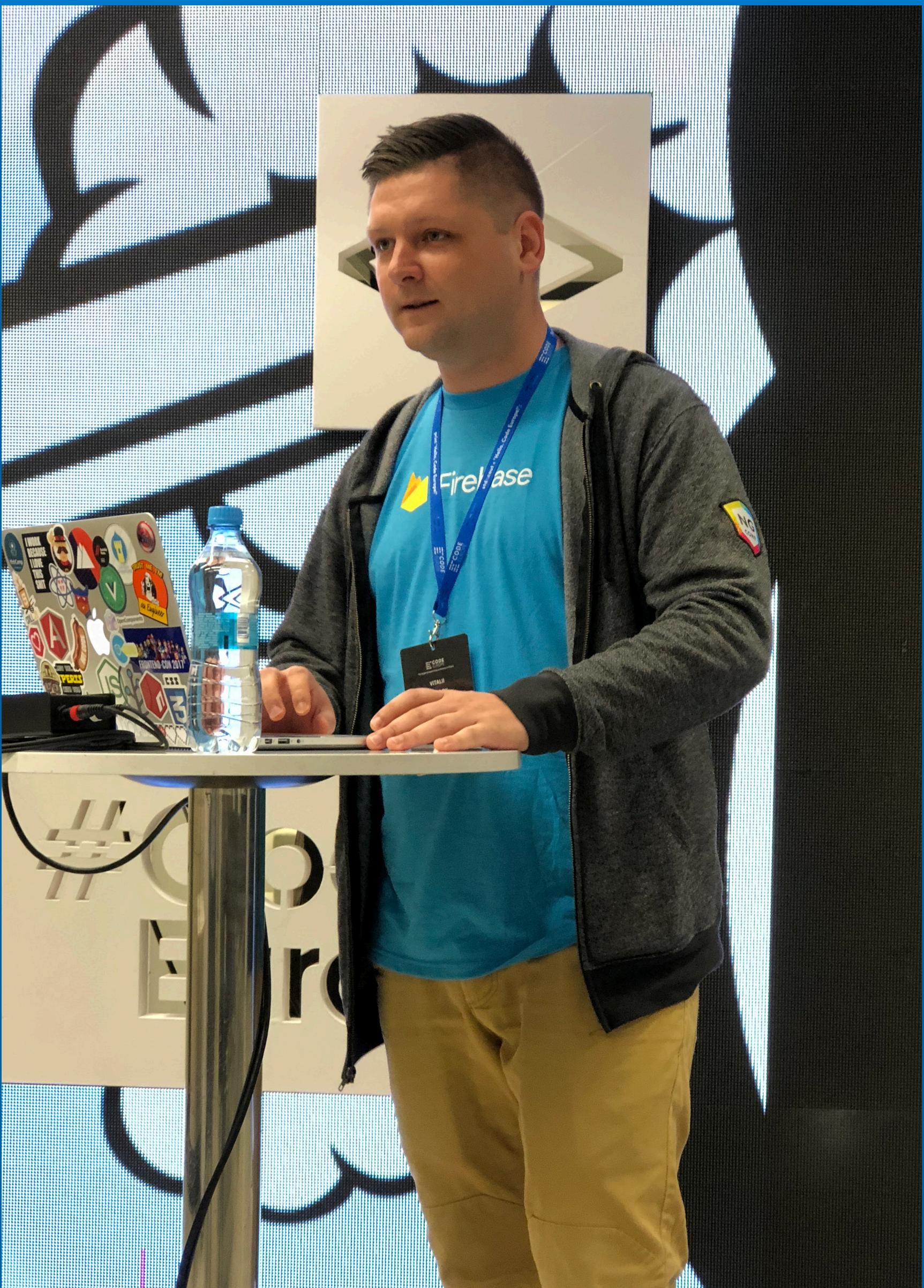
TypeScript: JavaScript you are
gonna to ❤

Who am I?

- 5 years of front-end experience
- big enterprise projects
- mentor
- OSS contributor

@bobrov1989

<https://vitaliy-bobrov.github.io/>



Agenda

- WTF TypeScript!?
- Type system
- How to Start?
- Tooling
- Integrations



WTF TypeScript!?

Trivial Problem

```
function add(a, b) {  
    return a + b;  
}  
  
const summ = add(2, 3); // 5  
const newSum = add(1, '1'); // 11
```

JS Expression Trivia



JS Expression Trivia

```
{ } + { } === ?
```

JS Expression Trivia

```
1 + + '1' === ?
```

JS Expression Trivia

[1] + [2] === ?

There are **NO DEFECTS**
if there is **NO DOCS**

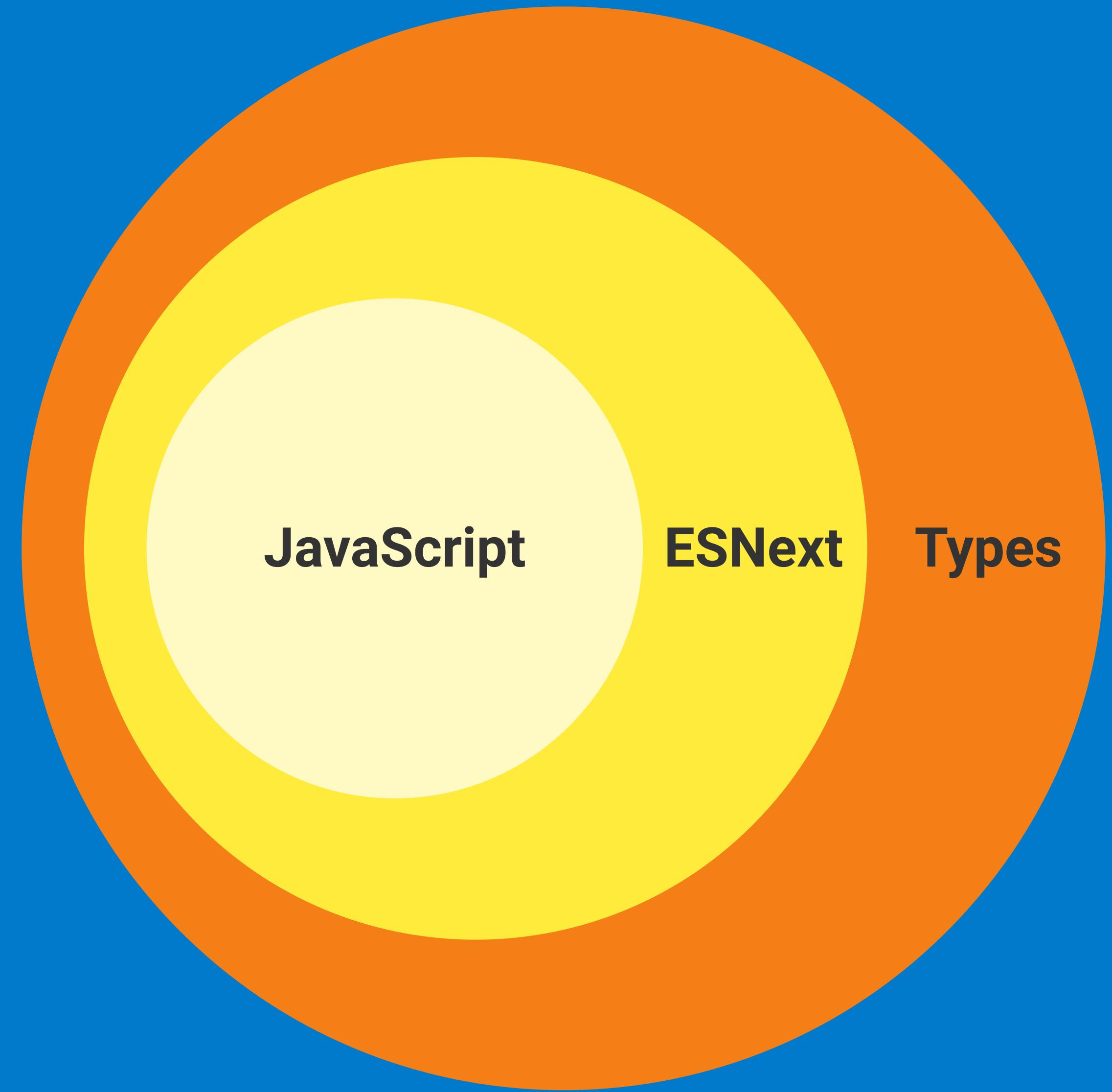
Trivial Solution

```
/**  
 * Adds two numbers.  
 * @param {number} a  
 * @param {number} b  
 * @return {number}  
 */  
function add(a, b) {  
    return a + b;  
}
```

jsDoc

- ◉ Too verbose
- ◉ Remember to update
- ◉ No warnings of wrong usage

TypeScript



Any JavaScript
is **VALID** TypeScript

ESNext

```
async function getData(url) {  
    return await fetch(url);  
}  
  
@Injectable()  
class DecorateMe {  
    static field = true;  
  
    @Log('error')  
    someMethod() {}  
}
```

WHY?

- Complex projects
- Static analysis
- Reduce errors rate
- Easy to refactor and maintain



Type System

let variable: TYPE;

Cheat

```
let variable: any;  
  
variable = null;  
variable = 1;  
variable = 'text';
```

Base Types

```
let flag: boolean = true;
let str: string = '';
let num: number = 2;
let a: undefined;
let b: null = null;
let obj: Object = {};

let map = new Map([['key', 'value']]);
let set = new Set([1, 2, 3, 1]);
let symbol = new Symbol('mySym');
```

Functions

```
const myFunc: Function = x => x * 2;

const anotherFn: (x: number) => number = x => x + 2;

function noReturn(a: number): void {
    a++;
}

function throwEx(): never {
    throw new Error('Never return value');
}
```

Array & Tuple

```
const list: Array<string> = ['a', 'b', 'c'];
const nums: number[] = [1, 2, 3];
const tuple: [string, number, boolean] = ['val', 12, false];
```

enum

```
enum Colors {  
  BLACK,  
  WHITE,  
  REBECCAPURPLE  
}  
  
enum Actions {  
  FETCH = 'FETCH',  
  FETCH_SUCCESS = 'FETCH_SUCCESS',  
  FETCH_ERROR = 'FETCH_ERROR'  
}
```

Interface

```
interface BasicResponse {  
    status: number;  
    message: string;  
}  
  
interface User {  
    fullName: string;  
    email: string;  
}  
  
interface GetUsersResponse extends BasicResponse {  
    data: User[];  
}
```

Abstract Class

```
abstract class RenderBehaviour {  
    constructor(private canvas: Canvas) {}  
  
    render(data: number[]) {  
        this.canvas.render(data);  
    }  
}  
  
class Ctx2DRender extends RenderBehaviour {  
    constructor(private canvas: Canvas) {  
        super(canvas);  
    }  
  
    helper(x: number): string {  
        return `${x}%`;  
    }  
  
    render(data: number[]) {  
        const mapped = data.map(this.helper, this);  
  
        this.canvas.render(mapped);  
    }  
}
```

Union types

```
type literal = 'val1' | 'val2';  
  
type unionAction = FetchAction | FetchSuccessAction;
```

Partials

```
type ObjWithDate = Partial<{date: Date}>;  
  
function sortByDate(a: ObjWithDate, b: ObjWithDate): number {  
    return a.date.getTime() - b.date.getTime();  
}
```

DOM API

```
const button: HTMLButtonElement = document.querySelector('.btn');

button.addEventListener('click', (event: MouseEvent) => {
  event.preventDefault();

  const x = event.pageX;
  const y = event.pageY;
});
```

<http://www.typescriptlang.org/docs/>

The screenshot shows the TypeScript documentation website. The header includes links for Quick Start, Documentation, Download, Connect, and Playground, along with a 'Fork me on GitHub' button. A banner at the top announces 'TypeScript 2.8 is now available. Download our latest version today!'. The main content area is titled 'Basic Types' and includes sections for 'Introduction', 'Boolean', and 'Number'. Each section contains a code snippet example.

TypeScript Quick Start Documentation Download Connect Playground

Fork me on GitHub

TypeScript 2.8 is now available. [Download our latest version today!](#)

Documentation ▾

Basic Types

Introduction

For programs to be useful, we need to be able to work with some of the simplest units of data: numbers, strings, structures, boolean values, and the like. In TypeScript, we support much the same types as you would expect in JavaScript, with a convenient enumeration type thrown in to help things along.

Boolean

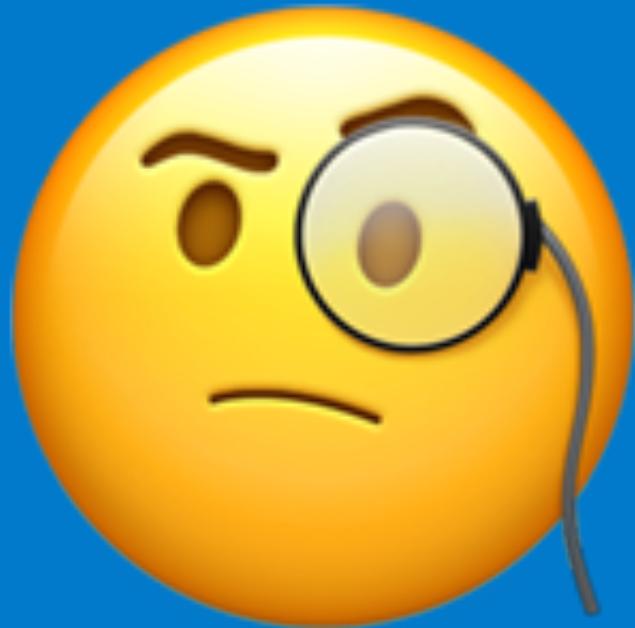
The most basic datatype is the simple true/false value, which JavaScript and TypeScript call a `boolean` value.

```
let isDone: boolean = false;
```

Number

As in JavaScript, all numbers in TypeScript are floating point values. These floating point numbers get the type `number`. In addition to hexadecimal and decimal literals, TypeScript also supports binary and octal literals introduced in ECMAScript 2015.

```
let decimal: number = 6;
```



How to Start?

REPL

The screenshot shows the TypeScript Playground interface. At the top, there's a navigation bar with links: TypeScript, Quick Start, Documentation, Download, Connect, and Playground. A blue banner below the navigation bar says "TypeScript 2.8 is now available. Download our latest version today!" and features a "Fork me on GitHub" button.

The main area contains two code editors. The left editor is titled "Using Classes" and contains the following TypeScript code:

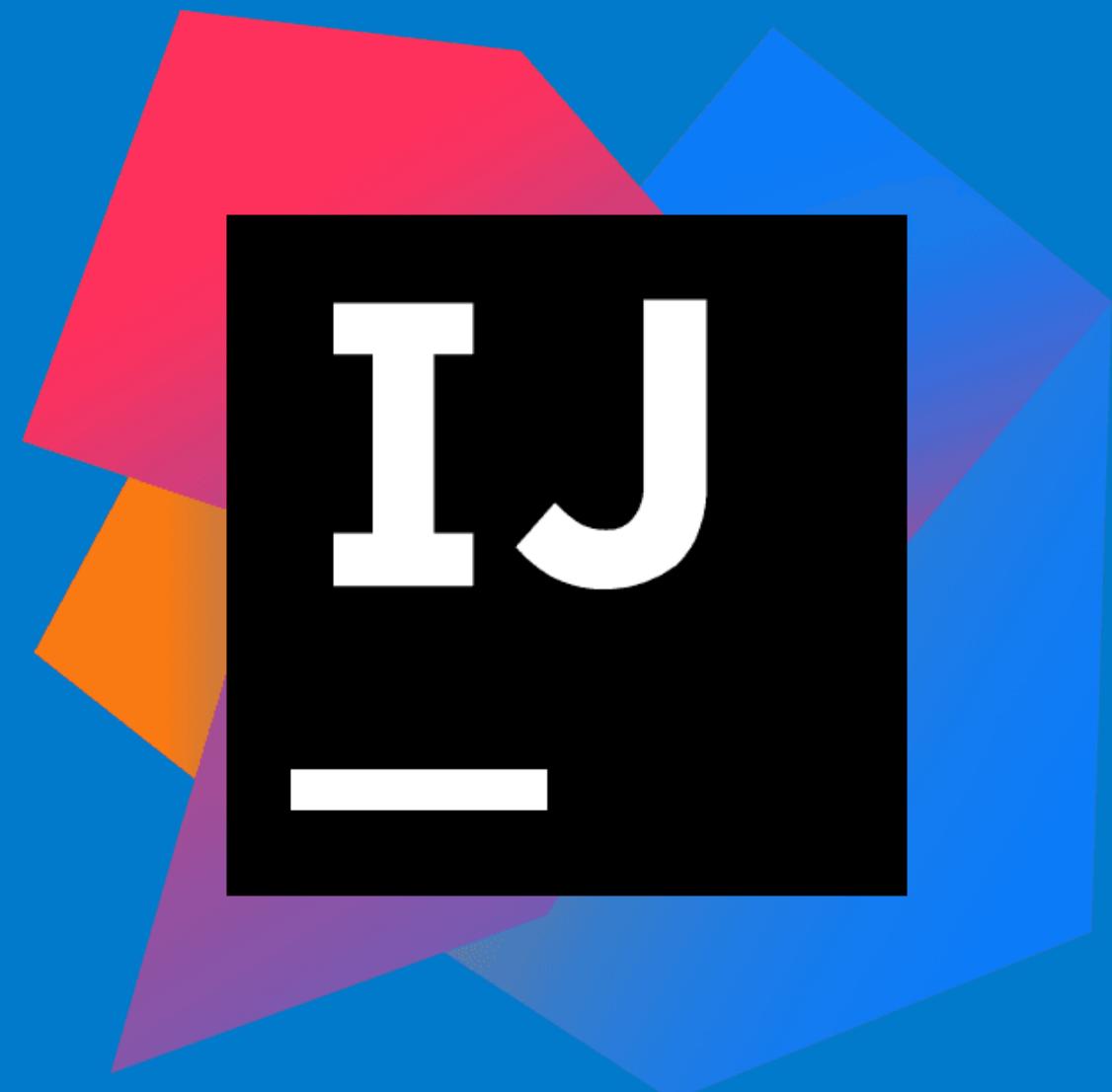
```
1 class Greeter {
2     greeting: string;
3     constructor(message: string) {
4         this.greeting = message;
5     }
6     greet() {
7         return "Hello, " + this.gre
8     }
9 }
10
11 let greeter = new Greeter("world");
12
13 let button = document.createElement("button");
14 button.textContent = "Say Hello";
15 button.onclick = function() {
16     alert(greeter.greet());
17 }
18
19 document.body.appendChild(button);
```

The right editor contains the generated JavaScript code:

```
1 var Greeter = /** @class */ (function () {
2     function Greeter(message) {
3         this.greeting = message;
4     }
5     Greeter.prototype.greet = function () {
6         return "Hello, " + this.gre
7     };
8     return Greeter;
9 })();
10 var greeter = new Greeter("world");
11 var button = document.createElement("button");
12 button.textContent = "Say Hello";
13 button.onclick = function () {
14     alert(greeter.greet());
15 };
16 document.body.appendChild(button);
17
```

<http://www.typescriptlang.org/play/index.html>

IDE & Text Editors





Just change extension
from **.js** to **.ts**



Tooling

JavaScript checks

```
// @ts-check

/** 
 * Adds two numbers.
 * @param {number} a
 * @param {number} b
 * @return {number}
 */
function add(a, b) {
    return a + b;
}

const summ = add(2, 3); // 5
const newSum = add(1, '1'); // 11
```

TypeScript compiler

```
npm i -g typescript  
tsc --init  
tsc "./my-project/**/*.ts"
```

tsconfig.json

```
{  
  "compileOnSave": false,  
  "compilerOptions": {  
    "outDir": "./dist/out-tsc",  
    "sourceMap": true,  
    "declaration": false,  
    "moduleResolution": "node",  
    "emitDecoratorMetadata": true,  
    "experimentalDecorators": true,  
    "target": "es5",  
    "typeRoots": [  
      "node_modules/@types"  
    ],  
    "lib": [  
      "es2017",  
      "dom"  
    ]  
  }  
}
```



Webpack

```
module.exports = {  
  mode: "development",  
  devtool: "inline-source-map",  
  entry: "./app.ts",  
  output: {  
    filename: "bundle.js"  
  },  
  resolve: {  
    // Add `ts` and `tsx` as a resolvable extension.  
    extensions: [".ts", ".tsx", ".js"]  
  },  
  module: {  
    rules: [  
      // all files with a `ts` or `tsx` extension will be handled by `ts-loader`  
      { test: /\.tsx?$/, loader: "ts-loader" }  
    ]  
  }  
};
```

Parcel



```
npm i -g parcel-bundler  
parcel build src/index.html
```

<https://parceljs.org/>

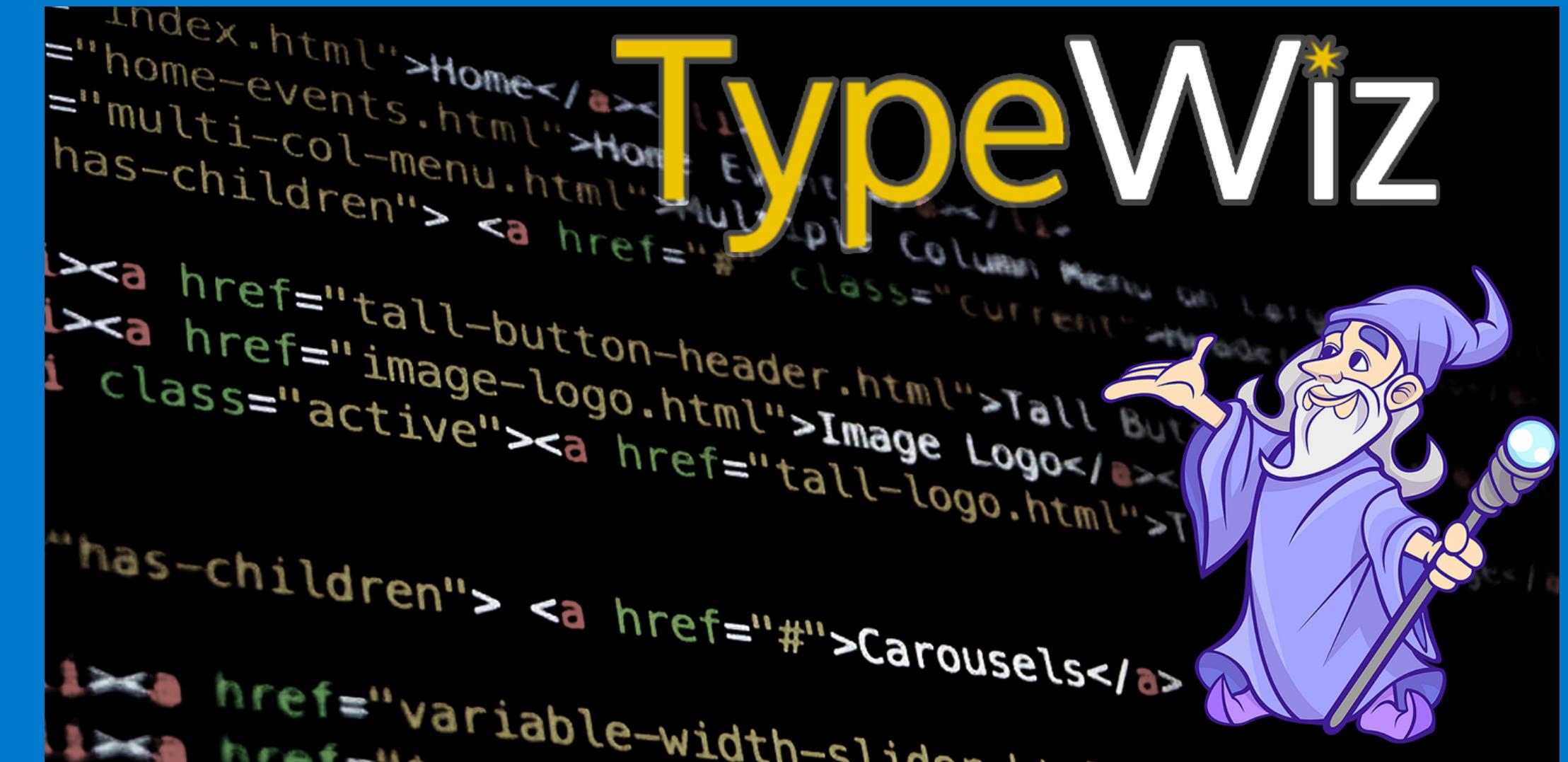
TypeWiz

TypeWiz

Automatically discover and add missing types in your TypeScript code.

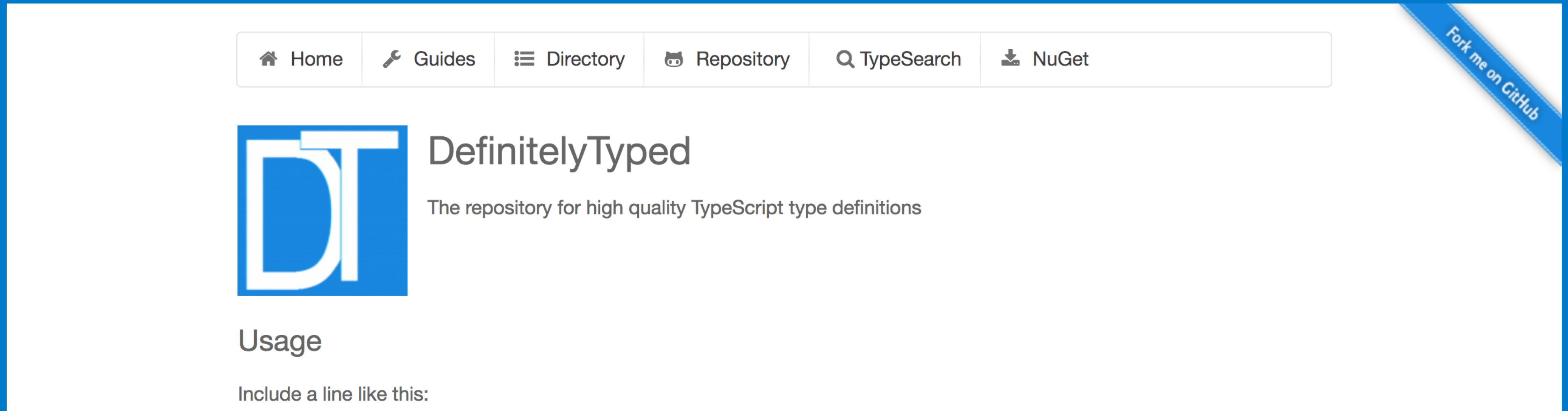
TypeWiz

build passing coverage 86% code style prettier



<https://github.com/urish/typewiz>

3rd party libraries



The screenshot shows the homepage of DefinitelyTyped. At the top, there is a navigation bar with links for Home, Guides, Directory, Repository, TypeSearch, and NuGet. A blue diagonal banner on the right says "Fork me on GitHub". The main content area features the DefinitelyTyped logo (a blue square with white letters DT) and the text "DefinitelyTyped: The repository for high quality TypeScript type definitions". Below this, there is a section titled "Usage" with the sub-instruction "Include a line like this:" followed by a code snippet.

Home Guides Directory Repository TypeSearch NuGet

Fork me on GitHub

DefinitelyTyped

The repository for high quality TypeScript type definitions

Usage

Include a line like this:

<http://definitelytyped.org/>



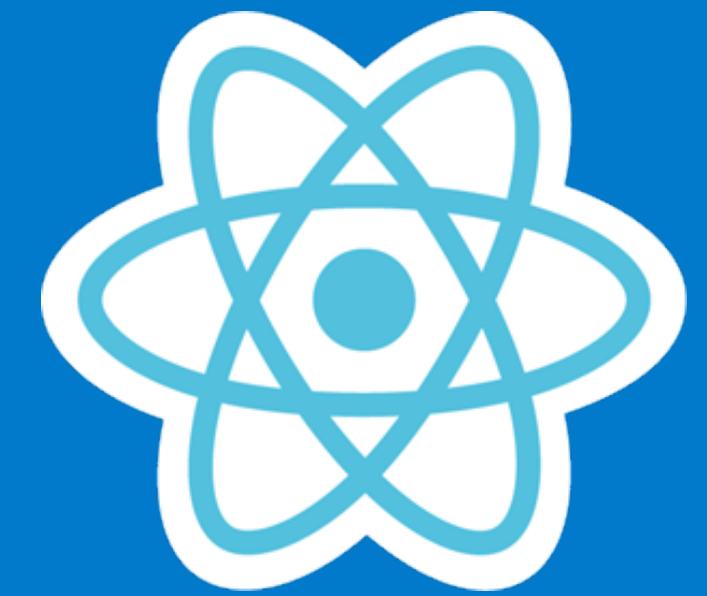
Frameworks



```
@Component({
  selector: 'app-heroes',
  templateUrl: './heroes.component.html',
  styleUrls: ['./heroes.component.css']
})
export class HeroesComponent implements OnInit {
  heroes: Hero[] = HEROES;
  private selectedHero: number;

  ngOnInit() {}

  onSelect(hero: Hero): void {
    this.selectedHero = hero;
  }
}
```



```
const initialState = { clicksCount: 0 };
type State = Readonly<typeof initialState>

class ButtonCounter extends Component<object, State> {
  readonly state: State = initialState;

  @bind
  handleIncrement() {}

  render() {
    const { clicksCount } = this.state;
    return (
      <>
        <Button onClick={this.handleIncrement}>Increment</Button>
        <Button onClick={this.handleDecrement}>Decrement</Button>
        You've clicked me {clicksCount} times!
      </>
    );
  }
}
```



```
@Component({
  template: '<button @click="onClick">Click!</button>'
})
export default class MyComponent extends Vue {
  message: string = 'Hello!'

  onClick (): void {
    window.alert(this.message)
  }
}
```

https://medium.com/@martin_hotell

Medium

Search

Martin Hochel

Jedi Knight / Developer Expert for @google | founder of @ngPartyCz | ng-metadata author | Speaker | Trainer, #js #typescript #rx #webcomponents #react #angular.



153 Following 570 Followers · [Twitter](#)

[Follow](#)

[Profile](#) [Latest](#) [Claps](#) [Highlights](#) [Responses](#)



Integrations

Strictly typed REST API calls

```
import { AuthResponse } from '@server/interfaces/auth';

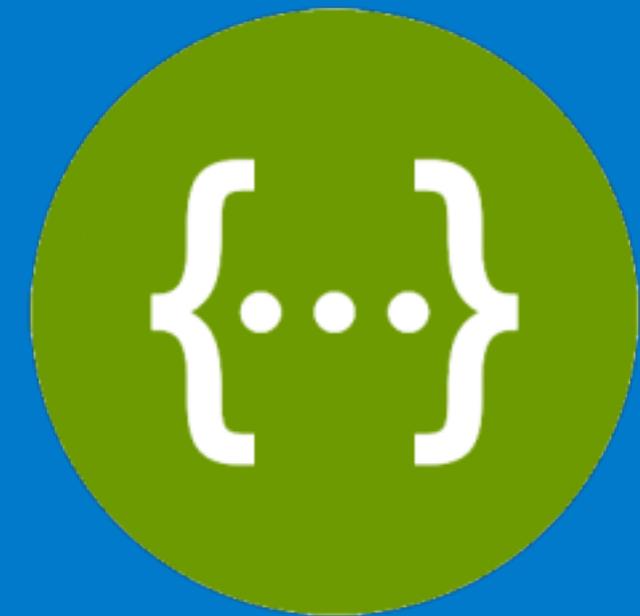
class AuthService {
  constructor(private http: HttpClient) {}

  authenticate(password: string) {
    this.http.post<AuthResponse>({ password })
      .then(response => response.data);
  }
}
```



Just use **TypeScript**
to write **NodeJS** project

Java / C#



Swagger

 JSON Schema Specification Examples Implementations

JSON Schema

The current version is draft-07!

JSON Schema is a vocabulary that allows you to **annotate** and **validate** JSON documents.

Advantages

JSON Schema	JSON Hyper-Schema
<ul style="list-style-type: none">describes your existing data formatclear, human- and machine-readable documentationcomplete structural validation, useful for<ul style="list-style-type: none">automated testingvalidating client-submitted data	<ul style="list-style-type: none">make any JSON format a hypermedia format - no constraints on document structureuse URI Templates with instance datadescribe client data for use with links using JSON Schemarecognize collections and collection items

JSON Schema

json-schema-to-typescript

Compile json schema to typescript typings

Example

Input:

```
{
  "title": "Example Schema",
  "type": "object",
  "properties": {
    "firstName": {
      "type": "string"
    },
    "lastName": {
      "type": "string"
    },
    "age": {
      "description": "Age in years",
      "type": "integer",
      "minimum": 0
    },
    "hairColor": {
      "enum": ["black", "brown", "blue"],
      "type": "string"
    }
  },
  "additionalProperties": false,
  "required": ["firstName", "lastName"]
}
```

Output:

```
export interface ExampleSchema {
  firstName: string;
  lastName: string;
  /**
   * Age in years
   */
  age?: number;
  hairColor?: "black" | "brown" | "blue";
}
```

install
➤ npm i json-schema-to-typ...

↓ last 7 days
6,531

version 5.4.0 license MIT

open issues 28 pull requests 1

repository [github.com](#)

last publish a month ago

collaborators



test in your browser



Thank YOU!

@bobrov1989

<https://vitaliy-bobrov.github.io/>