

Basic CONCEPTS TypeScript

Basic npm commands for TS project

```
npm init -y
npm i typescript
npx tsc --init
npx tsc --init --rootdir src --outdir lib
npx tsc --watch
```

New JS type Bigint

```
// new type
let bigint: bigint = 24n;

// Tuple type
let tuple: [number, number] = [0, 0];
```

Tuple type

```
// Tuple type
let tuple: [number, number] = [0, 0];
```

Duck type annotation

```
type Point2d = { x: number; y: number };
type Point3d = { x: number; y: number; z: number };

let point2d: Point2d = { x: 0, y: 0 };
let point3d: Point3d = { x: 0, y: 0, z: 0 };

/* EXTRA info ok */

point2d = Point3d;
function takesPoint2d(point2d: Point2d) {}
takesPoint2d(point3d);

// Error !!!
point3d = Point2d; // Error
function takesPoint3d(point3d: Point3d) {}
takesPoint3d(point2d); // Error
```

Unknown type

1. If you don't know what type you are looking

```
let exampleUnknown: unknown; // I don't now what is the type

if (typeof exampleUnknown === "string") {
  exampleUnknown.trim()
}

if (typeof exampleUnknown === "boolean") {
  let boolean: boolean = exampleUnknown
}
```

Assertions type

1. We are telling the type script compiler "I now what is a type"
2. don't use angel brackets <string>
3. when you use this type and changed type to another TS don't show error

```
let hello = load();

const timed = (hello as string).trim();
hello = 0; // now can't see !ErrorTS
```

+(Casting) type

1. We are telling the type script compiler "I now what is a type"

```
let letters = "hello";
const number = +letters;
console.log(number === 123123);
```

Declaration type

1. Declaration new global variable `env.d.ts`
 1. created file `env.d.ts`
 2. `export declared const process: any;`
 3. access to this variable `process.env.USER`

```
process.env.User; // can access
```

Preparation for work with TS

1. Working with node.js files

1. Node.s cant Working with TS files!

1. Need convert to JS use command `npx tsc`

2. `node index.js`

2. Skip this process:

1. Need install package `npm install ts-node`

2. use command `npm ts-node index.ts`

3. OR set `package.json`

```
"scripts": {start: "ts-node index.ts"}
```

2. Working with node.js variables for example: process.env.User

1. Need use nmp command `npm install @types/node`

2. You'll get access to `env` variable form node.js

```
env; // can access  
import fs from "fs"; // can access  
fs.writeFileSync("hello.txt", "hello");
```

3. Working with TS ExpressJS

```
npm install express  
npm install @types/express
```