

TypeScript Basics



What is TypeScript?

What is TypeScript?

- Language developed by Microsoft in 2012

What is TypeScript?

- Language developed by Microsoft in 2012
 - Free and open-source

What is TypeScript?

- Language developed by Microsoft in 2012
 - Free and open-source
- Provides static typing support to JavaScript

What is TypeScript?

- Language developed by Microsoft in 2012
 - Free and open-source
- Provides static typing support to JavaScript
 - Helps with IDE support: code completion and debugging

What is TypeScript?

- Language developed by Microsoft in 2012
 - Free and open-source
- Provides static typing support to JavaScript
 - Helps with IDE support: code completion and debugging
- Add support for object-oriented programming

What is TypeScript?

- Language developed by Microsoft in 2012
 - Free and open-source
- Provides static typing support to JavaScript
 - Helps with IDE support: code completion and debugging
- Adds support for object-oriented programming
 - Classes, objects, inheritance, interfaces, etc ...

What is TypeScript?

- Language developed by Microsoft in 2012
 - Free and open-source
- Provides static typing support to JavaScript
 - Helps with IDE support: code completion and debugging
- Adds support for object-oriented programming
 - Classes, objects, inheritance, interfaces, etc ...

www.typescriptlang.org

Angular Development

Angular Development

- For Angular development, we can develop using various languages

Angular Development

- For Angular development, we can develop using various languages
 - **JavaScript**: extremely popular programming language

Angular Development

- For Angular development, we can develop using various languages
 - **JavaScript**: extremely popular programming language
 - **ECMAScript**: standardized version of JavaScript (ES6, ES9, ...)

Angular Development

- For Angular development, we can develop using various languages
 - **JavaScript**: extremely popular programming language
 - **ECMAScript**: standardized version of JavaScript (ES6, ES9, ...)
 - **TypeScript**: adds optional types to JavaScript

Angular Development

- For Angular development, we can develop using various languages
 - **JavaScript**: extremely popular programming language
 - **ECMAScript**: standardized version of JavaScript (ES6, ES9, ...)
 - **TypeScript**: adds optional types to JavaScript
 - *Other languages such as Dart, etc ...*

Angular Development

- For Angular development, we can develop using various languages
 - **JavaScript**: extremely popular programming language
 - **ECMAScript**: standardized version of JavaScript (ES6, ES9, ...)
 - **TypeScript**: adds optional types to JavaScript
 - *Other languages such as Dart, etc ...*
- TypeScript is the most popular language for Angular development

Relationships

Relationships

- TypeScript is a superset of JavaScript and ECMAScript

Relationships

- TypeScript is a superset of JavaScript and ECMAScript



JavaScript

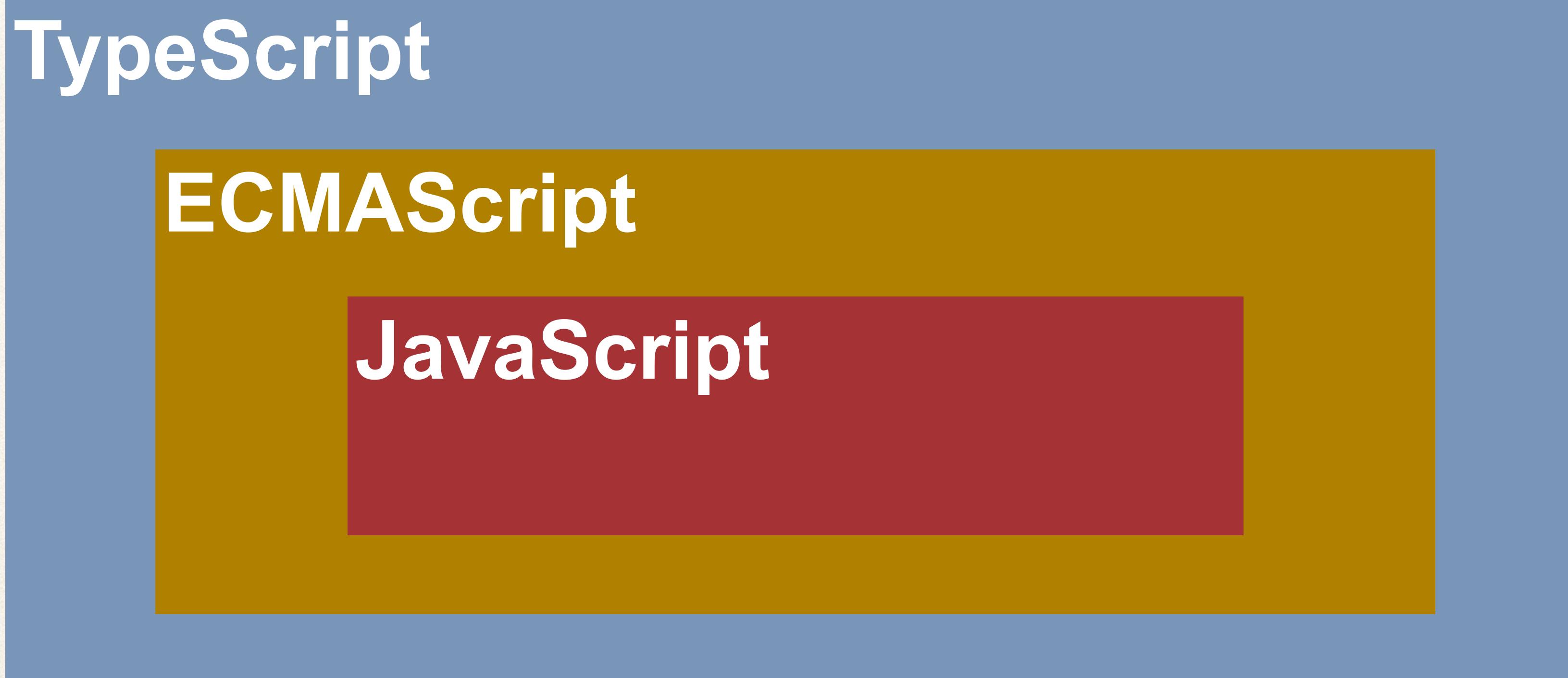
Relationships

- TypeScript is a superset of JavaScript and ECMAScript



Relationships

- TypeScript is a superset of JavaScript and ECMAScript



TypeScript

TypeScript

- FAQ: Why do most Angular developers use TypeScript?

TypeScript

- FAQ: Why do most Angular developers use TypeScript?
- Strongly-typed language with compile time checking and IDE support

TypeScript

- FAQ: Why do most Angular developers use TypeScript?
- Strongly-typed language with compile time checking and IDE support
- Increased developer productivity and efficiency

TypeScript

- FAQ: Why do most Angular developers use TypeScript?
- Strongly-typed language with compile time checking and IDE support
- Increased developer productivity and efficiency
- The Angular framework is internally developed using TypeScript

TypeScript

- FAQ: Why do most Angular developers use TypeScript?
- Strongly-typed language with compile time checking and IDE support
- Increased developer productivity and efficiency
- The Angular framework is internally developed using TypeScript
- Docs, online blogs and tutorials use TypeScript for coding examples

Practical Results

Practical Results

- Introduction to TypeScript development

Practical Results

- Introduction to TypeScript development
- Not an A to Z reference

Practical Results

- Introduction to TypeScript development
- Not an A to Z reference
- For complete reference, see TypeScript Documentation

Practical Results

- Introduction to TypeScript development
- Not an A to Z reference
- For complete reference, see TypeScript Documentation

www.typescriptlang.org

Development Process

Step-By-Step

Development Process

Step-By-Step

1. Create TypeScript code

Development Process

Step-By-Step

1. Create TypeScript code
2. Compile the code

Development Process

Step-By-Step

1. Create TypeScript code

2. Compile the code

3. Run the code

Step 1: Create the TypeScript code

Step 1: Create the TypeScript code

- TypeScript files have the .ts extension

Step 1: Create the TypeScript code

- TypeScript files have the .ts extension

File: mydemo.ts

```
console.log("Hello World!");
```

Step 2: Compile the Code

Step 2: Compile the Code

- Web browsers do not understand TypeScript natively

Step 2: Compile the Code

- Web browsers do not understand TypeScript natively
- Have to convert TypeScript code to JavaScript code

Step 2: Compile the Code

- Web browsers do not understand TypeScript natively
- Have to convert TypeScript code to JavaScript code
- This is known as "**transpiling**"

Step 2: Compile the Code

- Web browsers do not understand TypeScript natively
- Have to convert TypeScript code to JavaScript code
- This is known as "**transpiling**"

TypeScript

mydemo.ts

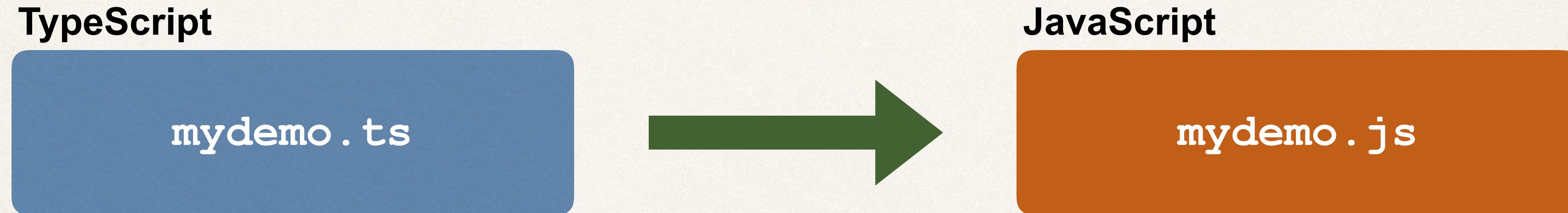
Step 2: Compile the Code

- Web browsers do not understand TypeScript natively
- Have to convert TypeScript code to JavaScript code
- This is known as "**transpiling**"



Step 2: Compile the Code

- Web browsers do not understand TypeScript natively
- Have to convert TypeScript code to JavaScript code
- This is known as "**transpiling**"



Step 2: Compile the Code (cont)

Step 2: Compile the Code (cont)

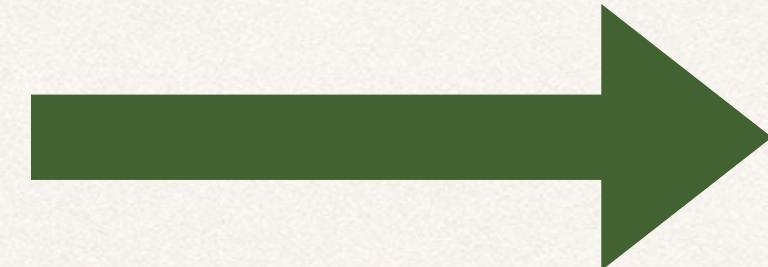
- "Transpiling" is accomplished with the **tsc** command

Step 2: Compile the Code (cont)

- 'Transpiling' is accomplished with the **tsc** command

TypeScript

`mydemo.ts`

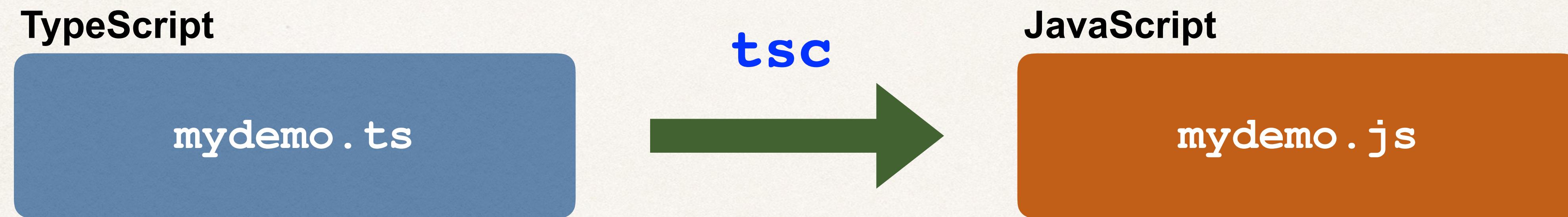


JavaScript

`mydemo.js`

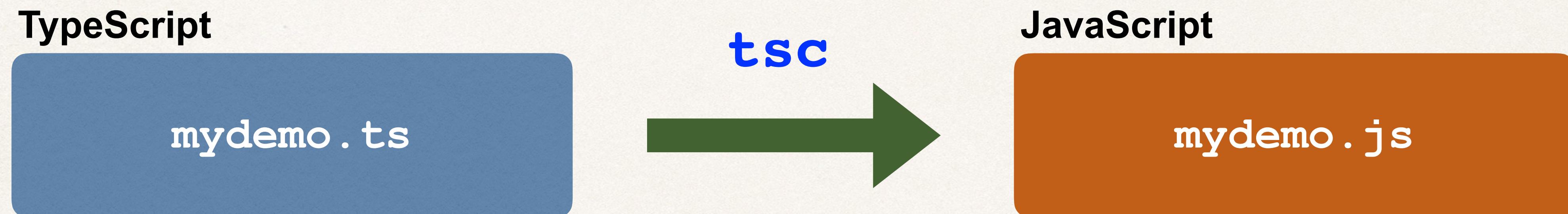
Step 2: Compile the Code (cont)

- 'Transpiling' is accomplished with the **tsc** command



Step 2: Compile the Code (cont)

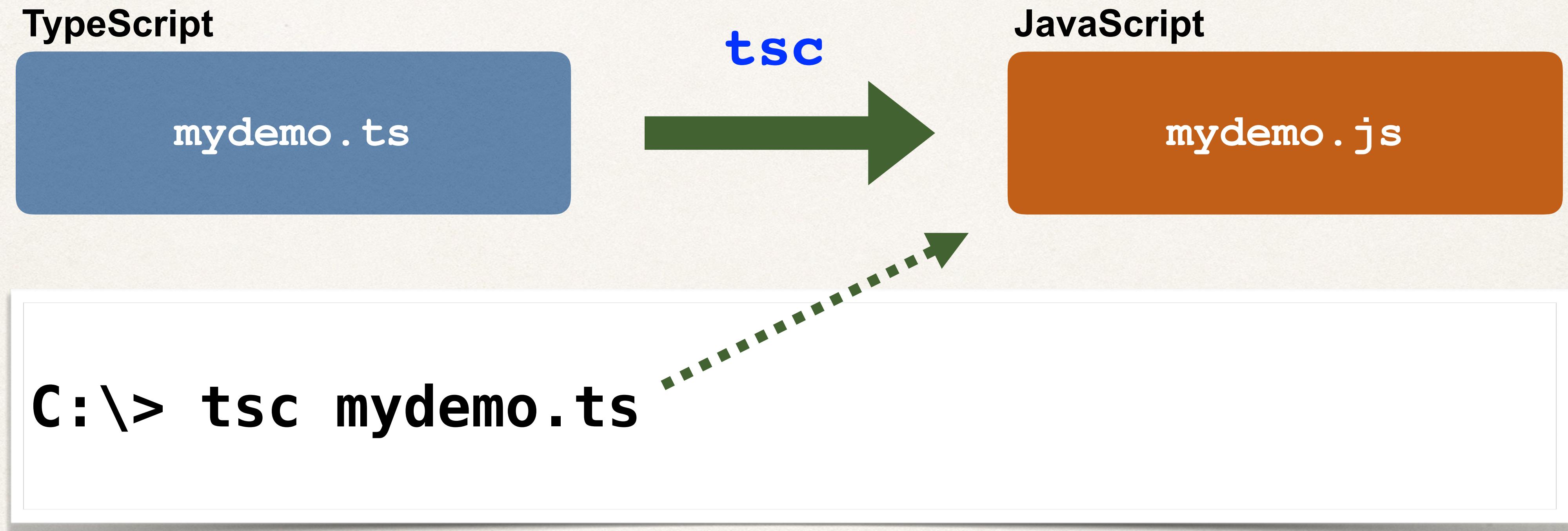
- 'Transpiling' is accomplished with the **tsc** command



```
C:\> tsc mydemo.ts
```

Step 2: Compile the Code (cont)

- 'Transpiling' is accomplished with the **tsc** command



Step 3: Run the code

Step 3: Run the code

- To run the JavaScript code, we use the **node** command

Step 3: Run the code

- To run the JavaScript code, we use the **node** command
- Run the **generated JavaScript code (.js file)**

Step 3: Run the code

- To run the JavaScript code, we use the **node** command
- Run the **generated JavaScript code (.js file)**

```
C:\> node mydemo.js
```

```
Hello World!
```

Step 3: Run the code

- To run the JavaScript code, we use the **node** command
- Run the **generated JavaScript code (.js file)**

```
C:\> node mydemo.js
```

Hello World!

```
console.log("Hello World!");
```

The Compiler is Your Friend

The Compiler is Your Friend

- The compiler / IDE can find errors earlier at compilation time

The Compiler is Your Friend

- The compiler / IDE can find errors earlier at compilation time

```
console.LOGSTUFF("Hello World!");
```

The Compiler is Your Friend

- The compiler / IDE can find errors earlier at compilation time

```
console.LOGSTUFF("Hello World!");
```

Compile code using: tsc

```
C:\> tsc mydemo.ts
```

The Compiler is Your Friend

- The compiler / IDE can find errors earlier at compilation time

```
console.LOGSTUFF("Hello World!");
```

Compile code using: tsc

```
C:\> tsc mydemo.ts
```

```
myhello.ts:1:9 - error TS2339: Property 'LOGSTUFF' does not exist on type 'Console'.
```

```
console.LOGSTUFF("Hello World!!");  
~~~~~
```

```
Found 1 error.
```

Compilation error ... much better