



Chapter 2

Typescript Language



Agenda

- Basic Input Output
- Operation
- Variable
- Control Structure
- Repeat Structure
- Function

Before We Start

- Setup Development Environment
- Create this chapter project:

- **Command**

1. `mkdir ts_chapter2`
2. `cd ts_chapter2`
3. `npm init --y`
4. `npm i -D @types/node`
5. `tsc --init`
6. Change `tsconfig.json` to ==>



```
1 {
2   "compilerOptions": {
3     "target": "ESNext",
4     "module": "NodeNext",
5     "rootDir": "./src",
6     "moduleResolution": "NodeNext",
7     "outDir": "./dist",
8     "esModuleInterop": true,
9     "forceConsistentCasingInFileNames": true,
10    "strict": true,
11    "skipLibCheck": true
12  },
13  "include": ["src/**/*.ts"],
14  "exclude": ["node_modules", "dist"]
15 }
16
```



Check if the
project is
ready!!



Create **src** in the root of the project



Create **hello-world.ts** in **src**



Open console and build
hello-world.ts using command tsc



After that project will generate
dist folder with **hello-world.js** in it



To run the hello-world.js using command
node dist/hello-world.js



Basic Input Output

Basic Output



```
1 console.log("this is a basic output");  
2 console.error("this is an error");  
3 console.warn("this is a warning");  
4 console.info("this is an info");  
5 console.debug("this is a debug");
```

```
D:\backend_course_by_saha  
this is a basic output  
this is an error  
this is a warning  
this is an info  
this is a debug
```

Basic Input



```
1 let name = process.argv[2];  
2 console.log(`Hello, ${name}!`);
```

```
D:\backend_course_by_sahachan\second_ts_class>node dist/basic-input.js sahachan  
Hello, sahachan!
```



Operation



Operation in typescript

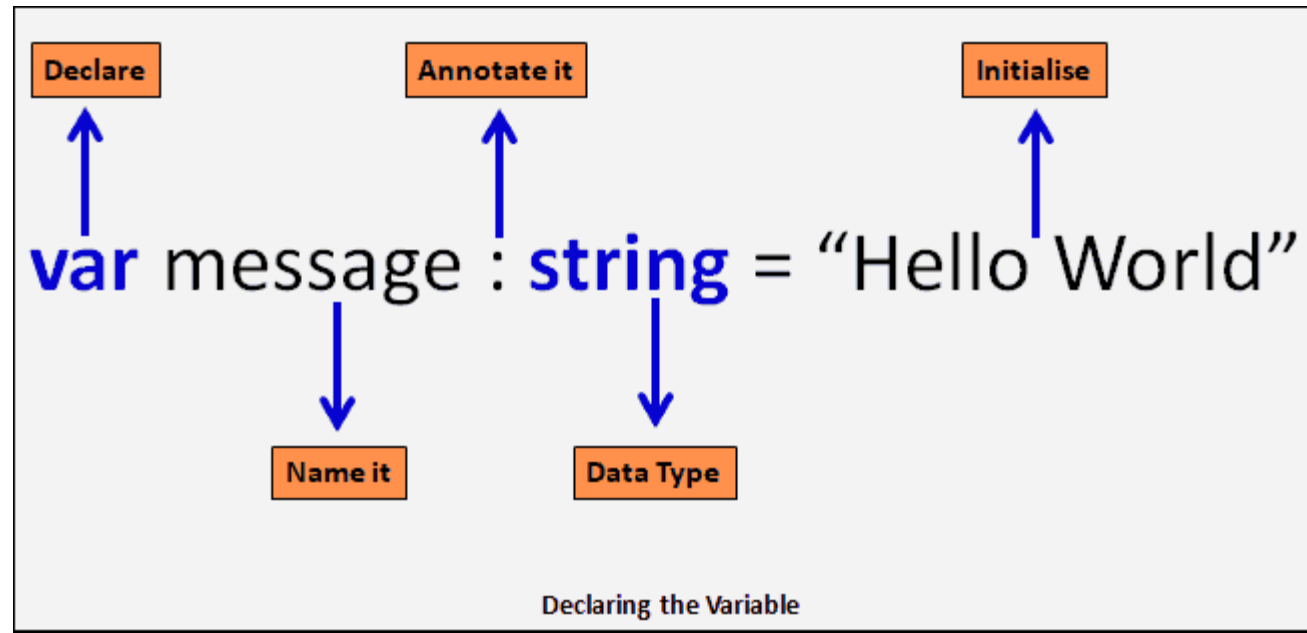
[TypeScript Operators - GeeksforGeeks](#)

[learn_ts_ch2/src/operation/operation.ts at main · oat431/learn_ts_ch2 \(github.com\)](#)

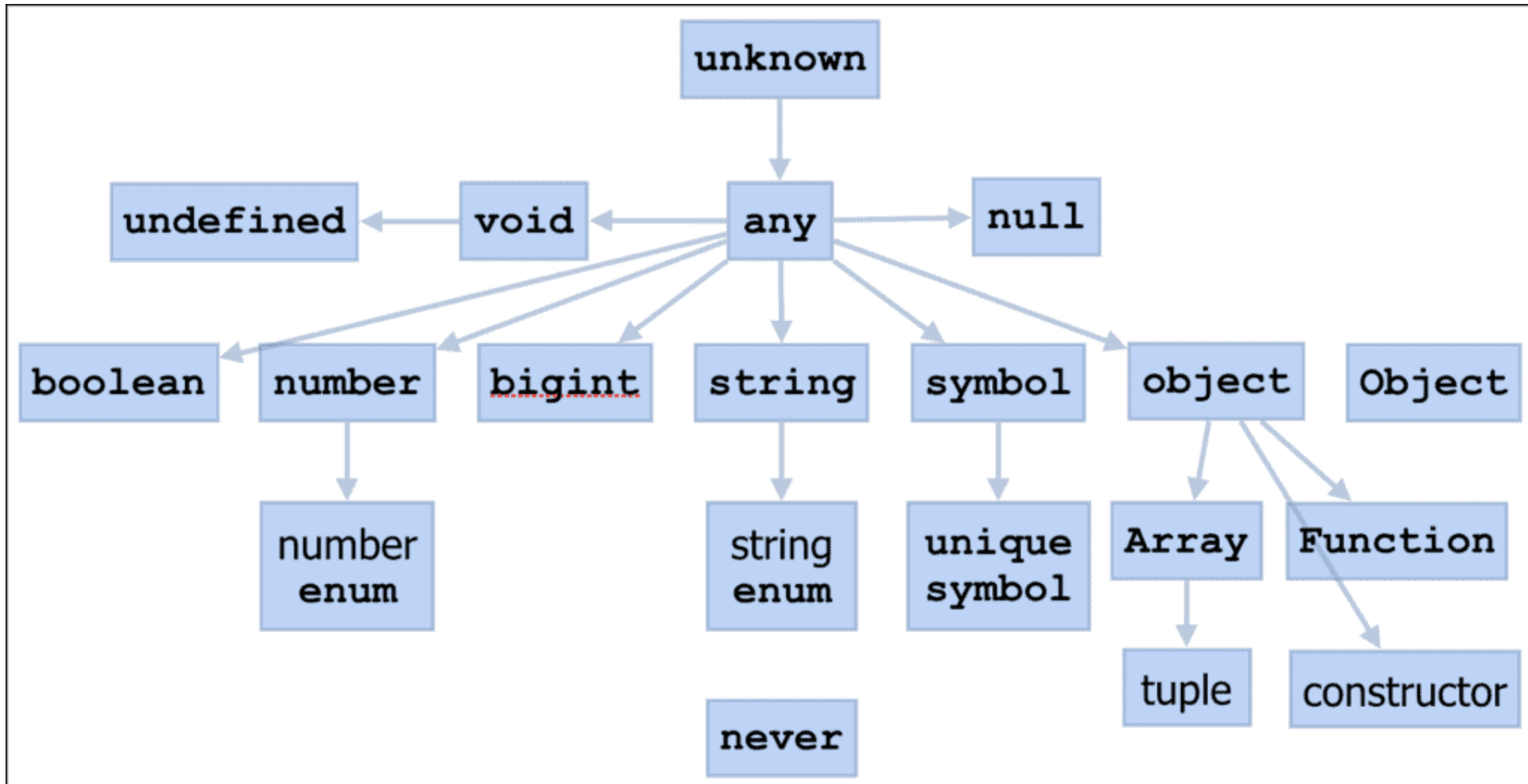


Variable

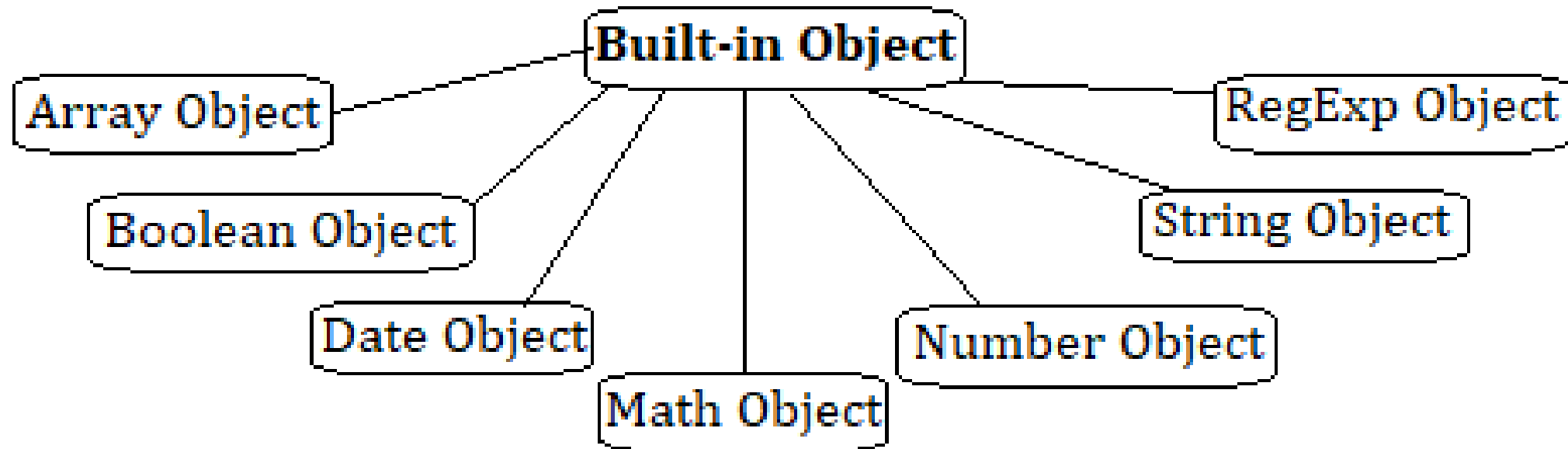
variable



Type Primitives



Built-in object





[learn_ts_ch2/src/variable at main · oat431/learn_ts_ch2 \(github.com\)](#)



Control Structure

Basic Control Structure

If Statements

Most narrowing comes from expressions inside if statements, where different type operators narrow inside the new scope

`typeof` (for primitives)

```
const input = getUserInput()
input // string | number

if (typeof input === "string") {
  input // string
}
```

`instanceof` (for classes)

```
const input = getUserInput()
input // number | number[]

if (input instanceof Array) {
  input // number[]
}
```

`"property" in object` (for objects)



```
const input = getUserInput()
input // string | { error: ... }

if ("error" in input) {
  input // { error: ... }
}
```

`type-guard functions` (for anything)

```
const input = getUserInput()
input // number | number[]

if (Array.isArray(input)) {
  input // number[]
}
```

[learn_ts_ch2/src/condition at main · oat431/learn_ts_ch2 \(github.com\)](#)



Repeat Structure

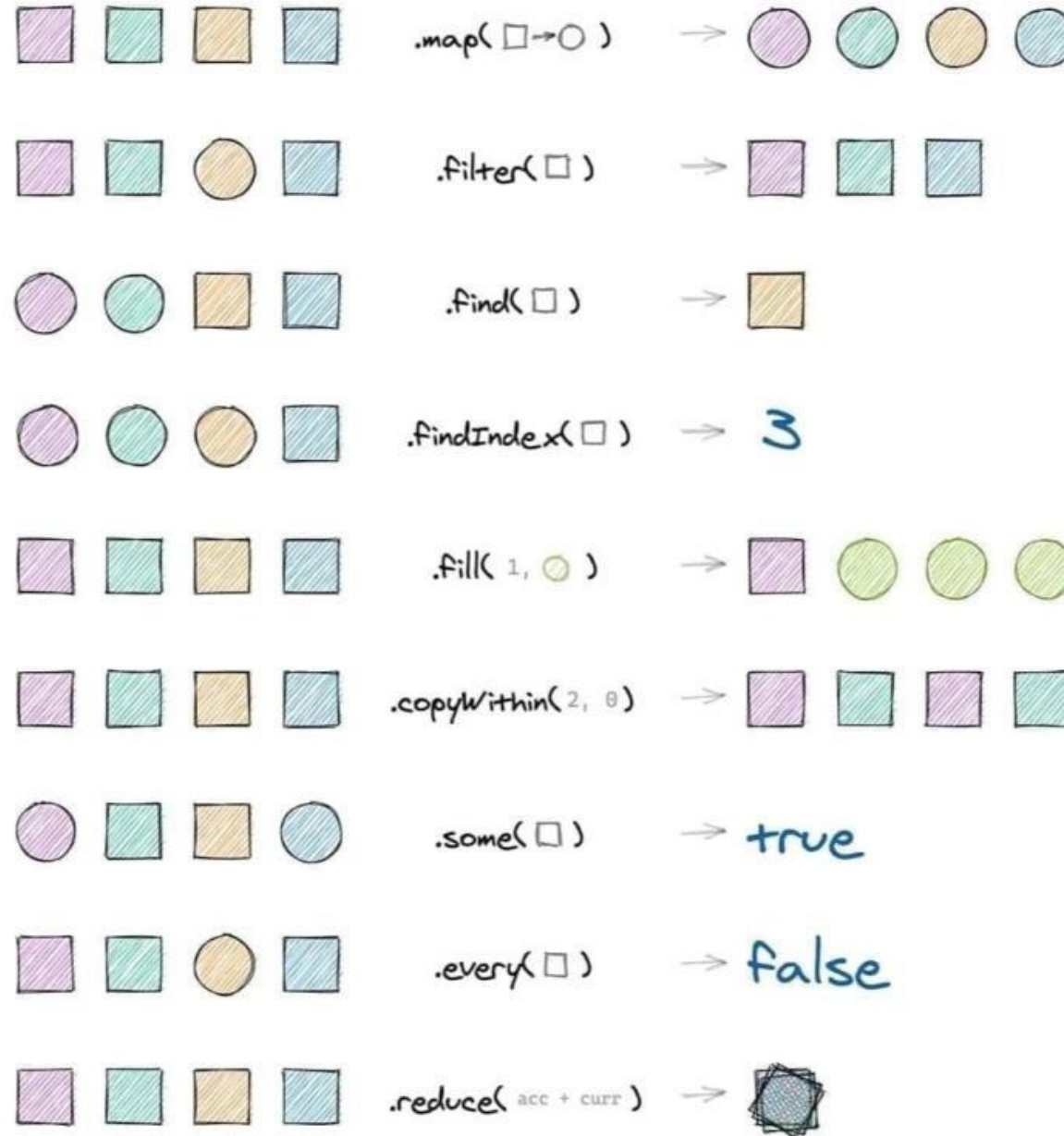
Repeat structure

[TypeScript For Loops \(tutorialsteacher.com\)](#)

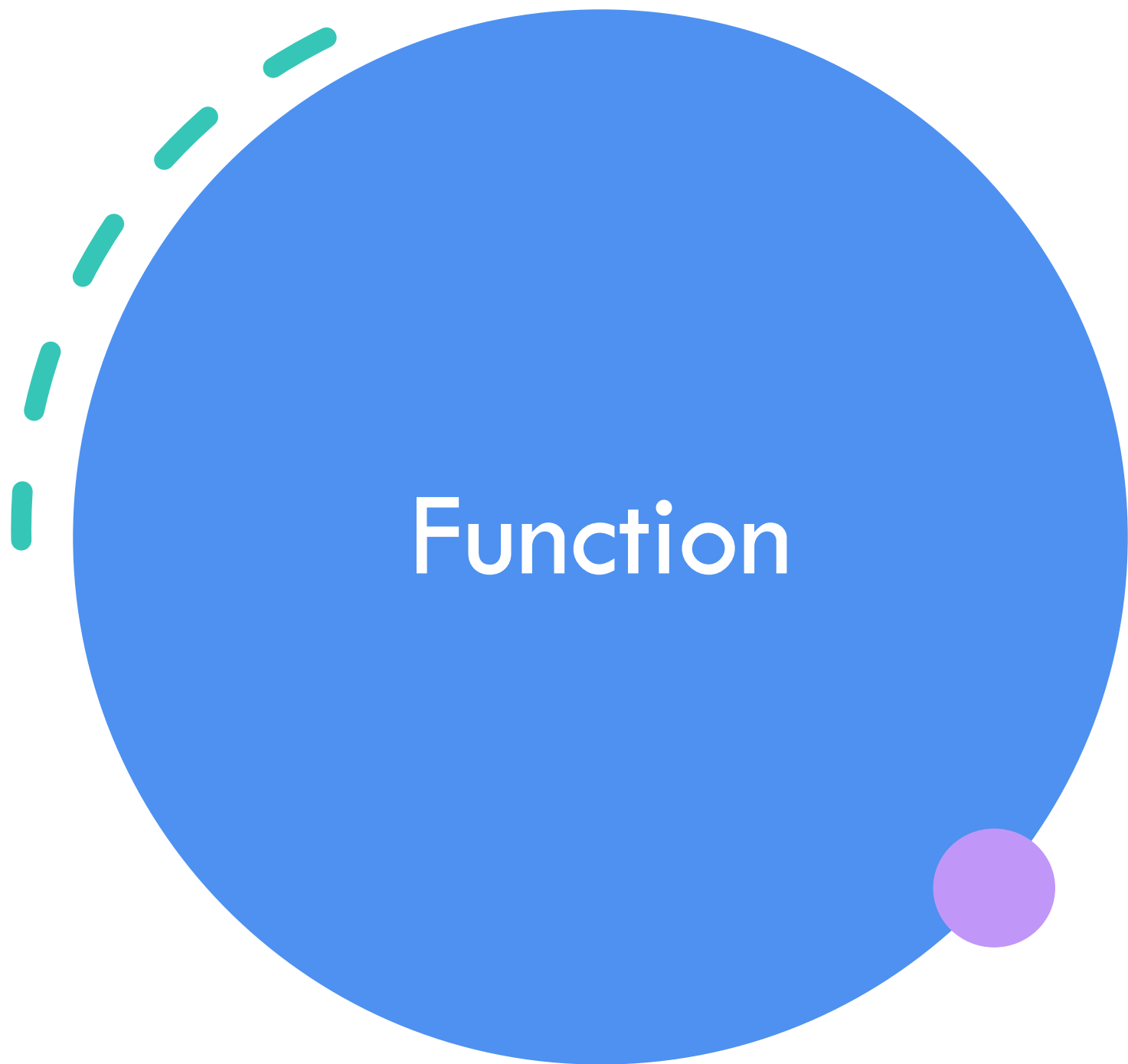
[TypeScript - while Loop \(tutorialsteacher.com\)](#)

[How to Iterate Array of Objects in TypeScript ? - GeeksforGeeks](#)

[learn_ts_ch2/src/loop at main · oat431/learn_ts_ch2 \(github.com\)](#)



presentation title





Function

[TypeScript: Handbook - Functions \(typescriptlang.org\)](https://typescriptlang.org)

[learn_ts_ch2/src/function at main · oat431/learn_ts_ch2 \(github.com\)](https://github.com/oat431/learn_ts_ch2/tree/main/src/function)



Thank you

Sahachan

oat431@gmail.com

Applied what we learn

- Basic feature of movie review
- Resource:
- Create the express api
- **Feature**
 1. Get All current comment of the movie
 2. Get Average score of movies
 3. Get specific comment
 4. Add comment

