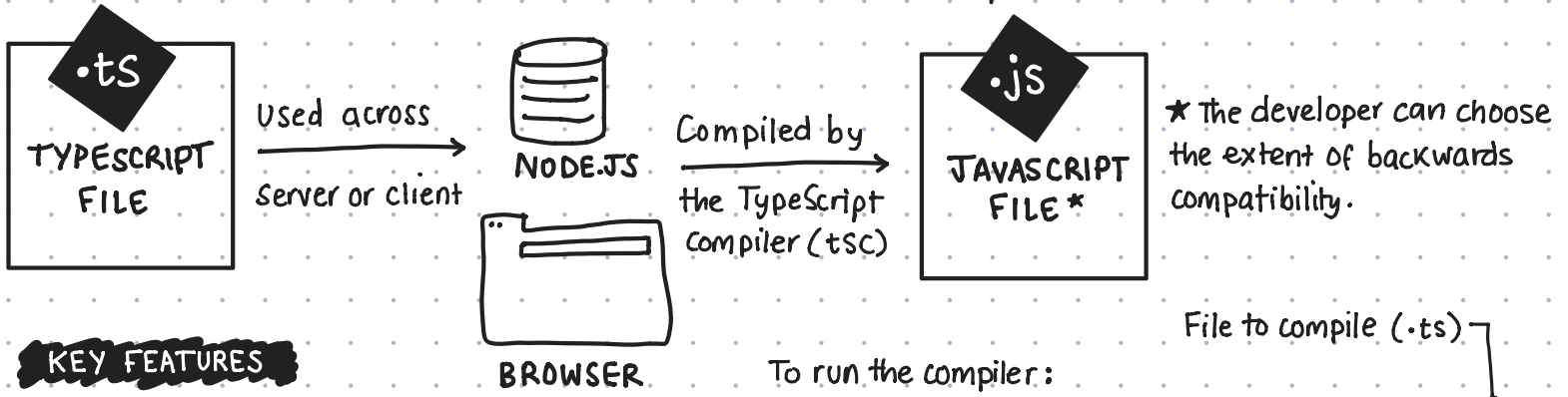


# TYPESCRIPT BASICS

TypeScript is an extension of JavaScript that adds a plethora of features to help developers:

- ☑ Write less buggy code
- ☑ Organize large codebases
- ☑ Take advantage of IDE functionality
- ☑ Generate backwards compatible JavaScript (easily!)



## KEY FEATURES

### STATIC TYPING

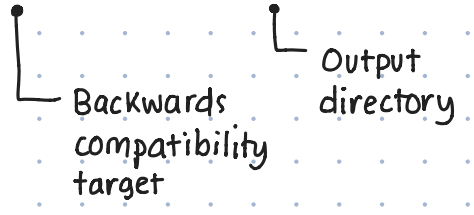
Type annotations indicate what type is expected for a variable:

```
let name: string = "fred";
```

```
let temps: number[] = [80, 78];
```

It works for arrays too!

```
tsc --target ES2015 --outdir js filename
```



TypeScript will also try to infer type based on the assigned value:

```
let greeting = "hi";
```

💡 This variable is of type string.

Once a variable's type has been set or inferred, you cannot assign to it a value of a different type:

```
greeting = 42; (X) Error!
```

Union two or more types for added flexibility:

```
let code: string | number;
```

You don't have to assign a value at time of declaration.

Or, opt out of type checking:

```
let hello: any;
```

Type annotations are also used in function headers:

```
function getStatus (code?: number): string { }
```

Note: In TypeScript, all function parameters are required unless explicitly marked as optional using `?`.

### IMPROVED IDE TOOLING

Because TypeScript is a compiled language with stronger typing, we get to take advantage of IDE features including: code completion, helpful errors, source maps & more.

### ORGANIZATION

**classes** TypeScript classes >> ES2015 classes because they can be compiled to earlier versions of JavaScript.

**name-spaces** These features enable you to compartmentalize your code, with similar functionality to other languages.

**modules**

**interfaces** Interfaces establish a contract for how code is supposed to look.