Swing application would depend on the underlying OS for:  
1) data storage capability

2) file access

3) multithreading

4) intraprocess communication/ sockets

5) application host: I should be able to deploy the application and run it

6) device access : printers etc

In web: todays we have

1. Data storage capability: local storage, session storage, index db
2. File access: APS -> file system API
3. Multithreading: web workers
4. Sockets/interprocess communication: web sockets , web RTC , web events
5. Application host: offiline application cache, service worker
6. Device access: APIs available by which we can access webcam, speakers etc

>> gmail offline : to access gmail even when you are offline.

>> cloud9 comapany : they have an IDE – cloud9 IDE.   
it is an online ide. We code on the browser and save your code on gitHub. You need to have a gitHub account to work with cloud9

>> appHarbour

Graph db, column db, relational db

Rich internet applications : flash applications, applets, activeX etc : even though they are web applications, they give features of desktop applications. And the richness is in the user experience. They were plugin architecture based applications.

To move away from plugins, we have to use HTML, CSS and JS. Such applications are called SPA (Single Page Applications)

Angular2:

Applications are composition of components

Web Components: - read

OLEdb and COM , DCOM,Microsoft transaction services (MTS), COM+, EJB, RMI (Remote method invocation)

CORBA, JNI, Web Services

Web Services: HTTP,XML,SOAP,WSDL,DISCO

HTML import : whichever HTML you import must have HTML (presentation) + CSS + Behaviour

My browser should have support for custom elements, so in my html I can use something like:  
<map></map>  
<weather></weather>

Chrome created a polyfill that allows you to create web components even when the browser is not allwoing you to create web components. That polyfill was called polymer. They can be used in any application in any framework.

Angular **component** = controller (processes data/Behaviour) + view (Accepts data) : user interaction logic goes in component

**Pipe**: used for data transformation.

**Directive**: low level DOM manupilation.

**Component,pipe and directive used for UI**

**Service** : non user interaction logic goes in service

**Services used for non UI**

**Modules in angular is analogous to packages in java.**

Angular is availale in 3 flavours:

Angular in JS

Angular in TS

Angular in DART (developed by google with the intension of replacing js in the browser)

Languages like coffeeScript, typeScript,GWT (google web toolkit) are high level languages where you code in that language and that converst it into js.

Typescript is a superset of JS. It is not a new language, its just new features added onto JS.   
typescript=js+type safety.

What is type safety? 1) type safety is taken care of by the compiler in strongly types languages.

Jquery:

$ in jquery is just a function. JQuery is a function. And the $ is also just the same function as jQuery. Just given for ease of use.

$(document).ready(fn); // can pass a single dom node to $ function

$([domNode1,domNode2]).ready(); // can pass array of dom nodes to $ function

You can also pass strings, a function and a $ itself to the $ function.

Refactoring by Martin fowler // buy this book

Java has maven , similarly there is npm for node and javascript.  
there is a central repository where all the packages are present : npmjs.com

A metadata file called **package.json** file contains the complete metadata information about your application. The application repository, the license, etc  
A list of dependencies is also maintained in package.json. dependencies are of two types:  
1) **dependencies**: required for the application to run. At run time

2)  **dev Dependencies**: dependencies you need at development time  
package.json also has the **script** section. Where you define the commands and execute the application.

Tsconfig.js

Configuration file for the typescript compiler  
e.g: one configuration is "target": "es5"

Bable JS is a transpiler. Used to transpile ES6 to ES5. Many browsers don’t support ES6 natively. Hence we have to transpile ES6 to ES5We code in ES6, so coding becomes better, but before putting code in production, we run the transpiler to convert ES6 toES5.   
our typescript compiler also performs this transpilation

"sourceMap": true,

Source map is used for debugging

"moduleDefinition": "commonJS",

AMD (asynchronous module definition) 🡪 a module specification. The library the supported AMD module specification was called requireJS

In NodeJS, they started using ANOTHER WAY OF CREATING MODULES and that specification was called commonJS specification.

“Browserify” : it converted commonJS modules into a form that browser could identify.

Later AMD was decommissioned. And commonJS was used for both browser and server code

"moduleResolution": "node",

It says that resolve imports using node\_module folder.

Tslint.json file is used for static code analysis. Checking for spaces, syntax etc. this is called ‘linting’ to ensure code is written in a proper way. Tslint.json is the linter configuration for typescript

Karma.config.js : Tool for executing your tests. Karma is a test runner.

We can configure karma to execute our tests across browsers.

Protractor.conf.js : its a framework used for performing end to end testing. User acceptance testing. Protractor library is a wrapper library build on top of selenium webdriver

**Starting point of an angular application is main.ts**

Component composition : being able to pass data to the component from outside

Bug Tracker application:

1. Add a new bug
2. List the bugs
3. Toggle the ‘closed’ status of a bug
4. Remove closed bugs
5. Display statistics (# of closed bugs/total # of bugs)
6. Improvise bug display
7. Sort the bugs
8. Persist the bugs in the browser’s local storage
9. Display ‘createdAt’ for all the bugs
10. Persist the bugs in the server
11. Make the bug accessible through unique URLs

Zen coding

Structural directive (e.g.ngFor): a directive that can make changes to the DOM hierarchy. They can add/remove DOM nodes.

[**http://bit.ly/javascript-training-videos**](http://bit.ly/javascript-training-videos)

[**http://bit.ly/angular-training-videos**](http://bit.ly/angular-training-videos)

**angular/io.docs**

**common module is already there in the root module.**

**ngClass directive** is used to control class names in the module

[ngClass]="{closed: currentbug.isClosed}"

ngClass is the directive. Closed is the class name. isClosed is the property name

each component has a zone. And the zone tracks all the interpolations

“change detection by push”

Event driven application: there are 3 sources where your application state can change:  
1) any interaction performed by user (any DOM events)  
2) timer (e,g: setInterval 🡪 executes a given function for a given period of time )   
3) Ajax request (request something from the server and data comes at a later state in time)

On the developer options:

In the browser : ==$0 : browser assigns that particular dom node to a variable. That variable is $0

Console.dir{setInterval}

No processessing should happen in component. Any logic that is non ui should not be there in the component.

bugOperations:BugOperations = new BugOperations();

instead , use : a constructor (because of DI violation in the above case. We don’t want to use new every time)

to make a class a pipe, we use @ngPipe annotation. And we have to use an interface pipeTransform for it. A class makes a pipe in angular2. It becomes a pipe when we use the decorator @pipe

in the transform method 🡪 whatever is the first argument is what comes to the left of the pipe. And if there is another argument , then that has to be mentioned after a “:”

e.g. ‘first argument’ | sort:’second argument: third argument’

by default my pipes are considered ‘pure’ . it is adviced to write functions as ‘pure functions’. If a function takes an argument x, returning y, then this should be true 100% of the times. E.g:  
function add(x,y){return x+y;} is a pure function. Whereas function add(x,y){return x+y+z;} is not a pure function. Because it depends on something that is not part of its arguments and is not defined inside the function as well (z).

in pipes, they behave as pure functions. That is, if the input hasn’t changed (input to the pipe), it wont reevaluate.

If you want to efficiently handle pipe function, then everytime a bug is added you need to create a new array.

Local storage: to maintain data on the client side.  
1) it is a key value store

2) key has to be unique

3) both key and value must be of type string. Use JSON.stringify and JSON.parse for serialization and deserialization

4) follows Same Origin Policy

window.localStorage is the object.  
The API is :  
1) setItem(key,value)  
2) getItem(key) returns value  
3) removeItem(key)  
4) key(index) return key at index location  
5) length  
6) clear()

cross document messaging: send a message from one window to the child windows launched from that parent window.

Let is block scoped. Var is function scoped.

IIFE : Immendiately Invoked Function Expression. These functions are use and throw functions. Usually these functions are anonymous functions.

In the absence of module system , the browser behavior is public by default, private by choice. The module system took the opposite approach -> private by default. Public when you export

Minification and uglification

Paradox of choice by Barry Schwartz

Behavioural economics

Stumbling on happiness

<http://momentjs.com/>

how to handle asynchronous operations in javascript:  
Ajax is one of the asynchronous operation.  
What is the nature of an asynchronous:   
  
the browser uses an event queue. A key press first goes to this queue, and then later picked up from that queue and processed.  
1)a dom update doesn’t immediately lead to a screen repaint.  
2) in javascript , a function is an atomic unit of execution, i.e. you cannot perform another task till a function execution completes  
so based on 1) and 2) , the browser says, I’ll repaint the screen after the dom has been updated.  
  
how does the browser know when to repaint the screen ? how does it remember to repaint it after a function is executed? 🡪 so the browser uses a queue for it. The queue is the scheduler for the main thread. The queue decides what the thread has to do next. The thread’s job is to keep looking at the queue

DOM events are asynchronous events.  
  
wait for the operation to complete: synchronous  
don’t wait for the operation to complete: Asynchronous

Sequencing asynchronous operations can be tricky : Why? 🡪  
  
handling async operations :Callbacks.  
(setTimeout)  
But Callbacks started becoming a problem . Why? 🡪 the one who makes the call must also process te result. But this approach doesn’t work for large applications. So you have to break the operation and the result processing in 2 different things.  
so we use Events.  
But with events we cannot get the result if we subscribe after the operation completes.  
So we use Promise 🡪 we can subscribe to the result, even after the operation completes. (when you create a promise instance, you pass a function as an argument. And any async operation that you wish to perform, you perform it inside that function). The promise API invokes your function and it invokes your function with passing 2 functions as an argument - one is the resolve function, other is the reject function.  
promise object exposes 2 methods : the ‘then’ and ‘catch’ method.

Generators , async awake and Observables are used by angular2 or async handling. Promises are used by angularjs

promisesA+.com

<https://www.youtube.com/playlist?list=PL2OMOVtxSKThVeJJQjDPj5uAzsgIowF6x>

an observable is a data structure/collection where the data will populate sometime in the future and you just subscribe to the observable to give the data to you when data arrives in it.

Rx 🡪 reactive extensions🡪 Eric Meijer videos

reactivex.io/

npm install json-server –g

json-server bugsDb.json.json

crossroads.js library 🡪 read about it

at the time of build, all the ngModel that you have in your html, must be mentioned/declared in your component. Otherwise it will throw a build error. Its not a compile time error, but is a build error.

[tkmagesh77@gmail.com](mailto:tkmagesh77@gmail.com)

99019-11221