BAITUSSALAM

BAITUSSALAM —TECH PARK—

Class Agenda

BAITUSSALAM
—TECH PARK—

Introduction to Typescript, CRUD and Blog Application

Typescript

What is Typescript?

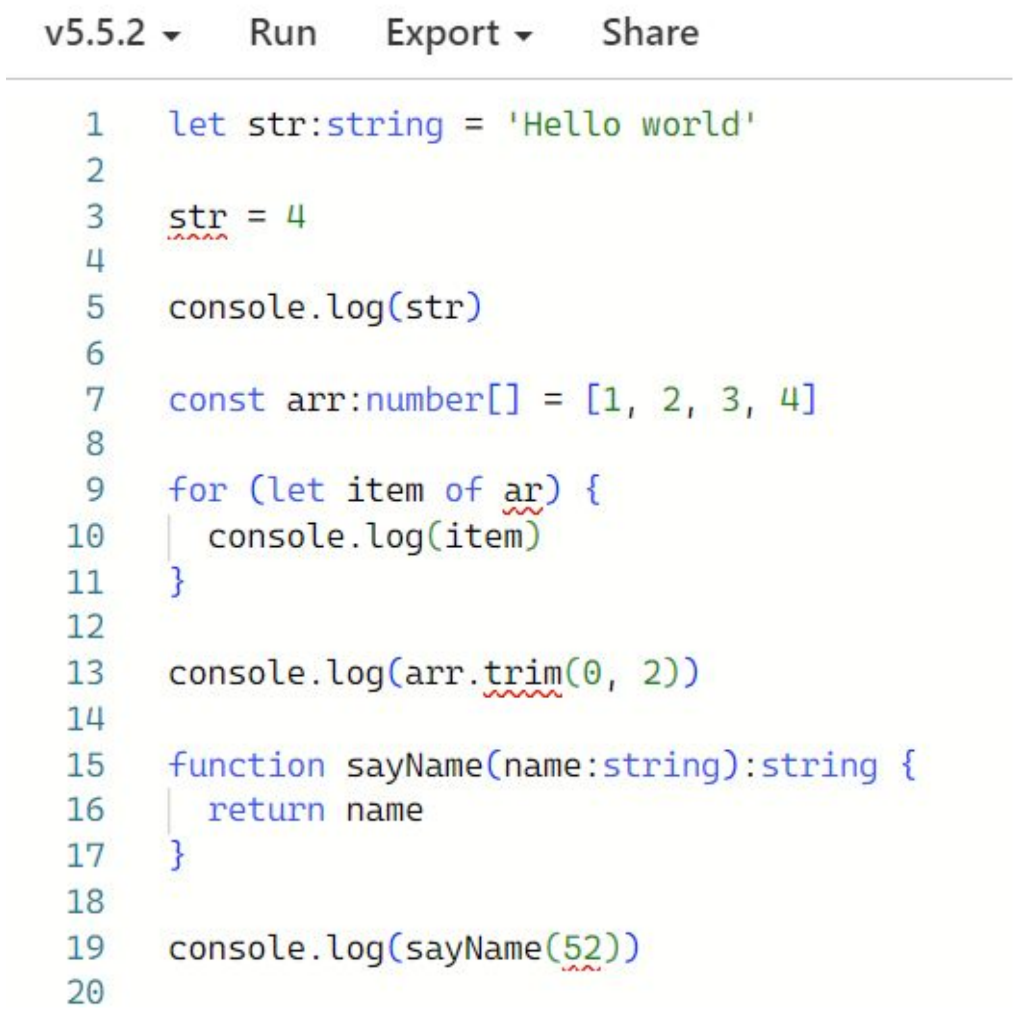
- TypeScript is JavaScript with syntax for types.
- TypeScript add static typing to JavaScript.
- TypeScript allows specifying the types of data being passed around within the code, and has the ability to report errors when the types don't match

Features of Typescript

Benefits of TypeScript Over JavaScript

- Improved Code Quality: Early detection of errors through static type checking.
- Better Tooling: Enhanced auto completion, refactoring, and inline documentation.
- Scalability: Easier to manage and maintain large codebases.
- Documentation: Types serve as documentation, making the code more understandable.

Typescript Examples



```
let str:string = 'Hello world'

str = 4

console.log(str)

const arr:number[] = [1, 2, 3, 4]

for (let item of ar) {
  console.log(item)
}

console.log(arr.trim(0, 2))

function sayName(name:string):string {
  return name
}

console.log(sayName(52))

```

Javascript and Typescript Types

BAITUSSALAM
—TECH PARK—

Javascript

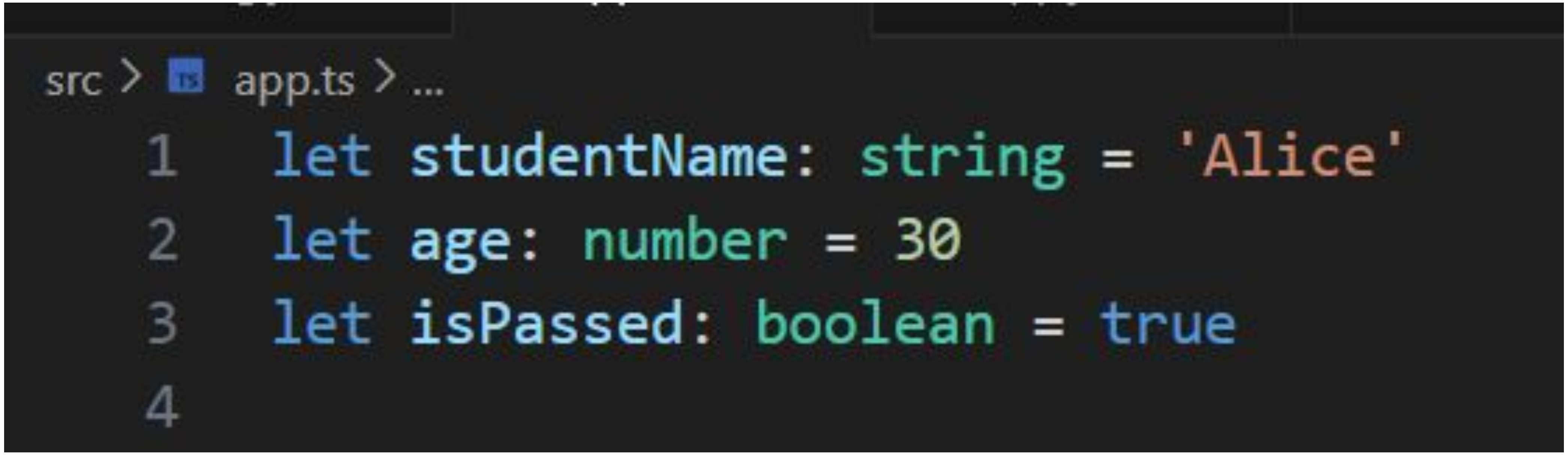
- number
- string
- boolean
- null
- undefined
- object

Typescript

- any
- unknown
- never
- enum
- tuple
- void

Types in TypeScript - Primitives

number, string and boolean



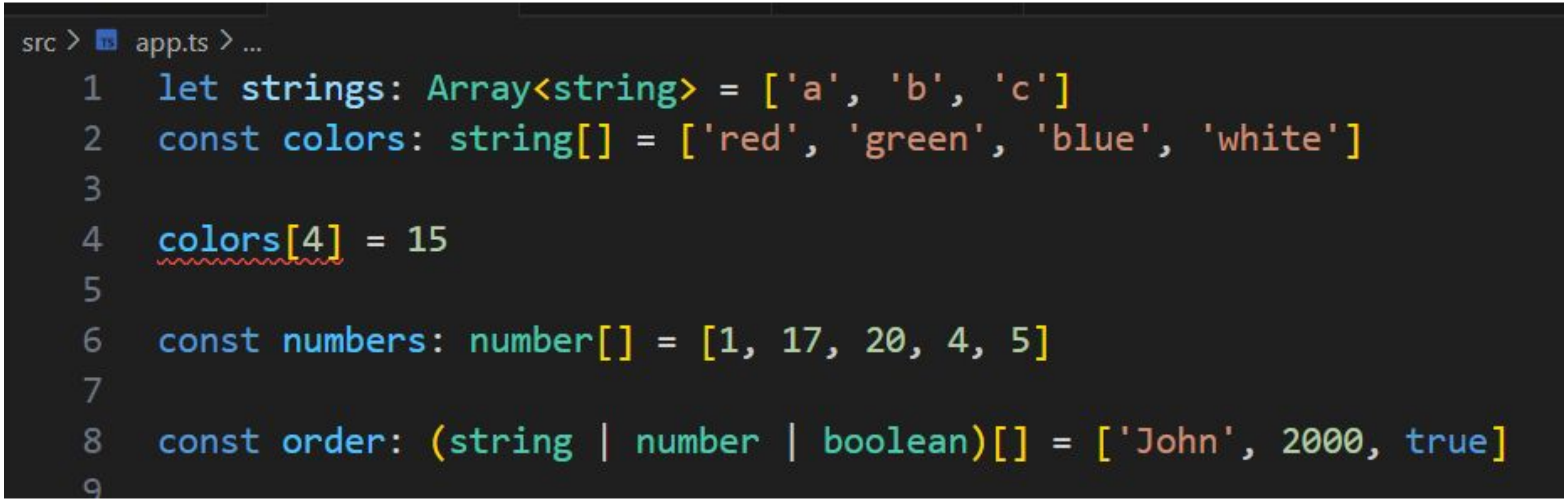
```
let studentName: string = 'Alice'
let age: number = 30
let isPassed: boolean = true
```

Any Type

Using any allows for more flexibility but at the cost of losing type safety.
It disables type checking for the variable it is assigned to.

```
let data: any = 42;
data = "Hello, world!";
data = true;
```

Arrays in Typescript



src > app.ts > ...

```typescript
let strings: Array<string> = ['a', 'b', 'c']
const colors: string[] = ['red', 'green', 'blue', 'white']

colors[4] = 15

const numbers: number[] = [1, 17, 20, 4, 5]

const order: (string | number | boolean)[] = ['John', 2000, true]
```

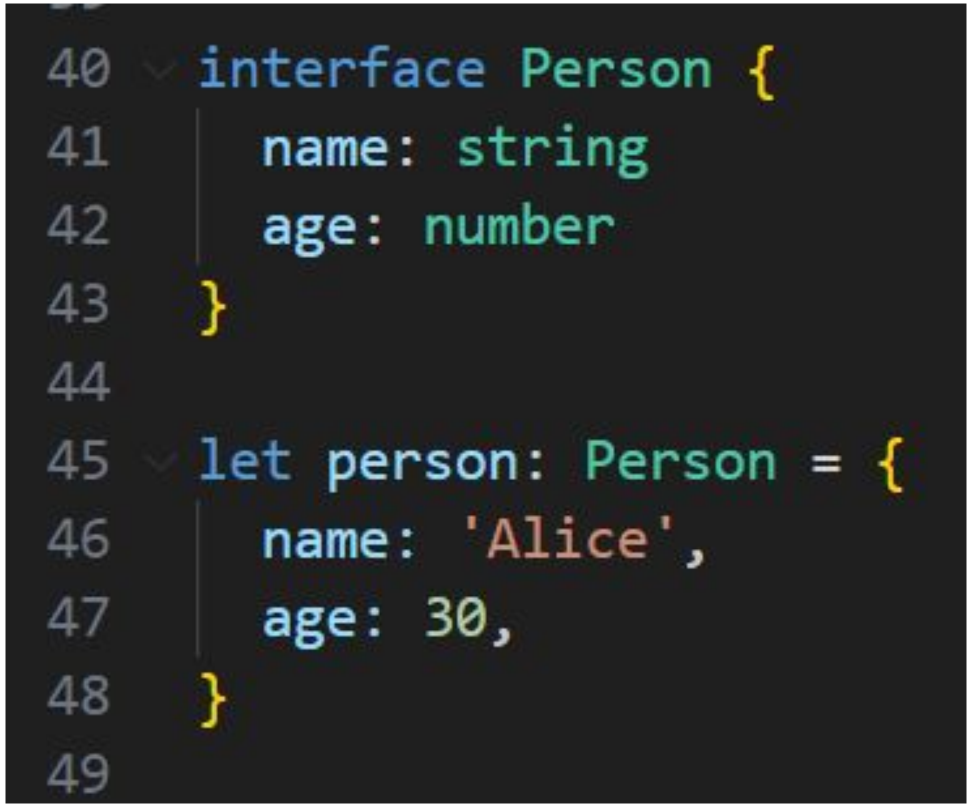
Tuples in Typescript

In TypeScript, a tuple represents a fixed-length array where each element has a specific type.

Tuples are useful if you have two values

```
let employee: [number, string, boolean] = [123, 'Alice', true]
```

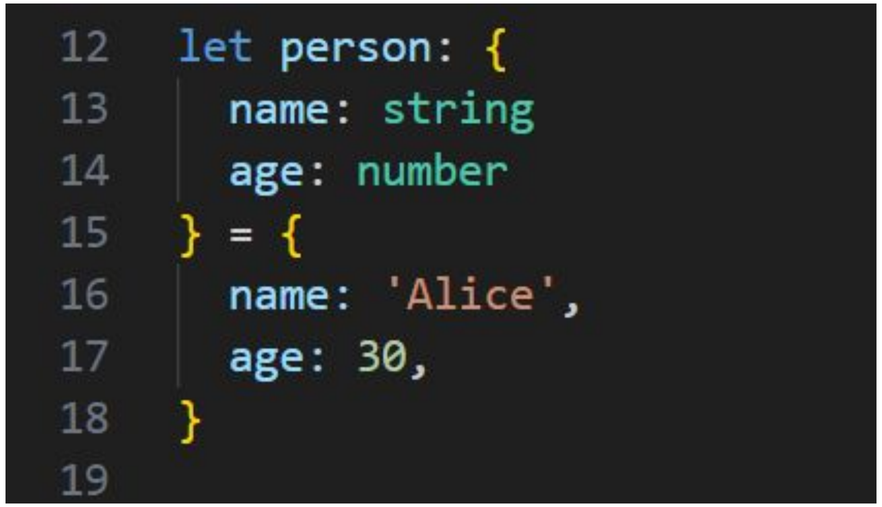
Objects in Typescript and Interface



```
interface Person {
  name: string
  age: number
}

let person: Person = {
  name: 'Alice',
  age: 30,
}

```

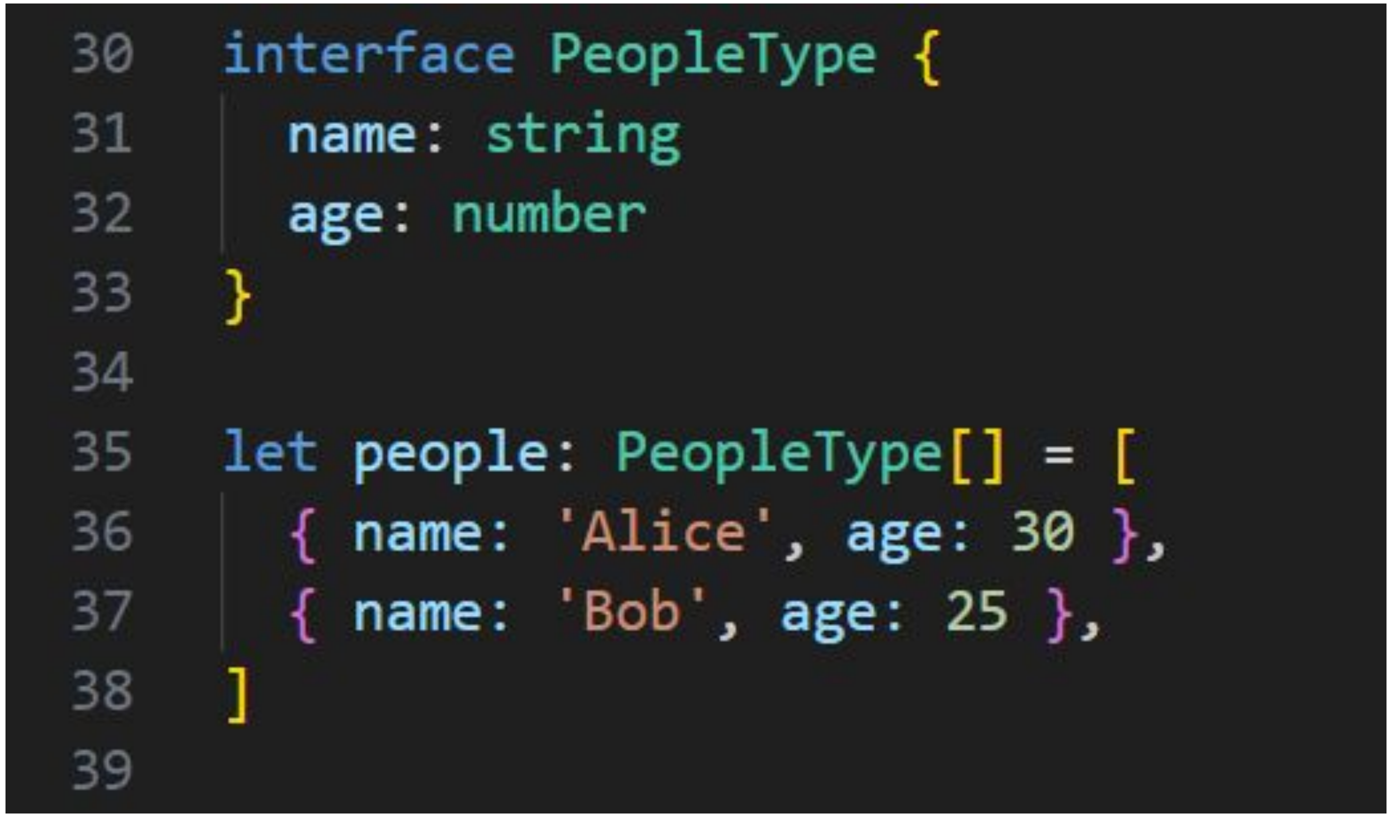
Interfaces improve code readability by documenting the expected properties and methods of objects.

It defines the shape of an object, specifying what properties and methods an object should have.



```
let person: {
  name: string
  age: number
} = {
  name: 'Alice',
  age: 30,
}

```

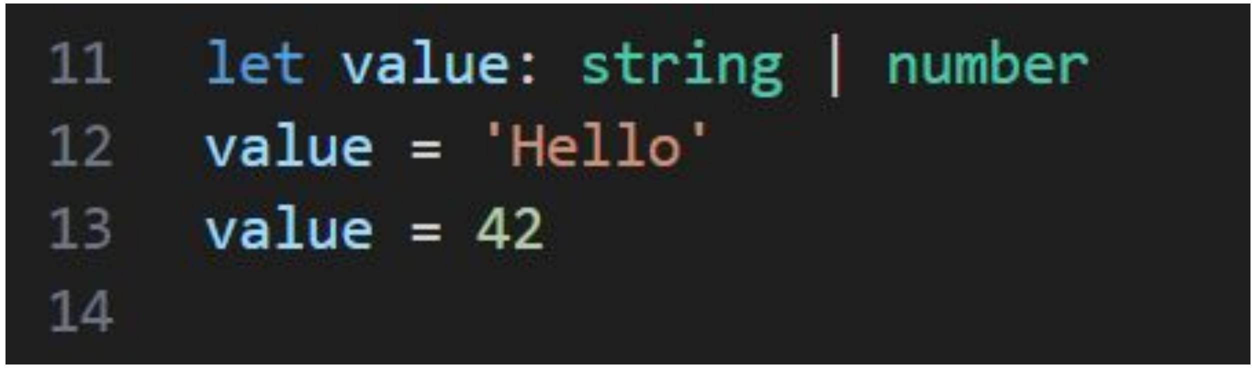
Array Of Objects in Typescript



```
interface PeopleType {
  name: string
  age: number
}

let people: PeopleType[] = [
  { name: 'Alice', age: 30 },
  { name: 'Bob', age: 25 },
]
```

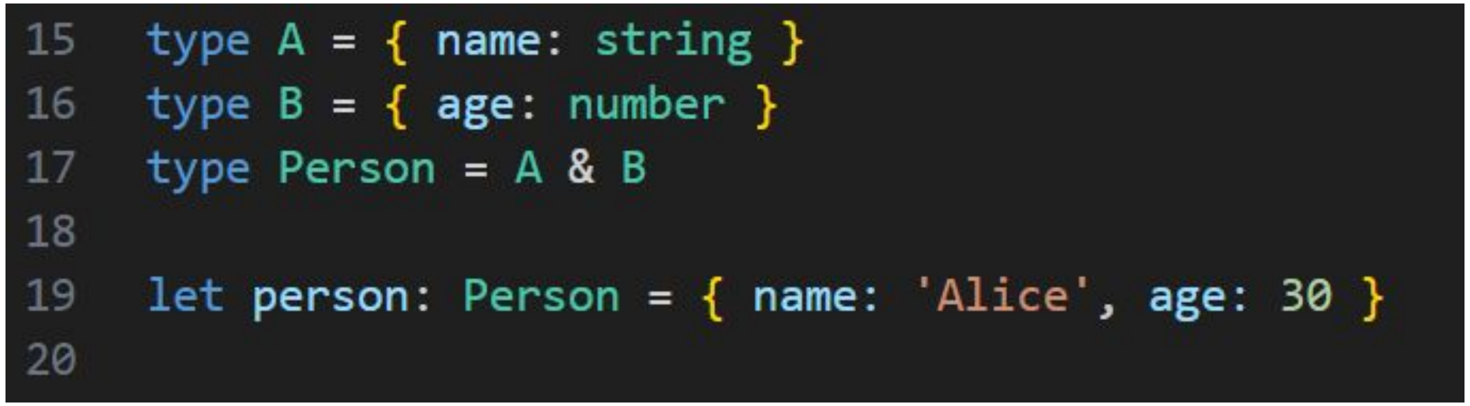
Union and Intersection types

Union Types:
Allows a variable to be one of several types.



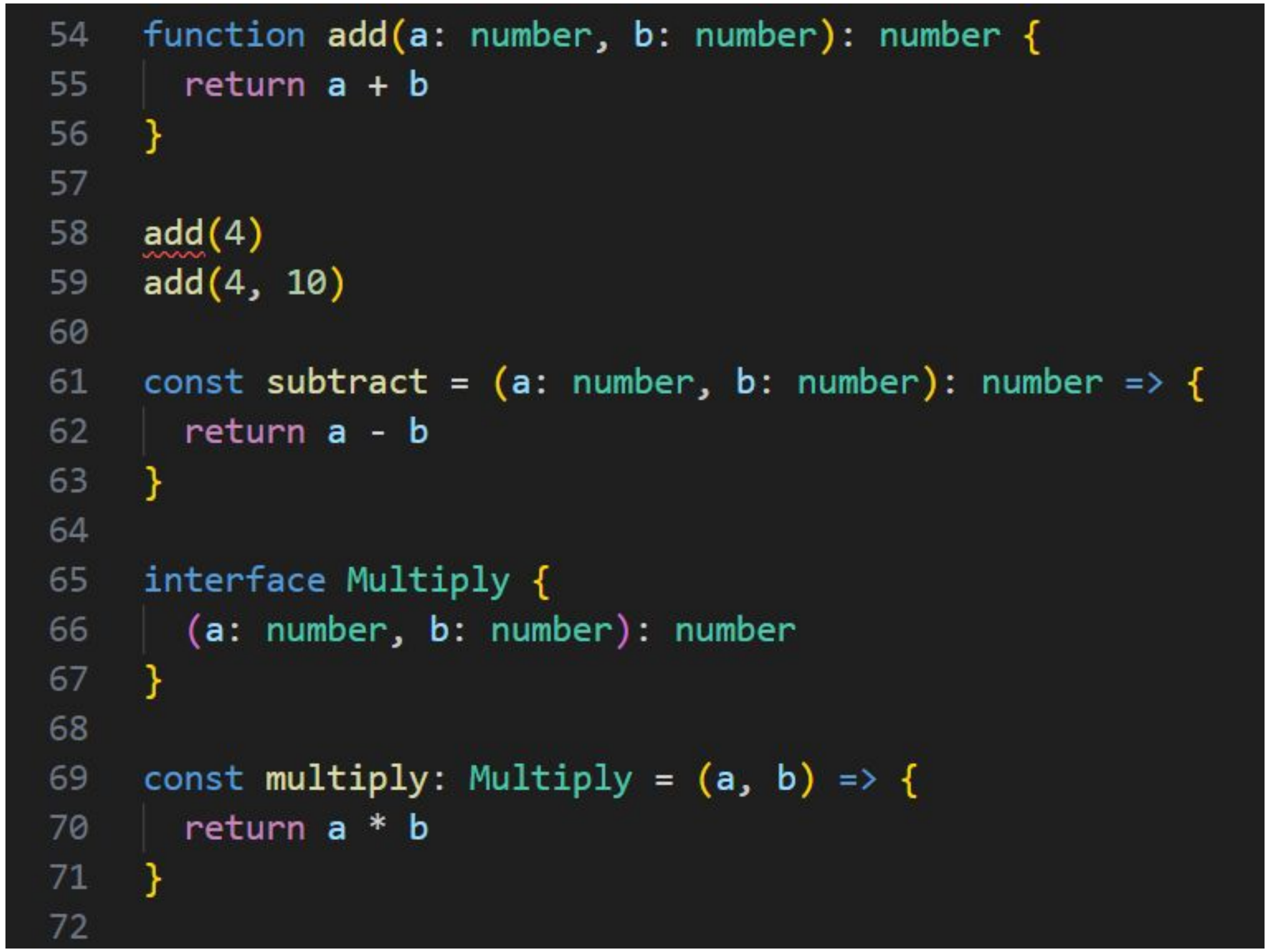
```
let value: string | number
value = 'Hello'
value = 42

```

Intersection Types:
Combines multiple types into one.



```
type A = { name: string }
type B = { age: number }
type Person = A & B

let person: Person = { name: 'Alice', age: 30 }

```

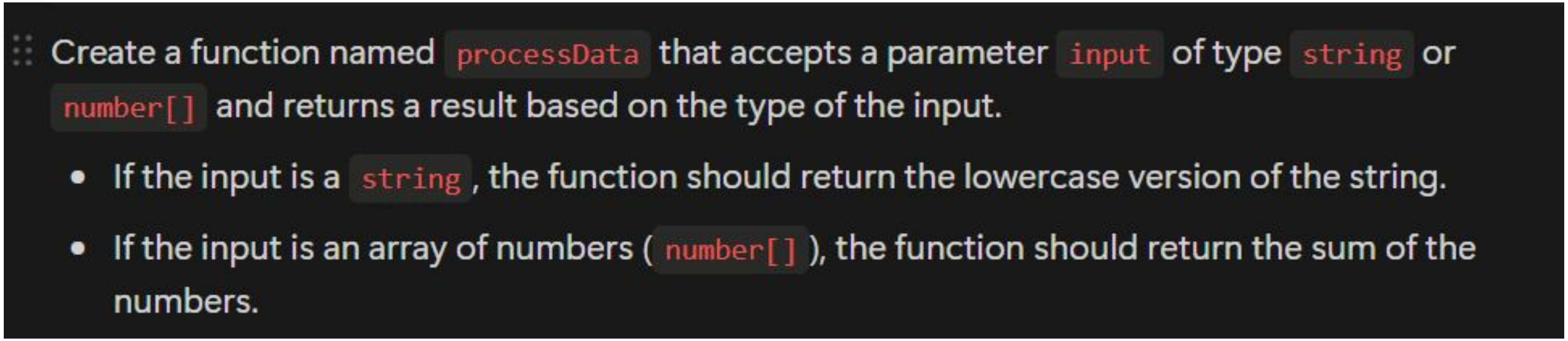
Functions in Typescript



```
function add(a: number, b: number): number {
  return a + b
}

add(4)
add(4, 10)

const subtract = (a: number, b: number): number => {
  return a - b
}

interface Multiply {
  (a: number, b: number): number
}

const multiply: Multiply = (a, b) => {
  return a * b
}

```

Syntax: (parameter: Type) => ReturnType

Typescript Exercise #1

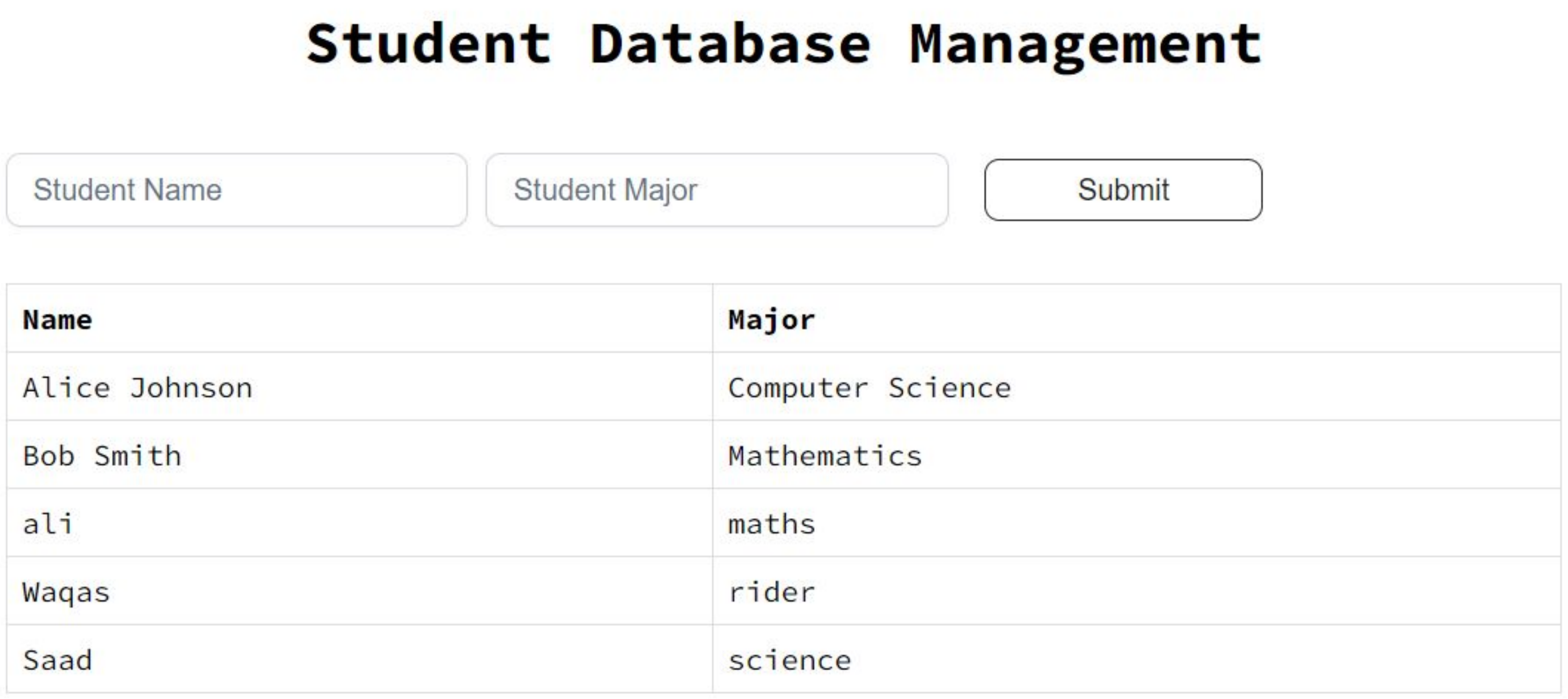

Create a function named `processData` that accepts a parameter `input` of type `string` or `number[]` and returns a result based on the type of the input.

- If the input is a `string`, the function should return the lowercase version of the string.
- If the input is an array of numbers (`number[]`), the function should return the sum of the numbers.

CRUD Application

BAITUSSALAM
—TECH PARK—

Create a CRUD Application with Fetch API



Student Database Management

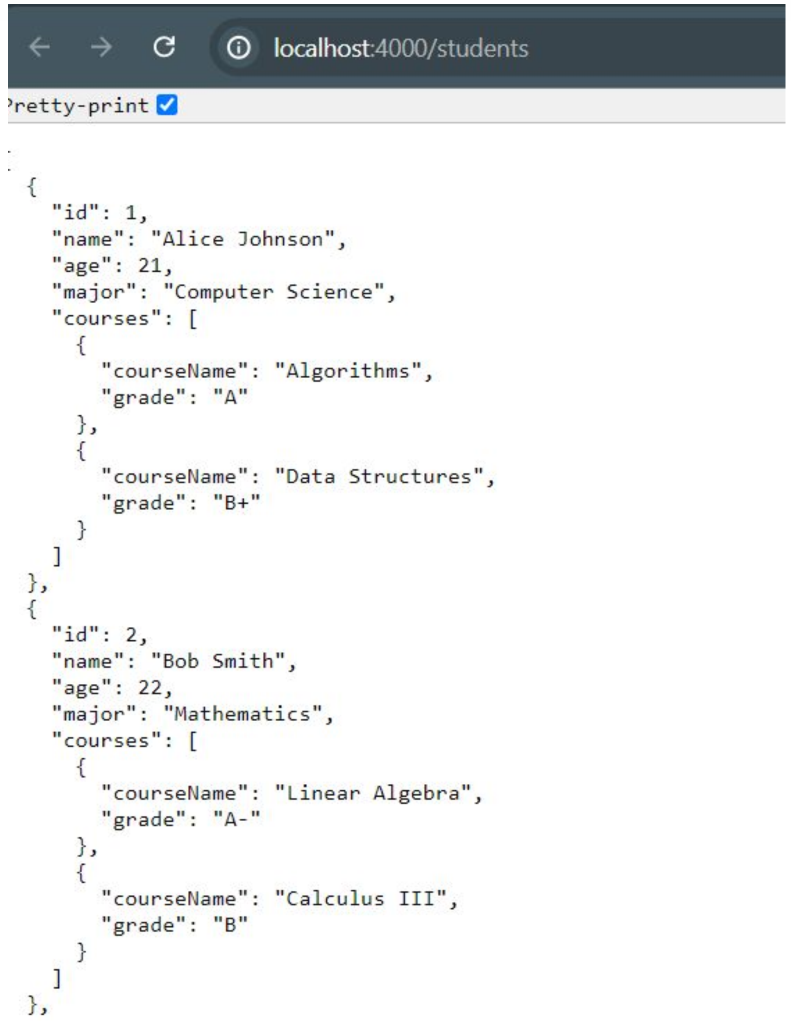
Name	Major
Alice Johnson	Computer Science
Bob Smith	Mathematics
ali	maths
Waqas	rider
Saad	science

JSON Server

JSON Server is a lightweight and easy-to-use Node.js tool that simulates a RESTful API using a JSON file as the data source.

With JSON Server, front-end developers can create mock APIs without the need to write complex server-side code, or when a backend API isn't ready yet.

Running a JSON Server



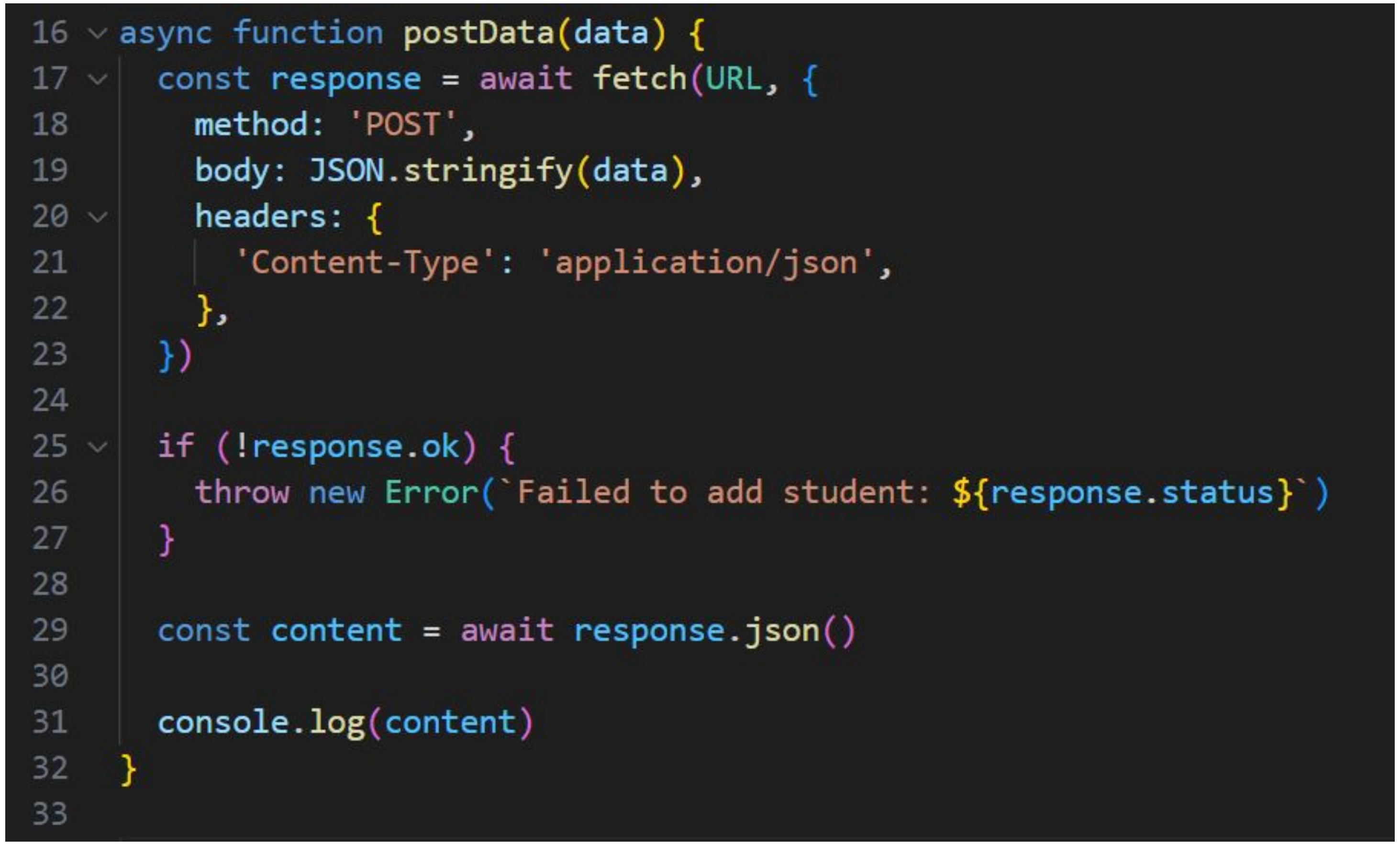
localhost:4000/students

Pretty-print ☑

```
  {
    "id": 1,
    "name": "Alice Johnson",
    "age": 21,
    "major": "Computer Science",
    "courses": [
      {
        "courseName": "Algorithms",
        "grade": "A"
      },
      {
        "courseName": "Data Structures",
        "grade": "B+"
      }
    ]
  },
  {
    "id": 2,
    "name": "Bob Smith",
    "age": 22,
    "major": "Mathematics",
    "courses": [
      {
        "courseName": "Linear Algebra",
        "grade": "A-"
      },
      {
        "courseName": "Calculus III",
        "grade": "B"
      }
    ]
  },
```

- Install json server globally using npm
- Create a **db.json** file
- Insert data in **db.json** file
- Run a json server using command **json-server --watch db.json --port 4000**
- Visit the url **http://localhost:4000/students** in your browser, you can view your API data

Post Request using Fetch



```
async function postData(data) {
  const response = await fetch(URL, {
    method: 'POST',
    body: JSON.stringify(data),
    headers: {
      'Content-Type': 'application/json',
    },
  })

  if (!response.ok) {
    throw new Error(`Failed to add student: ${response.status}`)
  }

  const content = await response.json()

  console.log(content)
}
```

URL: The endpoint you are sending the request to.

Options Object:

method: Specifies the HTTP method (e.g., 'POST').

headers: Includes headers for the request, such as 'Content-Type'.

body: The data being sent with the request, converted to a JSON string.

Blog Application

Build a blog application in JavaScript using Fetch API user will also be able to visit the individual blog page

The End

BAITUSSALAM
—TECH PARK—