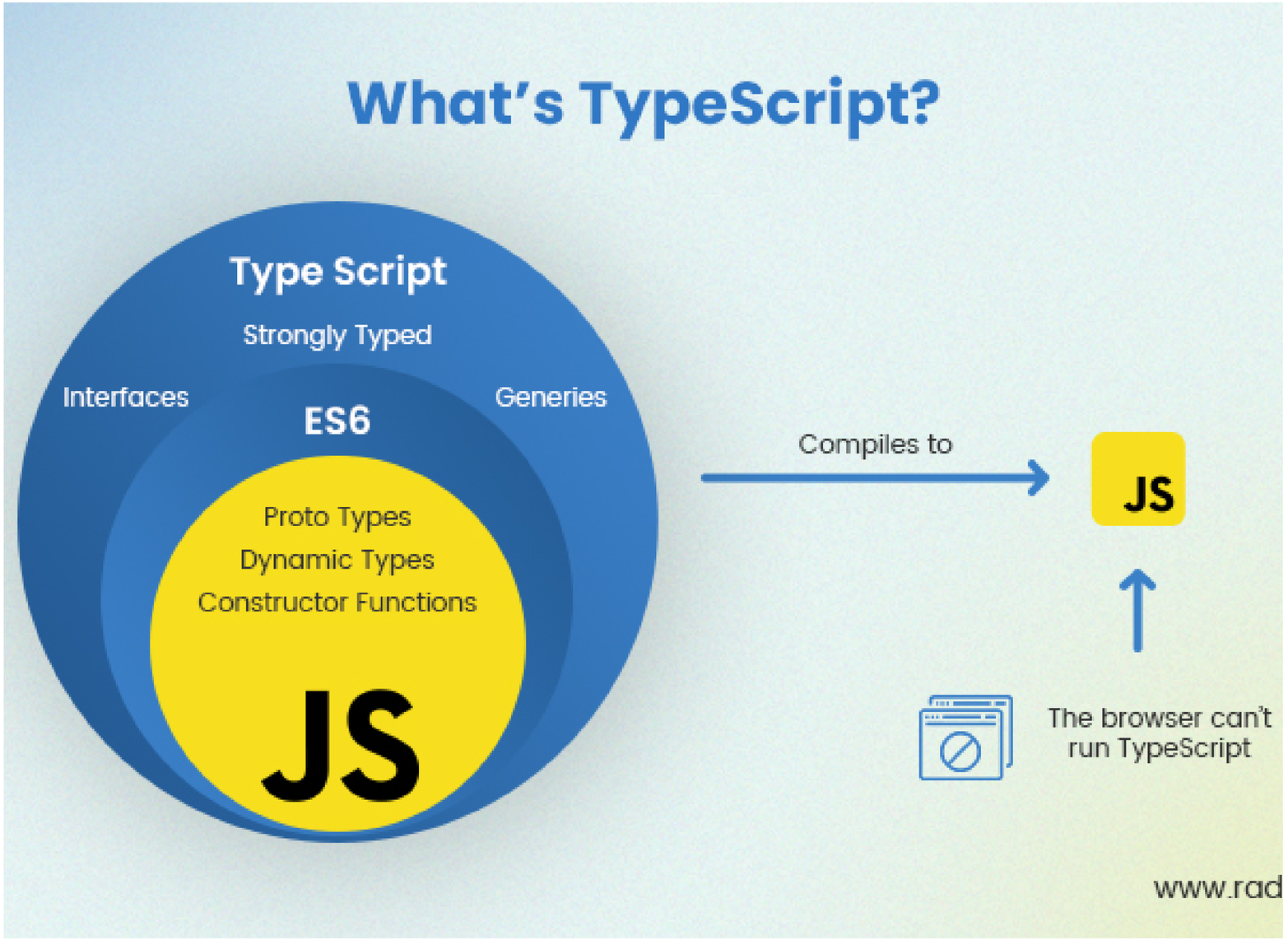
TYPESCRIPT

By GAYATRI

What is typescript?

- JavaScript is not originally designed for large complex applications (mostly a scripting language, with functional programming constructs), lacks structuring mechanisms like Class, Module, Interface.
- Typescript is a typed superset of JavaScript that compiles to plain JavaScript.
- Adds additional features like Static Type (optional), Class, Module etc. to JavaScript
- Microsoft technology.
- Open Source.
- Versions.
    - First made public in October 2012.
    - Latest version - Typescript 1.7.


What's TypeScript?
Type Script
Strongly Typed
Interfaces
Generies
ES6
Proto Types
Dynamic Types
Constructor Functions
JS
Compiles to
JS
The browser can't run TypeScript
www.rad

Typescript Features

Type Annotations
Compile Time Checking
Classes
Inheritance
Interfaces
Modules
Enums
Generics
Function Expressions

Type Annotation

- Any
  - Any Type is a super set of all types
    - var x : any;
    - var y;
- Primitive
  - Number
    - Does not have separate integer and float/double type.
    - var num : number = 20;
    - var num = 20;
  - String
    - Both single quote or double quotes can be used.
    - var name : string = “hello”;
    - var name =’hello’;
  - Bool
    - var isOpen =true;

Type Annotation

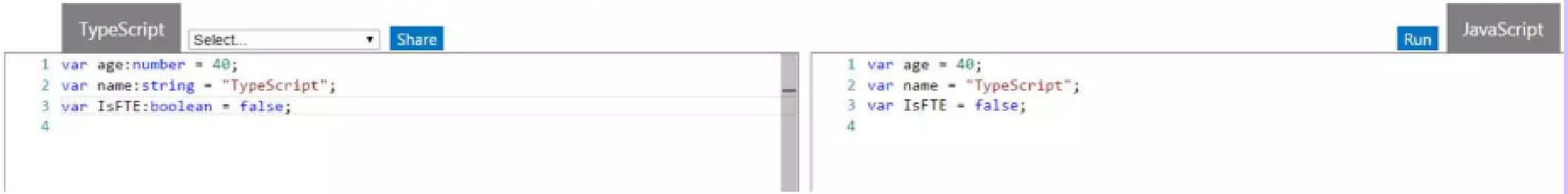
- Void
  - Used as the return type of functions that don’t return any value
- Object Types
  - class, interface, module.
- Array
  - Array types can be written in:
    - var list: number[] = [1, 2, 3];
    - var list: Array<number> = [1, 2, 3];
    - var list:any[] = [1, true, "free"]
- Enum
    - enum Color { Red, Green, Blue };
    - var color = Color.Blue;

Type Annotation

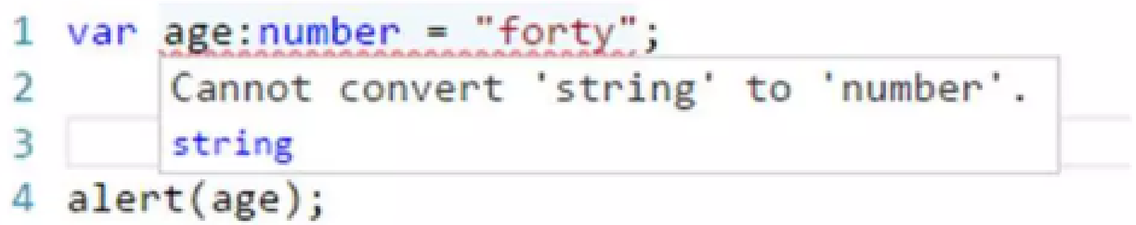
- Tuple
  - Tuple types allow you to express an array where the type of a fixed number of elements is known.
    - var x: [string, number];
      x = ['hello', 10];

Type Annotation

- Design time feature. No additional code is emitted in the final JavaScript that TypeScript compiler produces.



- If there's a type mismatch TypeScript shows a warning.



Inheritance

- Typescript supports inheritance of class through extends keyword
- super keyword.

Class

- Properties and fields to store data
- Methods to define behavior

TypeScript

Select...

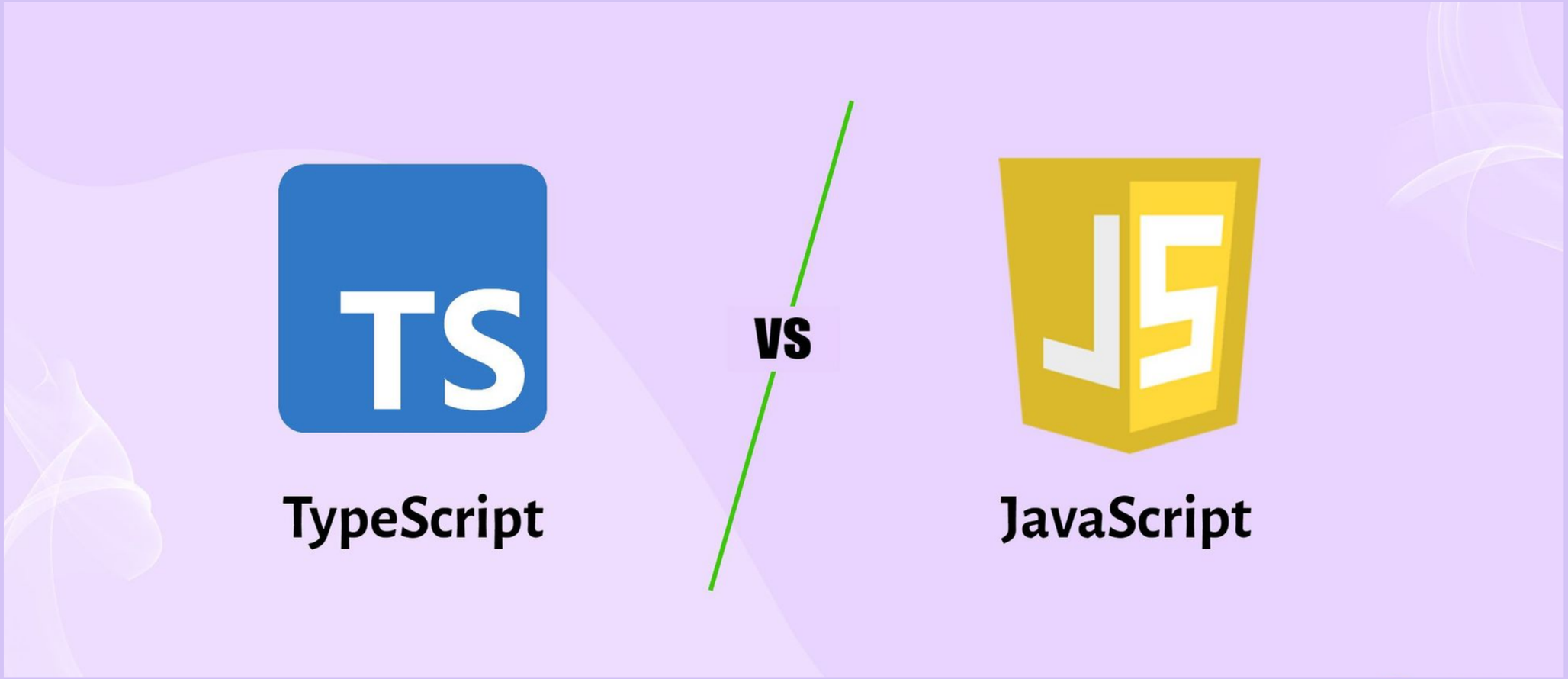
Share

```typescript
class Employee{
    private name:string
    private basic:number
    private allowance:number

    constructor(name:string, basic:number, allowance:number){
        this.name = name
        this.basic = basic
        this.allowance = allowance
    }

    public getSalary():number{
        return this.basic + this.allowance
    }
}

var emp = new Employee("Aniruddha",100,20)
alert(emp.getSalary())
```

Run

JavaScript

```javascript
var Employee = (function () {
    function Employee(name, basic, allowance) {
        this.name = name;
        this.basic = basic;
        this.allowance = allowance;
    }
    Employee.prototype.getSalary = function () {
        return this.basic + this.allowance;
    };
    return Employee;
})();

var emp = new Employee("Aniruddha", 100, 20);
alert(emp.getSalary());
```


TS
VS
TypeScript
JavaScript

JAVASCRIPT VS TYPESCRIPT

Parameters	JavaScript	TypeScript
Designed & Developed	Brendan Eich at Netscape Communications Corpora, Mozilla Foundation, ECMA International in 1995	Anders Hejlsberg at Microsoft in 2012
Extension	JavaScript source file is in ".js" extension.	TypeScript source file is in ".ts" or ".tsx" extension.
Syntax	All statements are written inside of the Script tag. It requests the browser program to interpret and execute all the text that comes between these tags like a script. <script>//javascript code</script>	A typescript program comprises Functions, Modules, Statement & Expressions, Variables, and Comments.
Annotations	Annotations not required for coding in JavaScript	Code must be annotated constantly to get the most out of TypeScript Features.

Example	<script>
	function addNumbers(a, b) {
	return a + b;
	}
	var sum = addNumbers(15, 25);
	document.write('Sum of the numbers is: ' + sum);
	</script>
		function addNumbers(a, b) {
		return a + b;
		}
		var sum = addNumbers(15, 25);
		console.log('Sum of the numbers is: ' + sum);

TypeScript syntax

```
class HelloWorld {

    adjective: string;

    constructor(adj: string) {

        this.adjective = adj;

    }

    sayHello(): string {

        return "Hello " + this.adjective +
        " world!";

    }

}

let helloWorldMessage = new
HelloWorld("beautiful");

console.log(helloWorldMessage.sayHello());
```

Java syntax

```
public class HelloWorld {

    String adjective;

    HelloWorld(String adj) {

        this.adjective = adj;

    }

   public String  sayHello() {

        return "Hello " + this.adjective +
        " world!";

    }

}

HelloWorld helloWorldMessage = new
HelloWorld("beautiful");

console.log(helloWorldMessage.sayHello());
```


thank
you