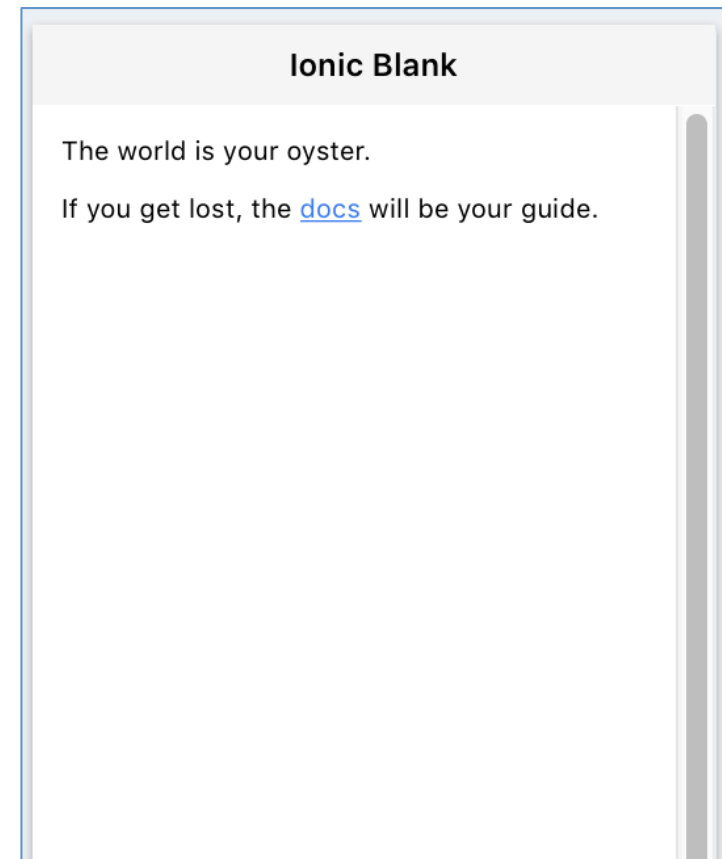


Ionic 2

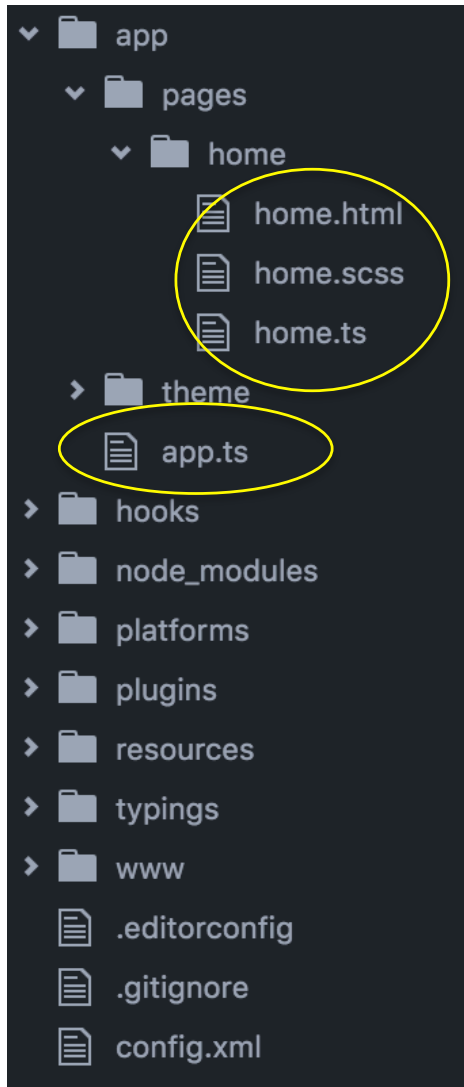
Getting started

Start with a blank app

- Create a project
 - `ionic start blankapp blank --v2 --ts`
- `cd blankapp`
- Add platforms
 - `ionic platform add android`
 - `ionic platform add ios`
- Test
 - In a browser
 - `ionic serve --lab`
 - Android emulator
 - `ionic run android`
 - iOS simulator
 - `ionic run ios`



Project structure



- app
 - All your codes go here
 - Organized by pages
 - Each page has: html, code, style
- WWW
 - contains all compiled webpage
 - DO NOT MODIFY

app.ts

```
import { Component } from '@angular/core';
import { ionicBootstrap, Platform } from 'ionic-angular';
import { StatusBar } from 'ionic-native';

import { HomePage } from '../pages/home/home';

@Component({
  template: '<ion-nav [root]="rootPage"></ion-nav>'
})
export class MyApp {
  rootPage: any = HomePage;

  constructor(public platform: Platform) {
    platform.ready().then(() => {
      // Okay, so the platform is ready and our plugins are
      // Here you can do any higher level native things you
      StatusBar.styleDefault();
    });
  }
}

ionicBootstrap(MyApp);
```

- Bootstrap an app
- Specify root page
- Associate with a view
- import
- @Component
- rootPage

import

- `import { <class name> } from <path>`
 - import only the specified class defined in the folder
 - <path>
 - predefined locations (found in *node_modules*)
 - @angular
 - ionic-angular
 - my locations (rooted to *current* directory)
 - ./pages/home/home (from app.ts)

```
import { Component } from '@angular/core';  
import { ionicBootstrap, Platform } from 'ionic-angular';  
import { StatusBar } from 'ionic-native';  
  
import { HomePage } from './pages/home/home';
```

} import classes from library

} Import my page classes
that needs to be referred

@Component decorator

- Define the class as a component
- Component?
 - reusable UI element, the building block
 - HTML+CSS+JS
 - HTML for Look, JS for model/behavior
- @Component
 - **template**: inline HTML
 - **templateURL**: refer to HTML file
 - **selector**: tag for this component
 - **providers**: services we want to inject to the component

```
@Component({  
  template: '<ion-nav [root]="rootPage"></ion-nav>'  
})  
export class MyApp {
```

*property-to-variable binding
[<property>]="variable"*



class MyApp

- `rootPage: any = HomePage`
 - define a member var `rootPage` of any type, assigned to `HomePage` component
 - Later referred by `<ion-nav>` element
 - `<ion-nav [root]="rootPage"></ion-nav>`
- `constructor()`
- `StatusBar (plugin) (`
<http://ionicframework.com/docs/v2/native/Status%20Bar/>)
 - native looking statusbar
 - `styleDefault()`: set default style

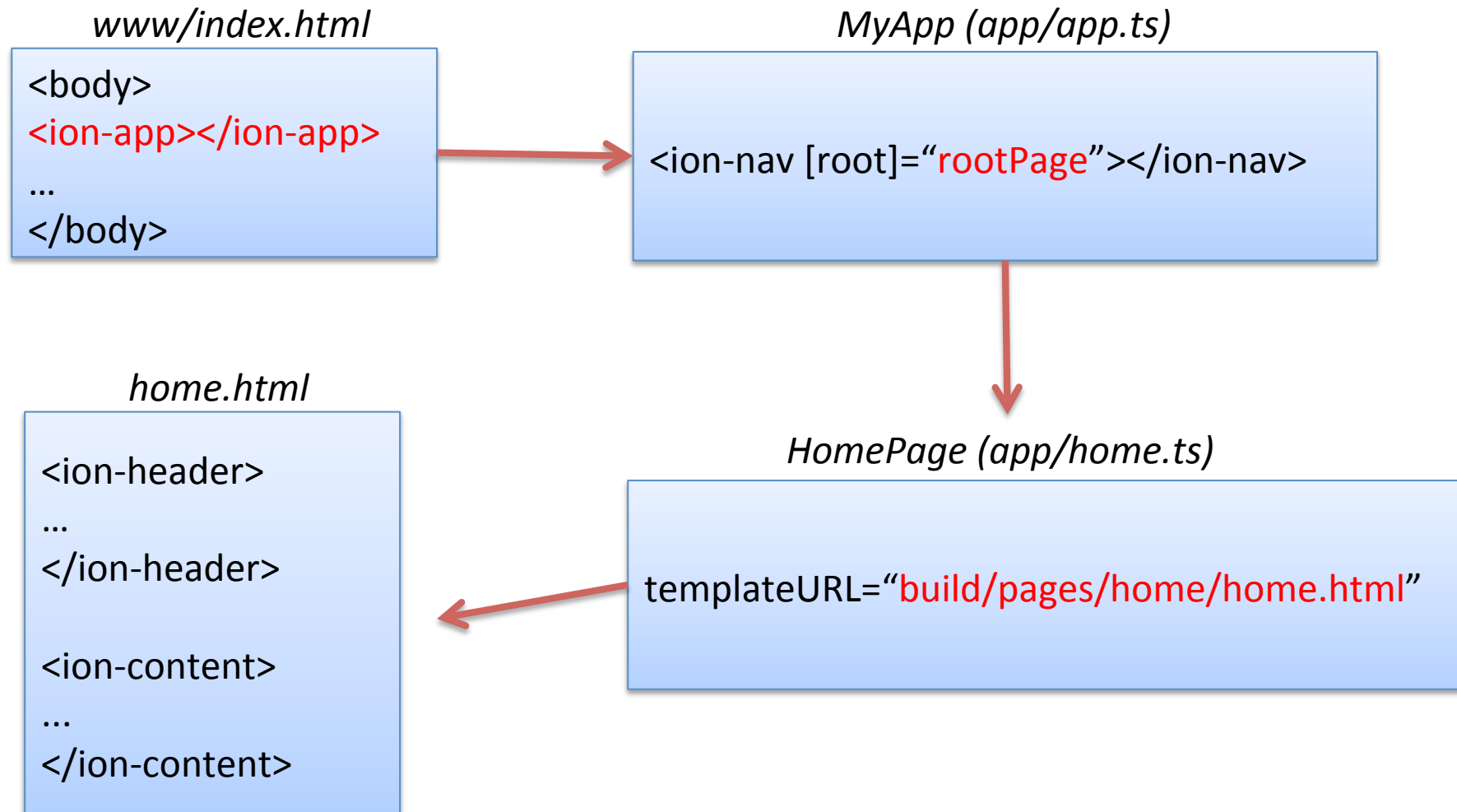
```
export class MyApp {  
  rootPage: any = HomePage;  
  
  constructor(public platform: Platform) {  
    platform.ready().then(() => {  
      // Okay, so the platform is ready and  
      // Here you can do any higher level n  
      StatusBar.styleDefault();  
    });  
  }  
}
```

ionicBootstrap

- bootstrap your entire application
- 1st param: a root component
- 2nd param: array of providers injected
- 3rd param: config object
 - ex: {tabsPlacement: 'bottom'}

```
ionicBootstrap(MyApp);
```


Roadmap



Home.ts/home.html

home.ts

```
<ion-header>
  <ion-navbar>
    <ion-title>
      Ionic Blank
    </ion-title>
  </ion-navbar>
</ion-header>
```

```
<ion-content padding>
  The world is your oyster.
  <p>
    If you get lost, the <a href="http
  </p>
</ion-content>
```

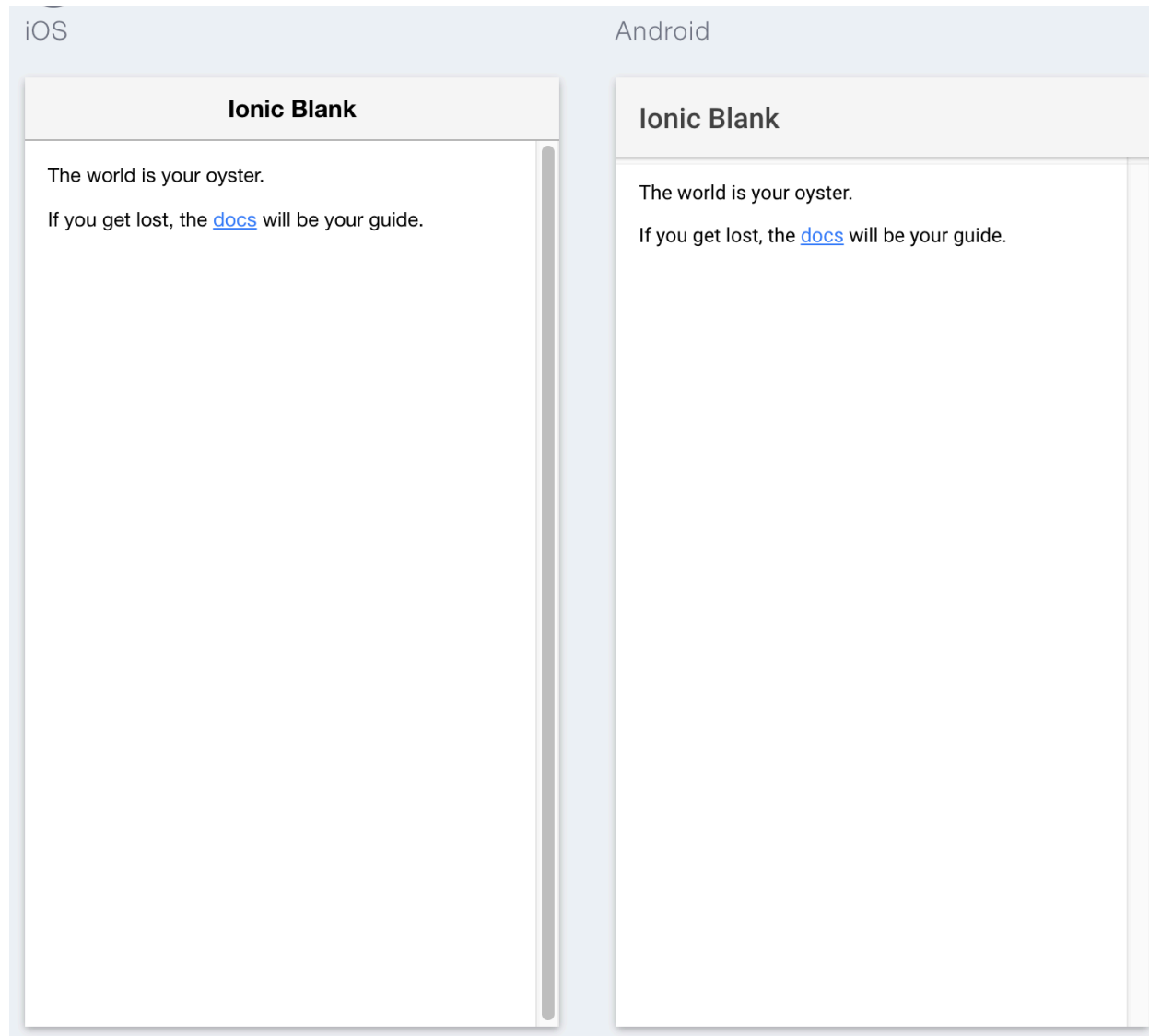
```
import { Component } from '@angular/core';
import { NavController } from 'ionic-angular';

@Component({
  templateUrl: 'build/pages/home/home.html'
})
export class HomePage {
  constructor(public navCtrl: NavController) {

  }
}
```

home.html

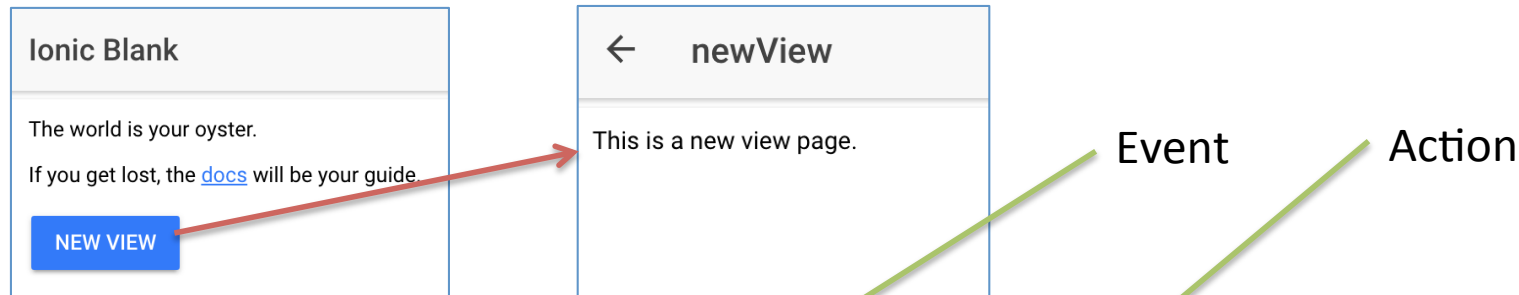
Ionic Lab



Add a new page

- New page
 - name: NewView
- Generate a page
 - `ionic g page NewView`
 - creates `pages/new-view/new-view.{ts,html,scss}`
- CSS Housekeeping
 - import `"../pages/new-view/new-view"` in `theme/app.core.scss`

Navigation (by Link)



home.html

```
<button (click) = "goNewView()">New View</button>
```

home.ts

```
import { Component } from '@angular/core';
import { NavController } from 'ionic-angular';
import { NewViewPage } from '../new-view/new-view'

@Component({
  templateUrl: 'build/pages/home/home.html'
})
export class HomePage {
  constructor(public navCtrl: NavController) {

  }
  goNewView() { this.navCtrl.push(NewViewPage)}
}
```

} import target comp

} push new page

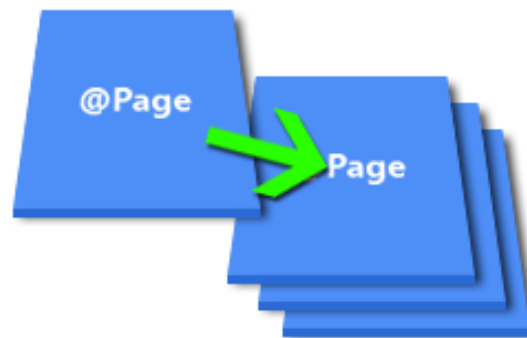
Navigation Stack

- Pages are stacked
- Push new page on the stack
- Pop current page to go back to previous page

Navigation Stack



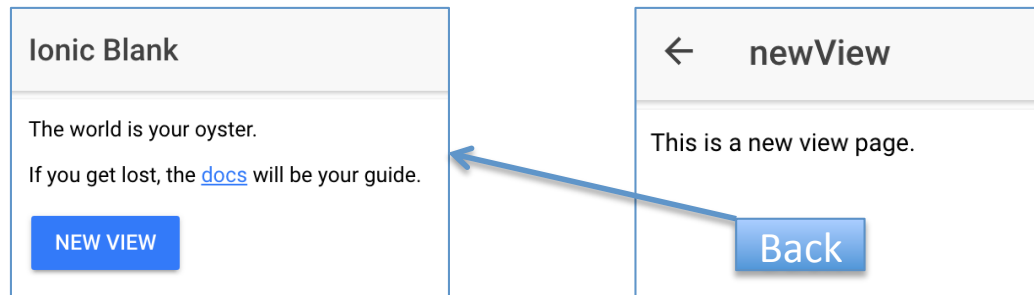
Push



Pop



Navigation Back (by Link)



HW3

1. Navigation by Link

1. Blank App 생성
 2. 새로운 페이지 New 생성
 3. Blank App에 “Go to New” 버튼 생성
 4. 버튼을 누르면 New 페이지로 이동하도록 구현
- 기한: 수요일 수업시간 시작
 - 제출:
 - pages 폴더 전체 압축해서 제출