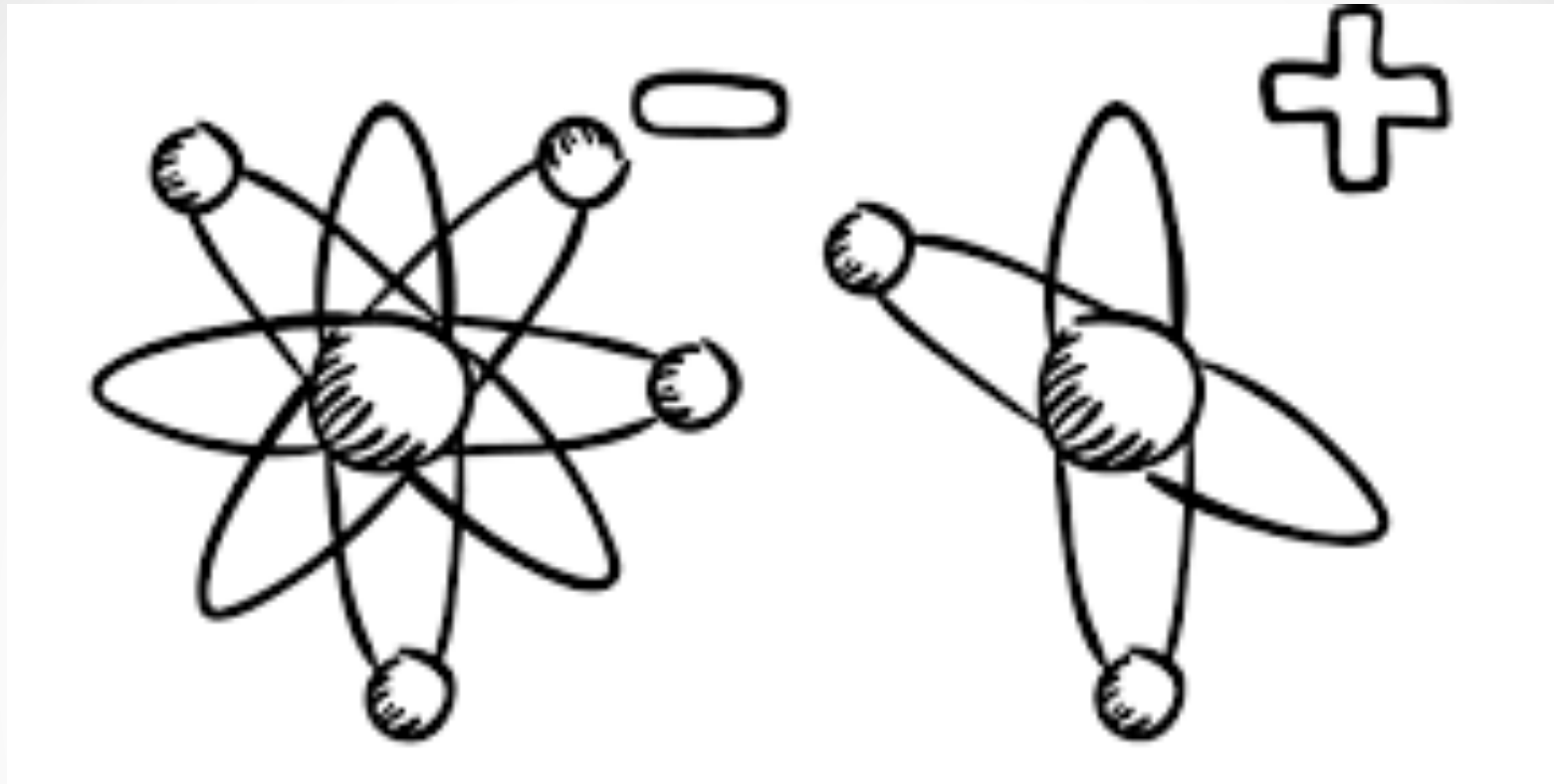




Ionic 2

Overview

- Open source SDK for hybrid mobile app dev.
- Based on
 - AngularJS
 - Apache Cordova
- Develop app using
 - CSS
 - HTML5
 - Sass
- Can distribute apps to native app stores
- Installation needs
 - Node.js
 - npm



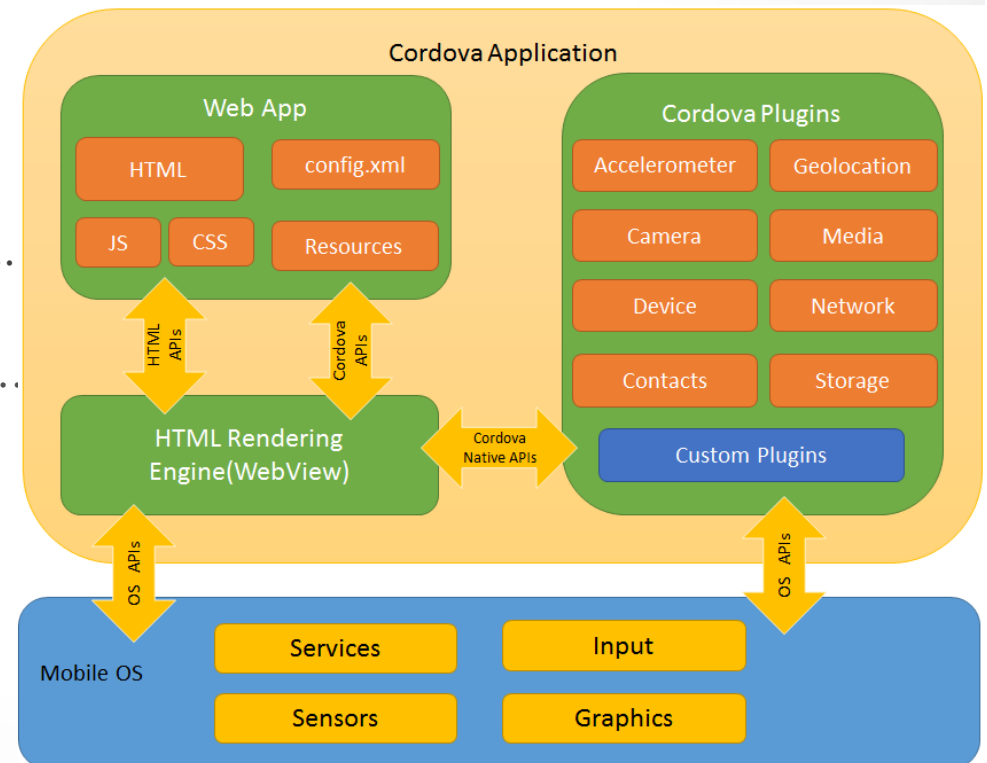
Apache Cordova

- formerly PhoneGap, Cordova Callback
- cross-platform mobile Apps using web technologies
 - HTML5: Rendering, HW access
 - CSS3: Styling
 - JavaScript: Logic
- Runs in a wrapper for target platform
- Can be wrapped
 - Ionic, Intel XDK, Monaca...
- Extendable by plugins
- Can mix with native code
- No UI widget/ MV* framework



Cordova Architecture

- **WebView**
 - Provides user interface by HTML rendering
- **Web App**
 - Your code, as a web page, index.html
- **Plugins**
 - let JS call native components
 - core plugins
 - acc, cam, comp, cont, loc,...
 - 3rd-party plugins
 - splashscreen, inAppBrowser,...



Cordova Resources

- Homepage
 - <http://cordova.apache.org/>
- Introduction
 - <https://cordova.apache.org/docs/en/latest/guide/overview/index.html>
- Supporting features
 - <https://cordova.apache.org/docs/en/latest/guide/support/index.html>
- Plugins
 - <https://cordova.apache.org/plugins/>
- Tutorial
 - <http://www.tutorialspoint.com/cordova/>

AngularJS

- Build a web app with *dynamic* views (1-page app)
 - JavaScript framework
 - Extends HTML attributes with **Directives**
 - Binds data to HTML with **Expression**
- By Misko Hevery in Google since 2009
- Called “Angular” because of “<” in HTML
- “ng” namespace because ng sounds like angular
- <http://angularjs.org>



AngularJS is JS library

- Written in JavaScript
- Add to webpage by
 - `<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>`

AngularJS Directives

- Using ng-Directives
- **ng-app** directive
 - Tells AngularJS is active on this part of HTML
 - `<div ng-app>`
- **ng-init** directive
 - initializes AngularJS application variables.
 - `<div ng-init="label='Your name is '">`
- **ng-model=*var_name*** directive
 - binds the value of HTML control (input, select, textarea) to the variable
 - `<input type="text" ng-model="name">`
- **ng-bind=*var_name*** directive
 - binds the variable to the HTML view
 - `<span ng-bind="name"></span>`
- In action
 - http://www.w3schools.com/angular/tryit.asp?filename=try_ng_intro
- Resource: Directives
 - http://www.w3schools.com/angular/angular_directives.asp

AngularJS Expressions

- **{{ *expression* }}**
 - expression is similar to JavaScript expressions
- Output data exactly as the expression is written
 - {{5+5}}
 - {{firstName + " " + lastName}}
 - {{person.age}} (Object: person={name="John", age=22})
 - {{points[2]}} (Array: points=[1,2,3,4])
 - In action
 - http://www.w3schools.com/angular/tryit.asp?filename=try_ng_expressions_strings
- Binds as same as **ng-bind** directive
 - {{name}}
 -
- Change CSS using expression
 - <input style="background-color:{{col}}" ng-model="col" value="{{col}}">
 - In action:
 - http://www.w3schools.com/angular/tryit.asp?filename=try_ng_expression_3
- Resource: expression
 - http://www.w3schools.com/angular/angular_expressions.asp

AngularJS Resources

- Homepage
 - <http://angularjs.org/>
- Reference
 - http://www.w3schools.com/angular/angular_ref_directives.asp
- Tutorials
 - http://www.w3schools.com/angular/angular_examples.asp
- Angular 2 cheatsheet
 - <https://angular.io/cheatsheet>

TypeScript

- Superset of JavaScript
- Free, open source prog. language by MicroSoft
- Compiled to JavaScript
- Benefit
 - Type-checking (static typed)
 - Can define new classes (more object oriented)
 - Function overloading

TypeScript: Resources

- Homepage
 - <https://www.typescriptlang.org>
- Tutorial
 - <http://www.typescriptlang.org/docs/tutorial.html>
- Play (TS2JS)
 - <http://www.typescriptlang.org/play/>
- Language Specifications
 - <https://github.com/Microsoft/TypeScript/blob/master/doc/spec.md>
- Reference
 - <https://www.typescriptlang.org/docs/handbook/basic-types.html>

Ionic 2

- Overview
 - Rewrite of the Ionic
 - Simpler
 - Easier to build
 - Improved performance
 - More native feeling UX
- Now in beta
- Depends on
 - TypeScript
 - Angular 2 (beta)
 - Sass
 - Cordova
 - Gulp



Ionic2: Environment

- OS
 - Linux, Windows, OS X
 - For iPhone deployment, use OSX or PhoneGap Build / Telerik AppBuilder
- Requirements
 - Xcode (Mac)
 - Android Studio
 - Genymotion
 - Editor: VS Code (<https://code.visualstudio.com>), Atom
 - NPM (Node Package Manager) within Node.js
 - can install Cordova / Ionic via NPM
 - `npm install -g cordova`
 - `npm install -g ionic@beta`
- Installation guide
 - <http://ionicframework.com/docs/guide/installation.html>

TypeScript: Types

- Static type
 - `var <id>:<type> = <value>`
 - `var name:string = "hello";`
- Dynamic type
 - `var <id>:any = <value>`
 - `var x:any = 4;`
 - `x = "hi";`
 - `x = new Object();`
- Automatic type inferring
 - `var <id> = <value>`
 - `var demo = 1;`
 - `demo = "abc";` → Error
- When compiled, types are erased

```
var index: number;  
index = "this is a string, not a number";
```

TypeScript: Class

- Class

```
class User {  
    name: string;  
  
    constructor(name: string) {  
        this.name = name  
    }  
  
    sayHello() {  
        console.log('Hello, I am', this.name);  
    }  
}  
  
var user = new User('Ashteya');  
user.sayHello();
```

```
var User = (function () {  
    function User(name) {  
        this.name = name;  
    }  
    User.prototype.sayHello = function () {  
        console.log('Hello, I am', this.name);  
    };  
    return User;  
})();  
  
var user = new User('Ashteya');  
user.sayHello();
```

```
class Person {  
    constructor( public name: String ) {  
        this.name = name;  
    }  
}
```


TypeScript: Interface

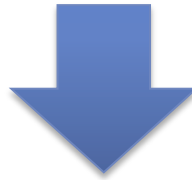
- Two types are compatible if internal structure is compatible

```
interface Person {  
    firstName: string;  
    lastName: string;  
}  
  
function greeter(person: Person) {  
    return "Hello, " + person.firstName + " " + person.lastName;  
}  
  
var user = { firstName: "Jane", lastName: "User" };  
  
document.body.innerHTML = greeter(user);
```

TypeScript: Arrow Func.

- shorter syntax for writing anonymous functions.

```
setInterval(function () { return _this.time = new Date(); }, 1000);
```



```
setInterval(() => this.time = new Date(), 1000);
```

Angular 2: Decorator

- Add functionality to a class
- @Component decorator

```
import {Component} from 'angular2/core';

@Component({
  selector: 'current-time',
  template: '<h1>Time: {{time}}</h1>'
})
export class CurrentTime {
  time: Date;

  constructor() {
    this.time = new Date();
    setInterval(() => this.time = new Date(), 1000);
  }
}
```

- Import module "Component"
 - no need <script>
- export means
 - can be imported
- Data binding

```
<body>
  <current-time>Loading...</current-time>
</body>
```

Ionic 2:resources

- Ionic website:
 - <http://ionicframework.com>
- Ionic 2 website
 - <http://ionicframework.com/docs/v2/>

HW2

- Install Ionic 2
- Create a blank app as follows.
 - `ionic start helloworld-blank blank -v2 -ts`
 - `ionic start helloworld-sidemenu sidemenu -v2 -ts`
 - `ionic start helloworld-tabs tabs-v2 -ts`
- Email to mp2016f@gmail.com by Monday 11am