Page No.
Date

Typescript.

→ Open source, object oriented, Developed and maintained by Microsoft.

→ Licensed under Apache 2

→ Javascript but statically typed

Typescript code (Typescript file) (*.ts)

↓

Compilation / Transpilation — Typescript compiler (tsc)

↓

Vanilla JS code — Javascript file (runs everywhere)

Design Goals of Typescript.

* Statically identify Javascript constructs at compile time, that are likely to be errors
* Be a cross platform development tool.
* Impose no runtime overhead on emitted programs.
* High compatibility with existing Javascript code.
* Provide a structuring Mech. for larger pieces of code.
* Align with current and future ECMAScripts proposals.

Typescript

- Language → consists of keywords, syntax and annotations
- Ts compiler → converts or transpiles instructions written in Ts to its Javascript equivalent.
- Language service → assist common editor operations including statement completion, sign help, code format and so on

```
function add(a:any, b:any): any {
    if (typeof a == 'number' && typeof b == 'number') {
        return a+b;
    }
    if (typeof a == 'string' && typeof b == 'string') {
        return a.concat(b)
    }
}
console.log(add(3,6));
console.log(add("Hello", "world"));
```

Javascript V/s Typescript

JS

- It is dynamically typed
- Netscape develop in 1995
- .js
- can be directly run on the browser
- It is just a scripting language.
- numbers and strings are objects
- code Example.

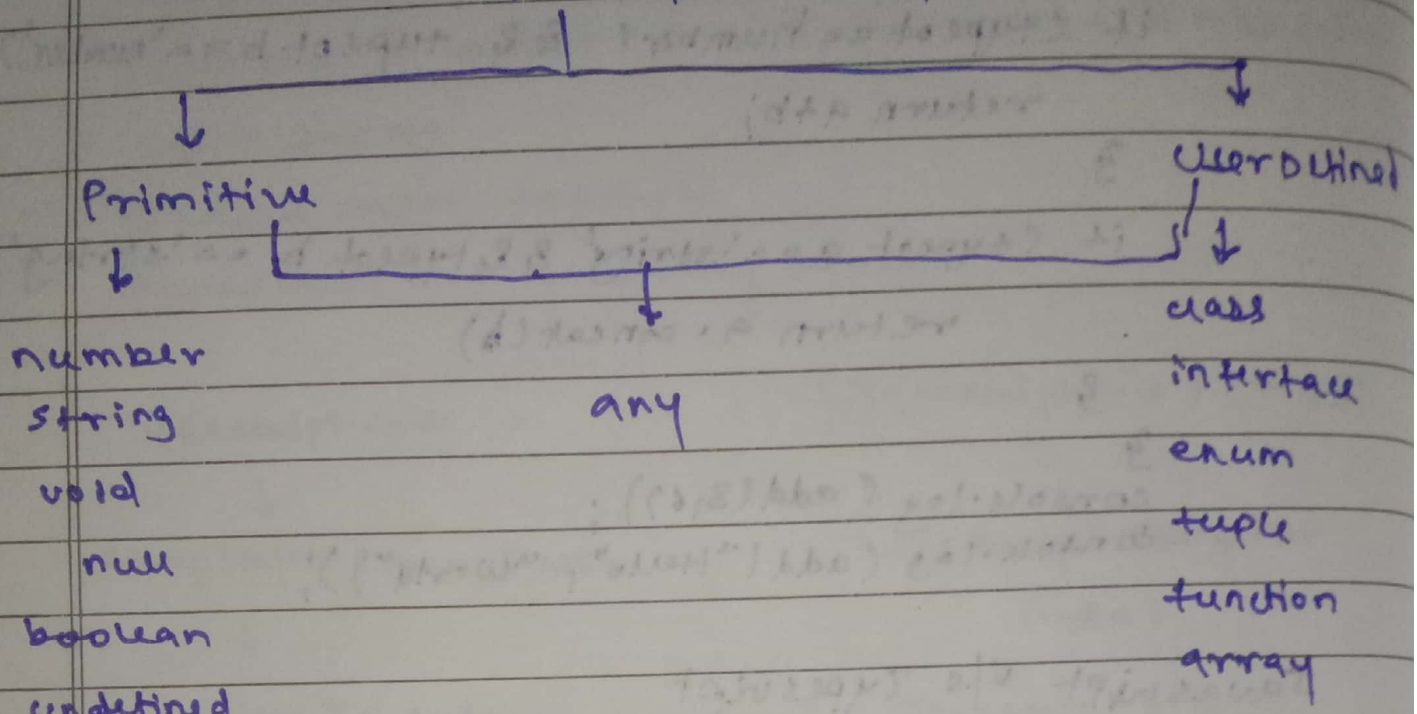
TS

- It is strongly typed.
- Google in 2012
- .ts
- cannot be directly run on the browser
- supports OOP, classes, interfaces.
- number, string are interface
- code Example.

[keyword] [variable_name] : [type_annotation] = value ;

Page No.
Date

TypeScript Datatypes



Typescript operators :-

- Arithmetic
- comparison
- Logical
- Bitwise
- Assignment
- Ternary
- concatenation
- Advanced type

- in → used to check for existence of property on an object

```
let Bike = { make: "Honda", model: "CLIQ" };
console.log('make' in Bike); // output: true.
```

- delete → used to delete properties from Bike.
- typeof → returns the data type of
- instanceof → used to check if object is of specified type or not

Decision Making:-

if statement;

if..else statement;

switch;

definite loop;

while loop;

do..while loop;

break;

continue;

Functions:-

function func_name ([parameters]);

1. Named function
2. Anonymous function

```
let myAdd = function(x:number, y:number): number {
    return x+y;
};
console.log(myAdd(2,3));
```

- Optional Parameters
- Default Parameters
- Rest Parameter

*Overloads:-

In many languages each overload has its own implementation, but in TS the overloads all decorate a single implementation.

```
//function overload for string param
function getAvg(a:string, b:string, c:string): string;
//function overload for number params
function getAvg(a:number, b:number, c:number): number;
//implemen.
function getAvg(a:string|num, b:string|num, c:string|num): str {
    var total = parseInt(a as string,10) + parseInt(b as string,10);
    var avg = total/3;
    return 'The avg is' + 'avg';
}
```

Page No.
Date

Arrow Functions:-

Also called as Lambda Function.

Typescript classes and Objects:-

A class in typescript is compiled to plain Javascript functions by Typescript compiler to work across platforms and browsers.

A class definition can contain following properties:-

1. Fields
2. Methods
3. constructors
4. Nested class and Interfaces.

Object Initialization:-

1. By reference variable
2. By Method
3. By constructor.

** Typescript supports only single Inheritance and multilevel Inheritance.

** We can use inheritance for Method Overriding (so runtime polymorphism is achieved) and code reusability.

TypeScript Interfaces:-

→ Acts as an contract in our application.
→ used to enforce the implementation of specified properties or methods of an object.
→ we cannot instantiate the interface but can be referenced by class object that implements it.
→ contains only 'declaration' of 'methods' and 'fields', but not implementation.

TypeScript Modules:-

- module is a way to create a group of variable, function, classes and interfaces, etc.
- Executes in local scope.
- create using export keyword, use in other module using import keyword.
- import another module using module loader.

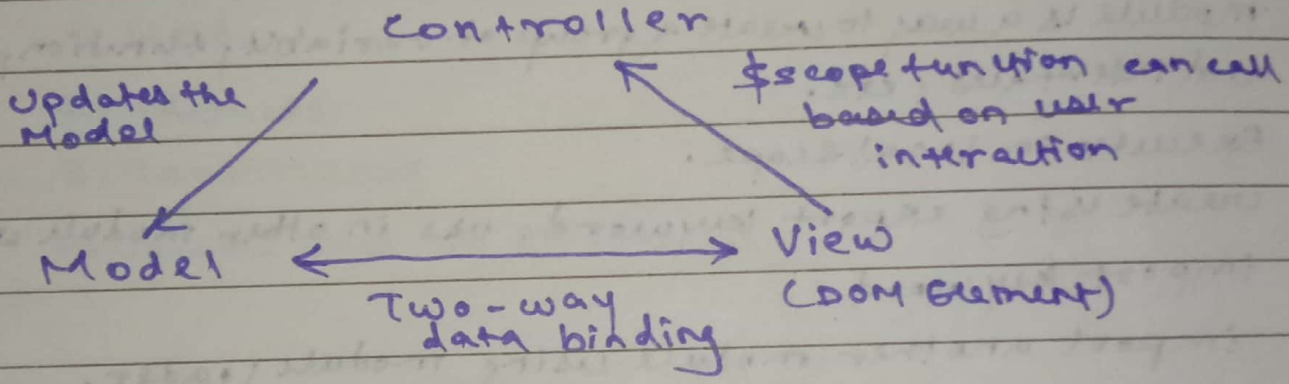
Angular JS

↳ Client Side Web Application Framework

- Separation of concerns

→ It is an MVC framework that is similar to JS framework.

→ Cross Browser compatible, can be easily added to HTML pages with simply a <script> tag.



- View is independent of Model
- Reusable pattern
- Decouple the modules
- Parallel Development
- Multiple view support.

Data Binding :-

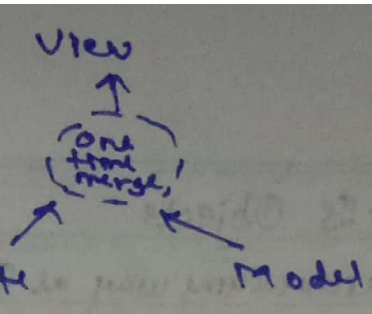
→ refers to the sync of data b/w model and view.

→ Data Binding in AngularJS is achieved by using Directives.

Provides two type of Data Binding :-

- One way Data Binding
- Two way Data BINding

One-way Data Binding

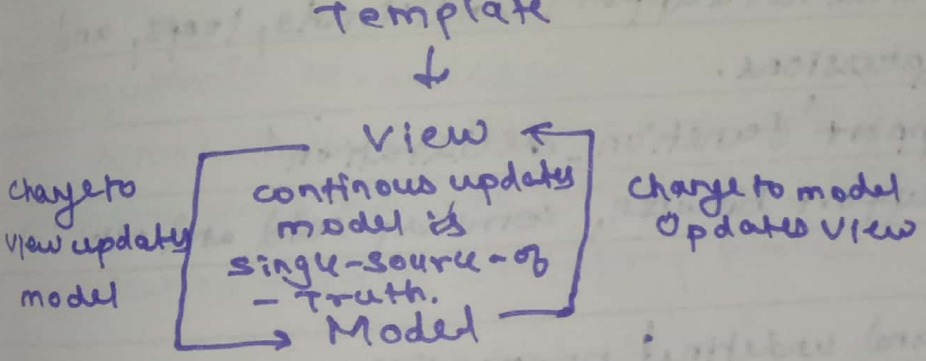


→ It can be achieved by :-
- Interpolation
- using ng-bind directive

* Interpolation :-
AngJS expression can contain literals, operators or variables, unlike traditional JS.

```
<div ng-app="">
<p> My expression in AngJs : {{ 3+3 }} </p>
</div>
```

Two way data Binding



Datatype :-

- Numbers
- Strings

```
<div ng-app="" ng-init="first_variable_name='your_string';
second_variable = 'your_string' ">
<p> My first exp in AngJs : {{ first_variable + second_variable }}
</p>
</div>
```

Page No.
Date

- Angular JS 'Objects'
  → behave same way as JS objects, can be accessed using dot operator.

Arrays:-
same as JS Arrays.

```
<div ng-app="" ng-init="your_array=[0,1,2,3]">
<p> My first value : {{ your_array[0] }} </p>
</div>
```

**** Note:-
- Angular JS does not support conditionals, loops, and exceptions in expressions.
- Does not support 'function_declaration'
- Does not support bitwise, comma, void and new operator
- Ignores null and undefined properties.
- Expression are evaluated belonging to scope object and not the global window.

Directives:-
- Tells the DOM what action (controller) need to be done. ; Directives are used like Attributes.
- 2 types:-
  Built-In
  Custom

Built-in Directives:-

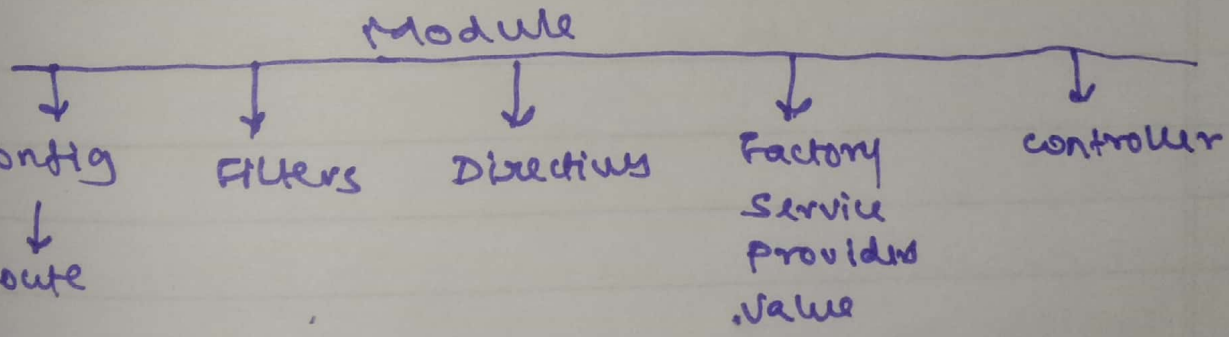
- ng-app :- Defines and links and AngularJS to HTML.
- ng-model :- binds the values of AngJS to HTML input
- ng-bind :- binds data to HTML tags.
- ng-controller :- to associate with controller
- ng-show :- show or hide control from DOM
- ng-init :- initialise any variable
- ng-repeat :- for looping
- ng-click :- is like onclick().
- ng-required :- for form validation.
- ng-disabled :- enable or disable control at runtime.

Custom Directives:-

To implement our own directives.

AngularJS Modules:-

A module is a collection of providers, services, directives, and optionally config file and run blocks which get applied to application during bootstrap process.



Adding a controller.

```
app.controller("Controller_name", function($scope){
        $scope.variable_name = "";
});
```

Page No.
Date

AngularJS Scope:-

→ Scope is an object that ~~refers~~ to application model.
→ used to bind both ~~controller~~ view and model.
→ accessible to both.

Angular Controllers:-

control flow of data.

AngularJs Filters:-

→ functions that are used to format, transform and filter data in expression.
→ They allow you to modify data before it is displayed

Intro to Web X.0

Web 1.0

www

Web 2.0

collab and share

Web 3.0

Decentralized web, semantic

web readable by machine not only by humans

Web 4.0

Connect device to humans

ML, VR, IOT, AI.

Symbiotic web

Web 5.0

reclaim data and ownership

Web Analytics.

→ Web analytics ~~helps~~ allow an organization to collect, measure and analyze the data from the visitors of organization's site

→ Used for marketing research.

→ where the traffic is coming from.

- Understanding user behaviour
- Identifying opportunities for improvement
- Measuring effectiveness of Marketing efforts
- Setting and Tracking goals.

Clickstream Analysis:-
building blocks of metrics and (KPIs).

The critical Eight:-
- visits
- unique visitors
- Time on page
- Time on site
- Bounce Rate
- Exit Rate
- Conversion Rate
- Engagement

VUTT
BECE

Web Analytics 2.0

→ analysis of quantitative and qualitative
→ to drive continual improvement, which translates into desired outcomes.

The What: clickstream
How much:- Multiple Outcome Analysis
The Why:- Experimentation and Testing
The Who:- Voice of customer
The what-Else:- competitive Intelligence

Strategic Shift

Web Analytics 1.0

Clicks Rule!

Head

Analysis Scope
Me, Me, Me

Data → Report →
You → Boss →

Discrete

HIPPO's Rule

Web Analytics 2.0

clicks Rule! Not

Head and Heart

You and competitors

Automated Decision making

continuous

Customer Rule

Page No.
Date

How to Choose Web Analytics Tool:-
Collect Business requirements
Collect Tech. Requirements
Documentation
Prepare Request for Proposal (RFP)
Send RFP to Vendor
Analyse Reply
Selects one Vendor
Implement

KPI for business:-
Task completion Rate:-
Share of search
Visitor Loyalty and Recency
RSS / Feed subscriber
% of valuable Exits.

For ecommerce site:-
Cart Abandonment
checkout Abandonment
Days and visits to purchase
Avg. order Value,

For Non-ecommerce
Visitor Loyalty, Vis. Recency,
Length visit, Depth of visit

Different types of Analysis :-

- Descriptive → what happened in the past
- Diagnostic → why did it happened in the past
- Predictive → what will happen in future
- Prescriptive → How can we make it happen.

Semantic Web

a goal that it would be possible for machine to understand data on the web rather than simply display it.

To unlock potential of web
Link data and relation with data
web of data rather than web of documents.

"An extension of current web in which information is given well-defined meaning better enabling computers and people to work together".

Main topics that provide conceptual underpinning

- Building models
- computing with knowledge
- Exchanging Info.

Linked Data :-

Define rel"ship with data.
accessibility and integration
standard Format :- Linked Data Triples.

bestbuy.com, Apple Inc., DBPedia, UN environmental indicators.

RDF → Resource Description Framework.

Page No.
Date

Semantic Web stack.