TypeScript

Typescript is a typed superset of JavaScript that compiles to plain JavaScript.

Can run any browser, any host, Any OS, Open Source

Typescript Compilation

Type Script ----🡪 compiled ----🡪 JavaScript

JavaScript -> scripting language but typescript -> OOO language

Why Typescript instead of javascript?

1. More maintainable code
2. More message JavaScript code
3. Javascript has dynamic type system(var a=1,var a=”test”). But that cuases confusion
4. Migrating from server side to client side apps can be challenging. (java to javascript is tough)
5. There are alternatives for javascript like CofeeScript or Dart, or you can write pure javascript or follow all javascript patterns

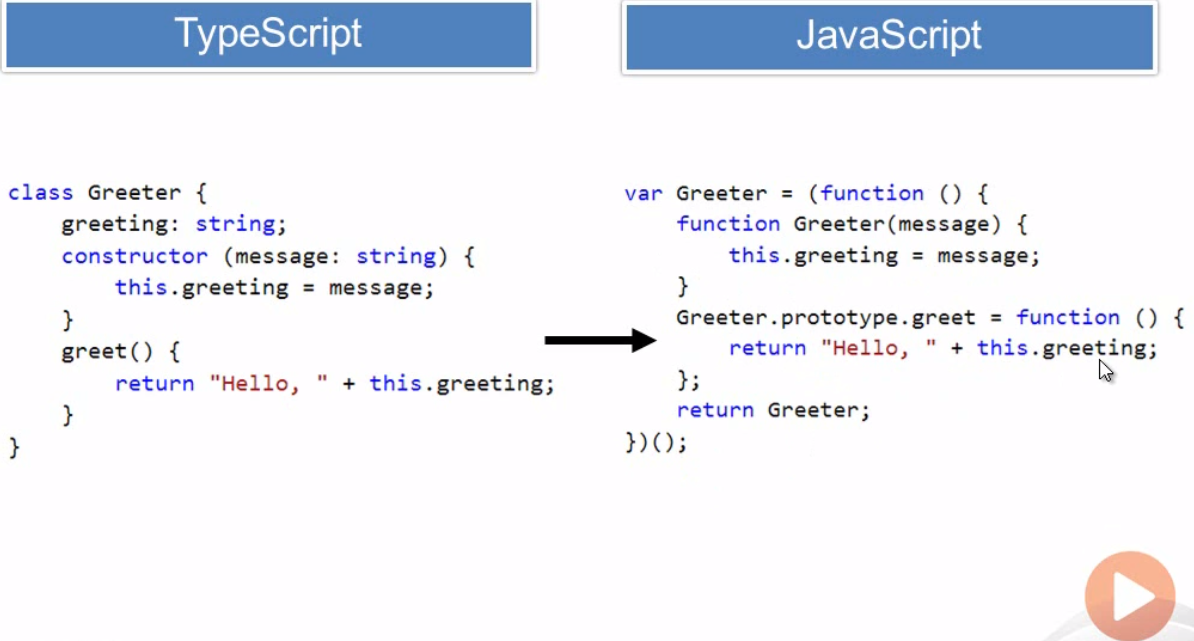
Typescript Features

1. Typescript will work in any browser
2. Work on any host.
3. Any OS, Open source, Good tool support
4. Typescript supports standard typescript code(typescript and javascript both )
5. Static typing (data typing)
6. Encapsulation through class modules(classes)
7. Support constructor, properties and functions
8. Supports Interfaces
9. => lamda functions support (anonymous functiosn)
10. Intellisense and syntax checking

Typescript Installation

1. Installation -> Npm install g typescript
2. To compile any typescript => tsc Customer.ts

Typescript vs Javascript



Typescript Keywords and Operators

Class – class

Constructor – initialize

Exports- exports a member from module

Imports – imports from module

Implements – implement interface

Extends – extends a class or interface

Public/private

Module/namespace – container for class and other code

… - rest parameter syntax

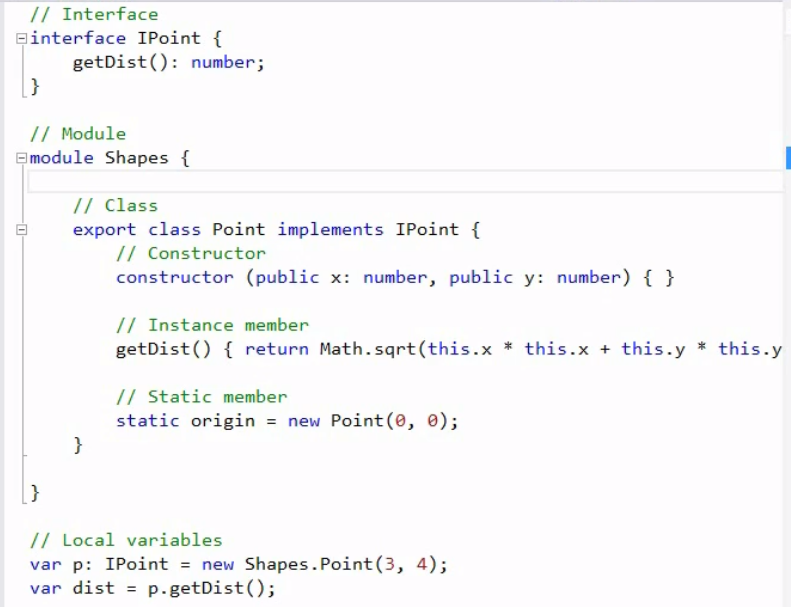
* - used with defintions and functions

<typename> - character used to cast/convert between types

: seperateor between variable and parameter and types

Code Hierarchy

Module/Namespace -> class -> fields, consturcot,r properties,functions



Basics of Typescript

1. Variables & datatypes
2. Inheritance: extends

class NewCustomer extends Customer {

1. Methods

validate(input:number): boolean {

alert(input +'i am in validate function');

return true;

}

}

4) get/set identifier

Private name:string

set CustomerName(ame:string){

this.name = ame;

}

get CustomerName():string{

return this.name;

}

This will work only ECMA 5 or higher

5) Exception handling

Throw in typescript

if (input >10) {

throw "exeption nae is required";

}

Catch in html

try{

var customernew = new NewCustomer();

customer.validate(33);

}

catch (ex ){

alert(ex);

}

6) interfaces

Class customer impelments Icustomer

7) class modules and reference them using export

export class Address {

Street1: string;

Street2: string;

}

import { Address} from './Address';

class Customer {

name: string;

public age: number;

address: Address;

All javascript code are valid inside typescript