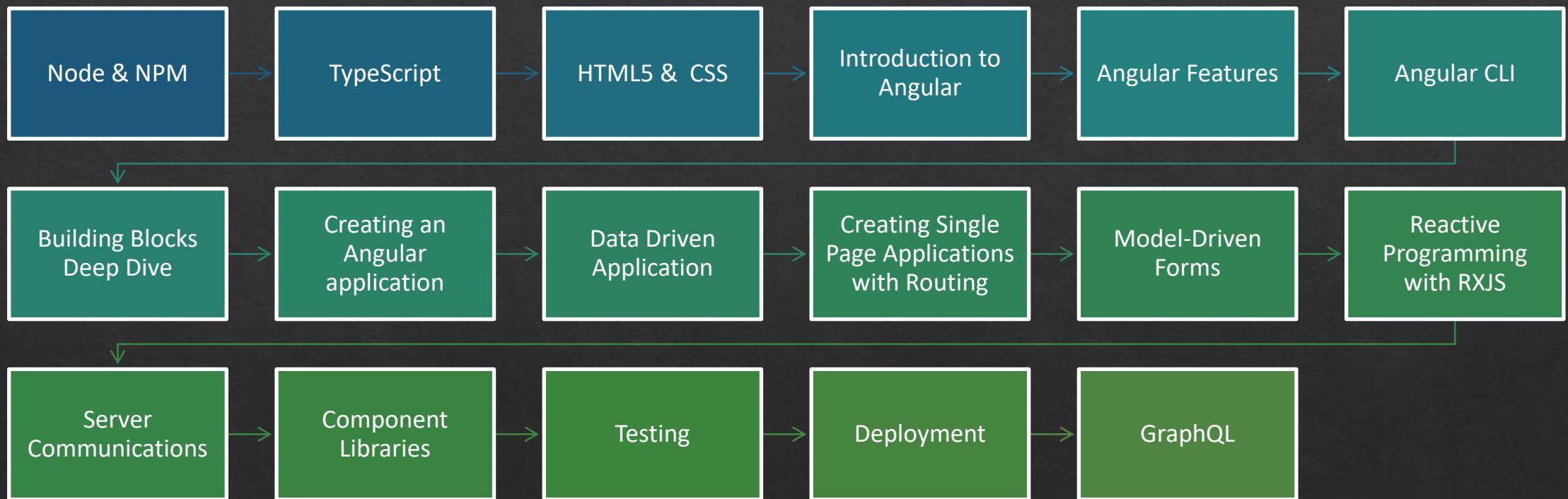
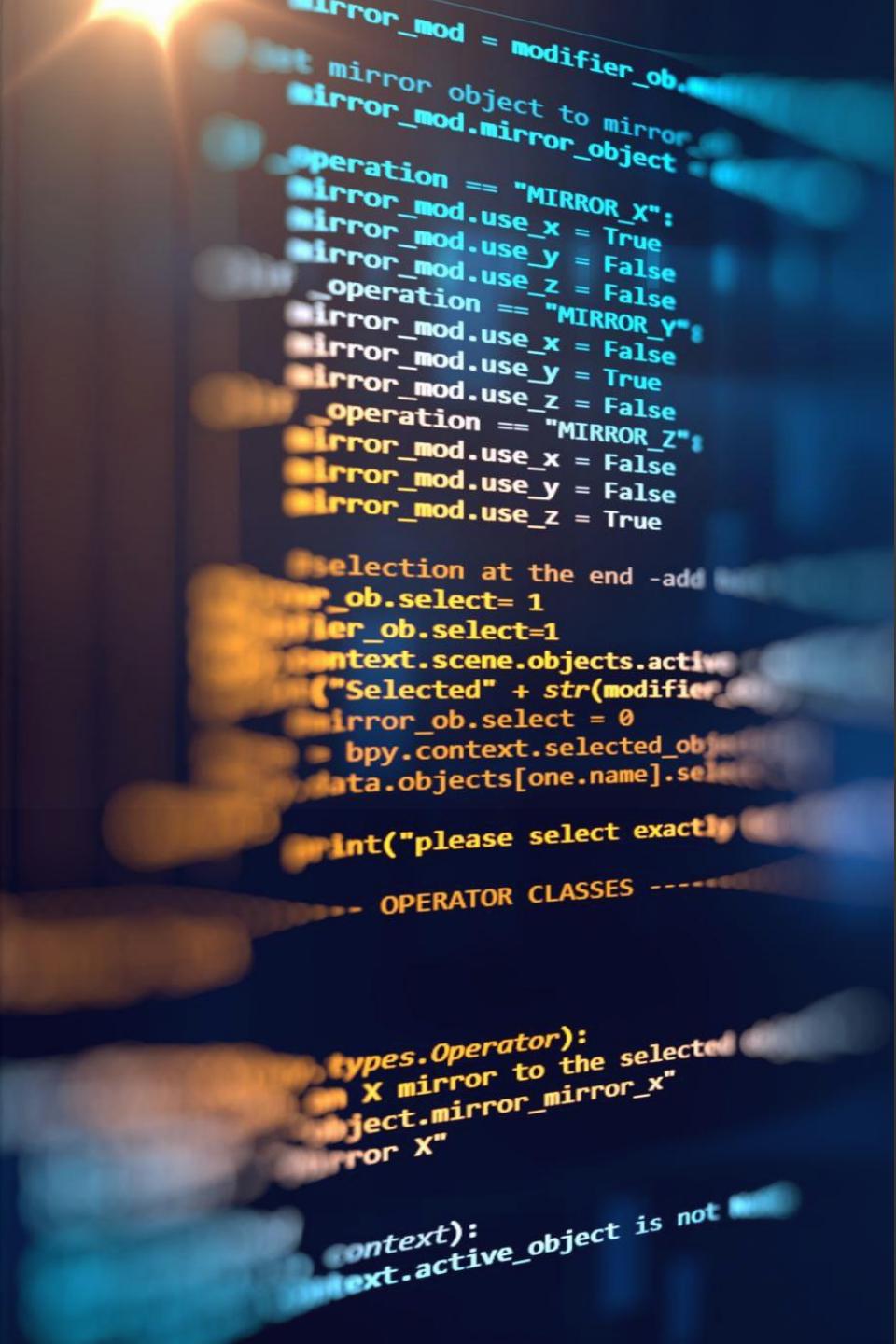




ANIL JOSEPH

Agenda





```
mirror_mod = modifier_obj
set mirror object to mirror
mirror_mod.mirror_object = modifier_obj
operation == "MIRROR_X":
    mirror_mod.use_x = True
    mirror_mod.use_y = False
    mirror_mod.use_z = False
operation == "MIRROR_Y":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

selection at the end -add
modifier_obj.select= 1
modifier_obj.select=1
context.scene.objects.active = modifier_obj
("Selected" + str(modifier_obj))
modifier_obj.select = 0
bpy.context.selected_objects.append(data.objects[one.name].select)
print("please select exactly one object")
- OPERATOR CLASSES -
types.Operator:
    X mirror to the selected object.mirror_mirror_x"
    or X"
context):
    context.active_object is not None
```

Introduction

Anil Joseph

- ❖ Over 20 years of experience in Training and Development
- ❖ Technologies
 - ❖ C++
 - ❖ Java
 - ❖ .NET and .NET Core
 - ❖ Node, Node Express and other frameworks
 - ❖ UI Technologies: React, Angular, ExtJS, jQuery, Knockout, Redux etc
 - ❖ Mobile: Native Android and React Native
- ❖ Worked on numerous projects
- ❖ Conducted training for corporates(700+)

Software



Node.js



Angular CLI



Visual Studio Code



Java Development Kit



Spring Tool Suite

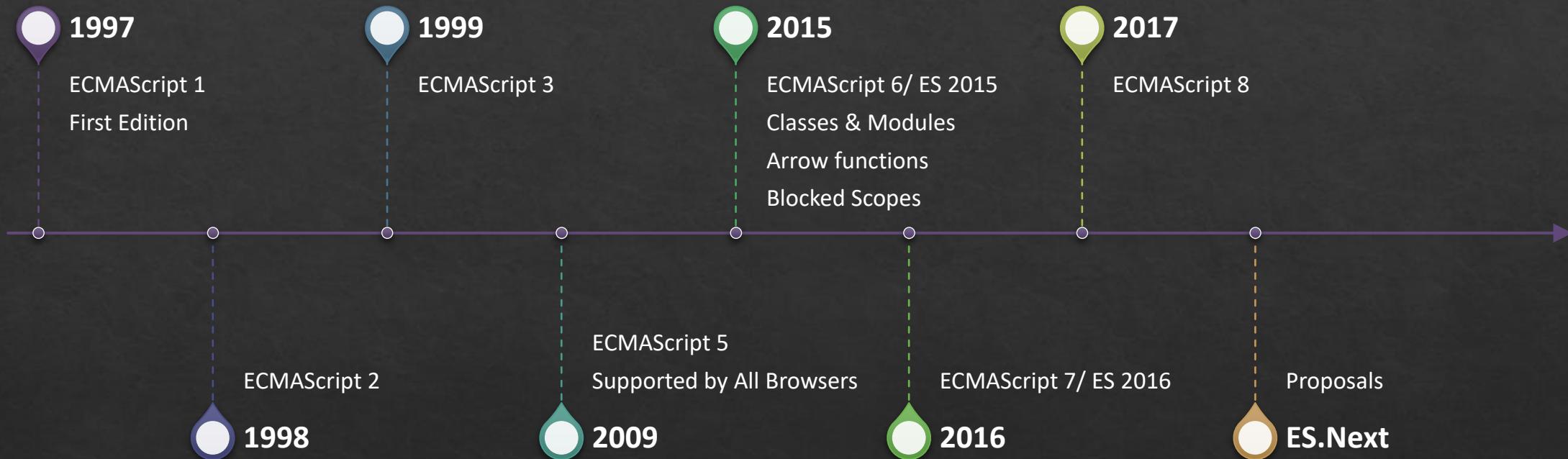


Browsers(Chrome)

JavaScript

- ❖ JavaScript (JS) is an interpreted programming language.
- ❖ It's a dynamic language.
- ❖ Supports object-oriented programming.
- ❖ As part of web browsers, implementations allow
 - ❖ Client-side scripts to interact with the user.
 - ❖ Alter the document content that is displayed.
 - ❖ Control the browser.
 - ❖ Communicate asynchronously.
- ❖ JavaScript was developed by Brendan Eich at Netscape.
- ❖ Released in September 1995

ECMAScript Versions





Node.js



A server-side JavaScript platform



Created by Ryan Dahl in 2009



Provides an easy way to build scalable network applications



Built on top of Google V8 JavaScript Engine and libuv

Node Features

Cross-Platform

Asynchronous I/O Framework

- Uses non-blocking, event-driven IO

Build fast, scalable network applications

- Capable of handling huge number of simultaneous connections with high throughput

Node Features

Node API

- A built-in library for common functionalities

Highly Extensible

- Customize and extend Node.js as per requirements.

A community driven platform

Full stack development

- Build a complete solution using JavaScript

Getting Started

- ❖ Install Node
 - ❖ <https://nodejs.org/en/>
- ❖ This installs Node and NPM
- ❖ Verify by opening a command prompt and executing the command
 - ❖ node –version
- ❖ To execute a JavaScript program
 - ❖ node hello.js

Node REPL(Read Eval Print Loop)

- ❖ Node provides a standalone and includible REPL.

global Object

Every node application has a global object



The global object exposes objects like console, process and timer methods

Modules

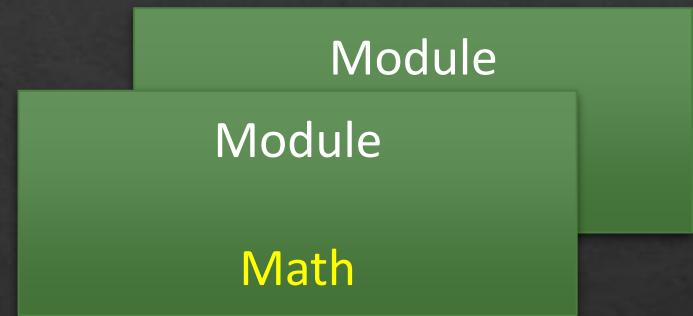
Node Applications are modular

In the Node.js module system, each file is treated as a separate module.

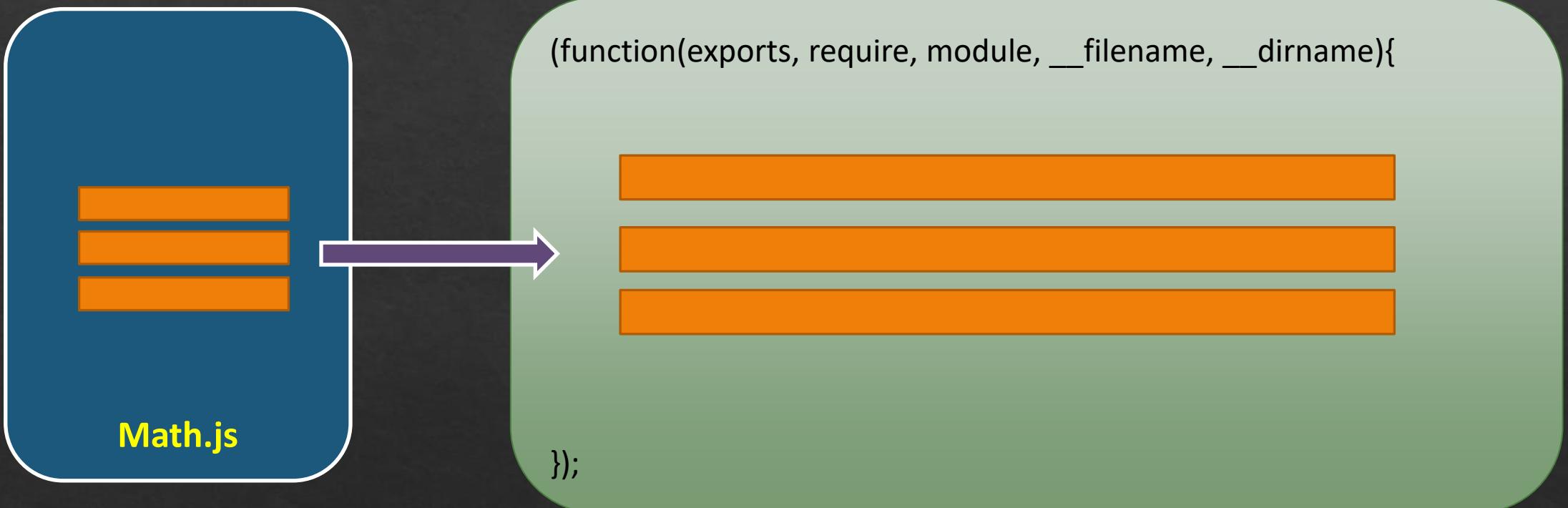
Every application will have one main module.

The require function is used to include/require modules

Module



Module Wrapper



Sources of Modules

Built-in modules

- Comes pre-packaged with Node
- Examples fs, http, crypto, os

The project files

- Each .js file is a module
- JavaScript files can export variables, functions and objects.
- JavaScript files can be imported by providing the path.

Module folder

- A folder with a package.json file containing a main field.
- A folder with an index.js file in it.

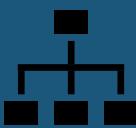
External Modules

- Install using NPM or yarn
- Installed into node-modules folder

require



Node.js follows the CommonJS module system,



The built-in **require** function is used to include modules in separate files.



The basic functionality of require is

Reads a JavaScript file
Executes the file
Return the exports object.

NPM(Node Package Manager)

- ❖ NPM is a package manager for the JavaScript programming language.
- ❖ Allows users to consume and distribute JavaScript modules that are available on the registry
- ❖ The default package manager for Node.
- ❖ Comprises of
 - ❖ Command line client, also called npm
 - ❖ An online database called the npm registry.
 - ❖ Public
 - ❖ Paid-for private packages
 - ❖ NPM website
- ❖ The package manager and the registry are managed by npm, Inc.

NPM Packages

- ❖ Packages are reusable code published in the npm registry.
- ❖ A directory with one or more files a metadata file called package.json.
- ❖ A package is a building block that solves a specific problem.
- ❖ An application generally depends on many packages.
- ❖ Packages can be used on
 - ❖ Server side. Example: Express, Request
 - ❖ Client Side. Example Angular, React
 - ❖ Command based. Typescript, Angular CLI, JSLint

Modules revisited

- ❖ A module is anything that can be loaded with require() in a Node.js program.
- ❖ Examples
 - ❖ A folder with a package.json file containing a main field.
 - ❖ A folder with an index.js file in it.
 - ❖ A JavaScript file.
- ❖ Most npm packages are modules

NPM Commands

- ❖ NPM is installed with Node
- ❖ Check the version
 - ❖ `npm -v`
- ❖ Update npm
 - ❖ `npm install npm@latest -g`
- ❖ Find the path to npm's directory:
 - ❖ `npm config get prefix`
- ❖ Help
 - ❖ `npm -h`
 - ❖ `npm [command] -h`
 - ❖ `npm help [command]`
 - ❖ `npm help-search [key]`

package.json

- ❖ A metadata file for the project
- ❖ Used to track dependencies
- ❖ Create Scripts
- ❖ Command to create package.json
 - ❖ npm init
- ❖ Setting defaults
 - ❖ npm set init-author-name 'Anil Joseph'
 - ❖ npm get init-author-name
 - ❖ npm config delete init-author-name

Installing Packages

- ❖ Install to a project
 - ❖ npm install express
- ❖ Install and save to dependencies
 - ❖ npm install angular --save
- ❖ Install and save to development dependencies
 - ❖ npm install express --save-dev
- ❖ Install globally
 - ❖ npm install gulp -g

Install Packages with version

- ❖ Specific version
 - ❖ `npm install underscore@1.8.2`
- ❖ Latest Version
 - ❖ `npm install underscore@1.8.x`
 - ❖ `npm install underscore@1.8`
 - ❖ `npm install underscore@1.8.2`
 - ❖ `npm install underscore@1.x`
 - ❖ `npm install underscore@1`
 - ❖ `npm install underscore`
- ❖ Other Options
 - ❖ `npm install underscore@">=1.4.x < 1.6.x"`

Listing & Removing Package

- ❖ npm list
 - ❖ List all packages
 - ❖ Shows all dependencies of packages
- ❖ npm list --depth 1
 - ❖ List packages show dependency depth to 1 level
- ❖ npm list --global true
 - ❖ List global packages
- ❖ npm list --global true --depth 0
 - ❖ List global packages show dependency depth to 0 level

Listing & Removing Package

- ❖ `npm list --dev true`
 - ❖ List development dependiecies
- ❖ `npm list --prod true`
 - ❖ List Production dependencies
- ❖ `npm ls`
 - ❖ ls as shortcut
- ❖ `npm uninstall underscore/ npm rm underscore/ npm un underscore`
 - ❖ Removes/Uninstall a package
- ❖ `npm uninstall underscore --save`
 - ❖ Uninstalls and updates the package.json
- ❖ `npm uninstall underscore -g`
 - ❖ Unstall global package

TypeScript

TypeScript

TypeScript

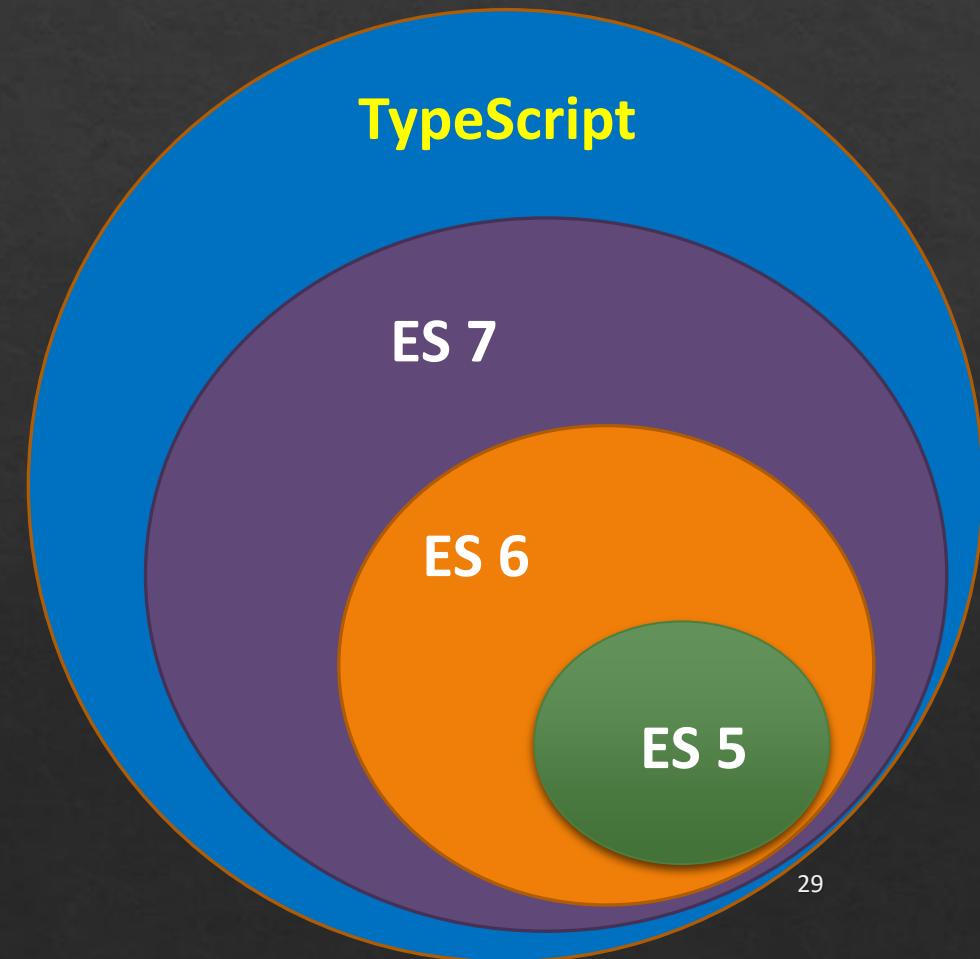
TypeScript is programming language developed and maintained by Microsoft.

TypeScript is a typed superset of JavaScript.

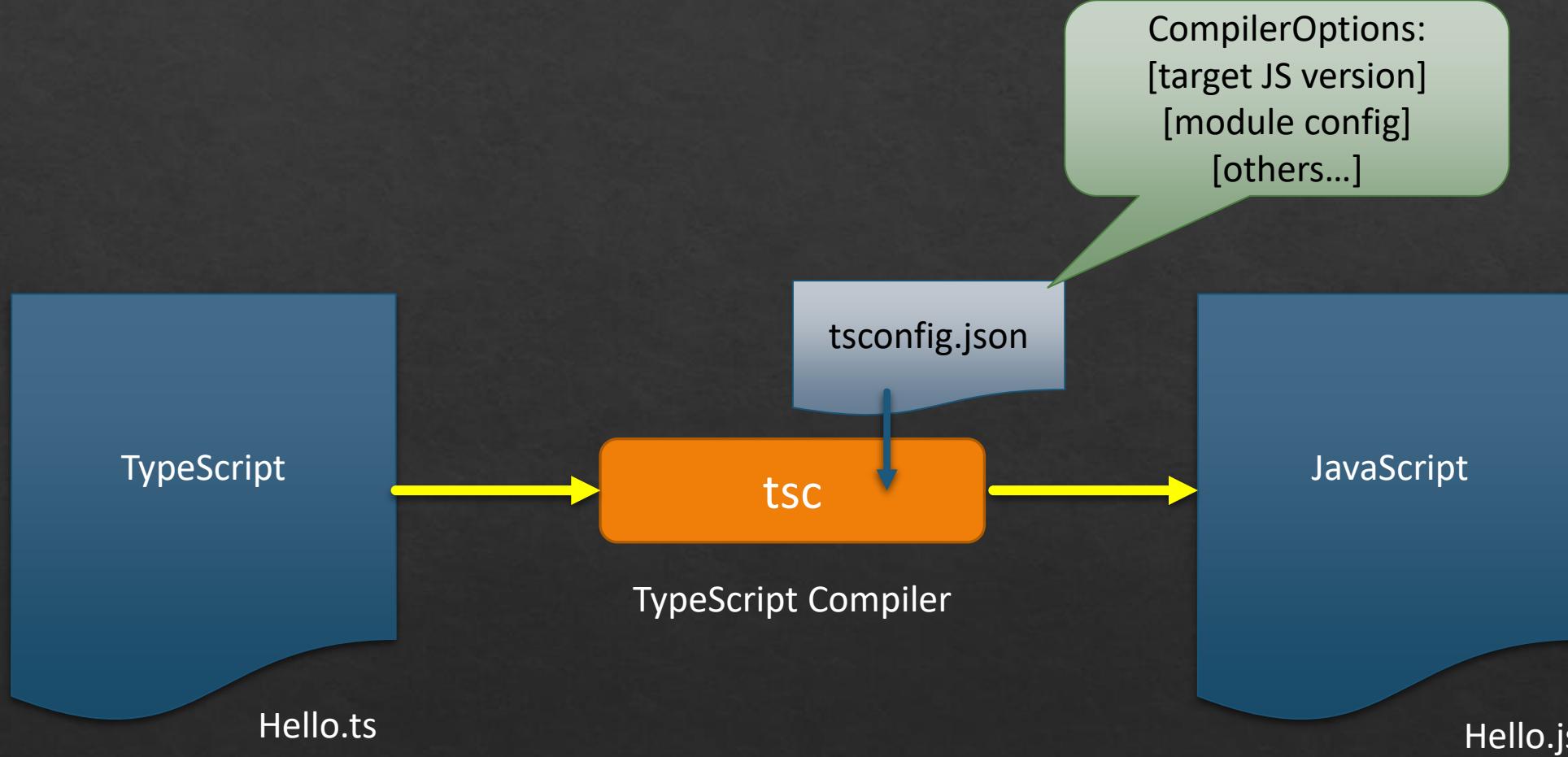
Transcompiles to JavaScript.

Designed for development of large applications.

Open Source.



TypeScript



TypeScript Features

Type Annotations

Compi-
le-
Time
Type
Check-
ing

Type Inference

Interfaces

Classes &
Inheritance

Namespaces and
Modules

Generics

Decorators

Arrow Functions

TypeScript Installation

Node.js

- `npm install -g typescript`

Available with

- Visual Studio 2015 & 2017
- Visual Studio Code

Plugins available with

- Eclipse
- Atom
- Sublime
- WebStorm
- Many more

TypeScript Types

Boolean

- **let** isAvailable: boolean = false;

Number

- **let** age: number = 16;
- **let** hex: number = 0xf00d;

String

- **let** name: string = "Anil";

Array

- **let** list: number[] = [1, 2, 3];
- **let** list: Array<number> = [1, 2, 3];

TypeScript Types

Enum

- **enum** Color {Red, Green, Blue}
- **let** c: Color = Color.Green;

Any

- **let** x: any = 4;
- x = "hello"

Tuple

- let data: [number, string];
- data = [1, "Anil"];

TypeScript Types

void

```
function foo(): void {  
    console.log("foo");  
}
```

Null and Undefined

- var x: string = null;
- var y:string= undefined

Interfaces

- ❖ Interfaces are a powerful way of defining contracts.
- ❖ Example

```
interface Vehicle{  
  
    name: string;  
    speed: number;  
    gear?: number;  
  
    applyBrakes(decrement: number): void;  
}
```

- ❖ Interfaces can extend interfaces
 - ❖ (keyword extends)
- ❖ Classes implements interfaces
 - ❖ (keyword implements)

Classes

- ❖ Traditional JavaScript uses functions and prototype-based inheritance to build up reusable components.
- ❖ Starting with ECMAScript 2015, also known as ECMAScript 6, JavaScript introduces the object-oriented class-based approach.
- ❖ TypeScript supports classes that compile down to JavaScript
 - ❖ Works across all major browsers and platforms
 - ❖ Without having to wait for the next version of JavaScript.

Classes

- ❖ Example

```
class Car implements Vehicle{  
    constructor(public name: string, public speed: number, public  
    gear: number){  
  
    }  
    applyBrakes(decrement: number): void{  
  
    }  
}
```

- ❖ Modifiers supported

- ❖ public, private, protected
- ❖ Can have constructors
- ❖ Supports Properties
- ❖ Supports static members
- ❖ Supports inheritance
- ❖ Classes and methods can be abstract

Arrow Functions

Represents a function expression

An arrow function expression has a shorter syntax than a function expression

They do not receive the implicit arguments “this” and “arguments”

Used widely for asynchronous and functional programming

TypeScript: Modules

- ❖ Starting with the ECMAScript 2015, JavaScript has a concept of modules.
- ❖ TypeScript shares this concept.
- ❖ Modules are executed within their own scope
 - ❖ Not in the global scope

Modules

- ❖ Use the import and export statements.

```
let foo = function(){  
    //some code  
}
```

```
export default foo;
```

one.js

```
import foo from './one';  
  
foo();
```

two.js

```
import bar from './one';  
  
bar();
```

three.js

Modules

```
export let foo = function(){  
    //some code  
}  
  
export let bar = function(){  
    //some code  
}
```

one.js

```
import {foo, bar} from './one';  
  
foo();  
bar();
```

two.js

NPM(Node Package Manager)

- ❖ NPM is a package manager for the JavaScript programming language.
- ❖ Allows users to consume and distribute JavaScript modules that are available on the registry
- ❖ The default package manager for Node.
- ❖ Comprises of
 - ❖ Command line client, also called npm
 - ❖ An online database called the npm registry.
 - ❖ Public
 - ❖ Paid-for private packages
 - ❖ NPM website
- ❖ The package manager and the registry are managed by npm, Inc.

NPM Packages

- ❖ Packages are reusable code published in the npm registry.
- ❖ A directory with one or more files a metadata file called package.json.
- ❖ A package is a building block that solves a specific problem.
- ❖ An application generally depends on many packages.
- ❖ Packages can be used on
 - ❖ Server side. Example: Express, Request
 - ❖ Client Side. Example Angular, React
 - ❖ Command based. Typescript, Angular CLI, JSLint

