

TypeScript

Revision:- .ts

to compile:- `npm run tsc` name.ts

optional

↳ filename

* `tscconfig.json`

```
{
```

```
  "compilerOptions": {
```

```
    "lib": ["dom", "es6"]
```

```
  }
```

```
}
```

* Implicit types:- (ts recognizes it itself)

* Explicit types:- (we define it)

Syntax:-

`let varName: type =;`

* Types:-

Boolean, Number, String, Array, Tuple,

Enum, Any

↳ basically js

* Tuple:- Array with fixed # of values with known types.

Syntax

`type typeName = [String, Number]`

↳ 2 value

* Enums: define a set of named constants

Eg:-

```
enum Continents {  
    N-A,  
    S-A,  
    Africa,  
    Asia,  
    Europe,  
    Antarctica,  
    Australia  
}
```

```
var region = Continents.Asia;
```

* Interface:- types, like a contract

```
interface User {  
    name: string,  
    id: number;  
}
```

```
→ const user: User = {  
    name: 'John',  
    id: 0,  
}
```


* Composing types (Union) +

set of thing a value can be.

```
type WindowStates = "open" | "closed" | "mini-  
-mized";
```

```
const state: WindowState = "x"; X  
= "open" ✓
```

* Function types:

```
const name = (param: string | string[]) => {  
  return param.length;  
}
```