

Ionic Study

Day 10

오늘 할 것들

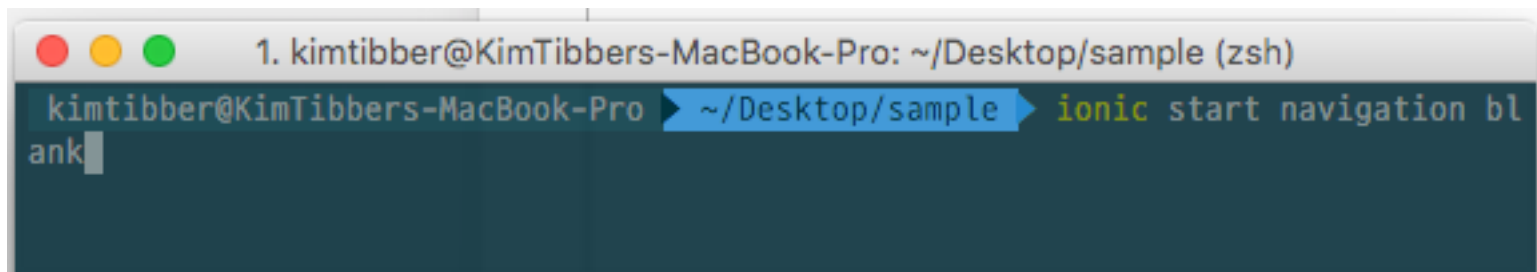
- 화면 네비게이션 알아보기
 - 화면 이동
 - 라이프 사이클
 - 루트 페이지 변경
 - 페이지간 파라미터 전달

화면 네비게이션

- 루트 페이지
 - 앱의 첫 페이지
 - 앱이 구동되면 `src/app` 폴더의 `app.component.ts`에 정의된 `MyApp`를 생성.
 - `MyApp` 생성자 내의 `rootPage` 멤버 변수에 지정된 객체를 앱의 시작점으로 정의.

화면 네비게이션

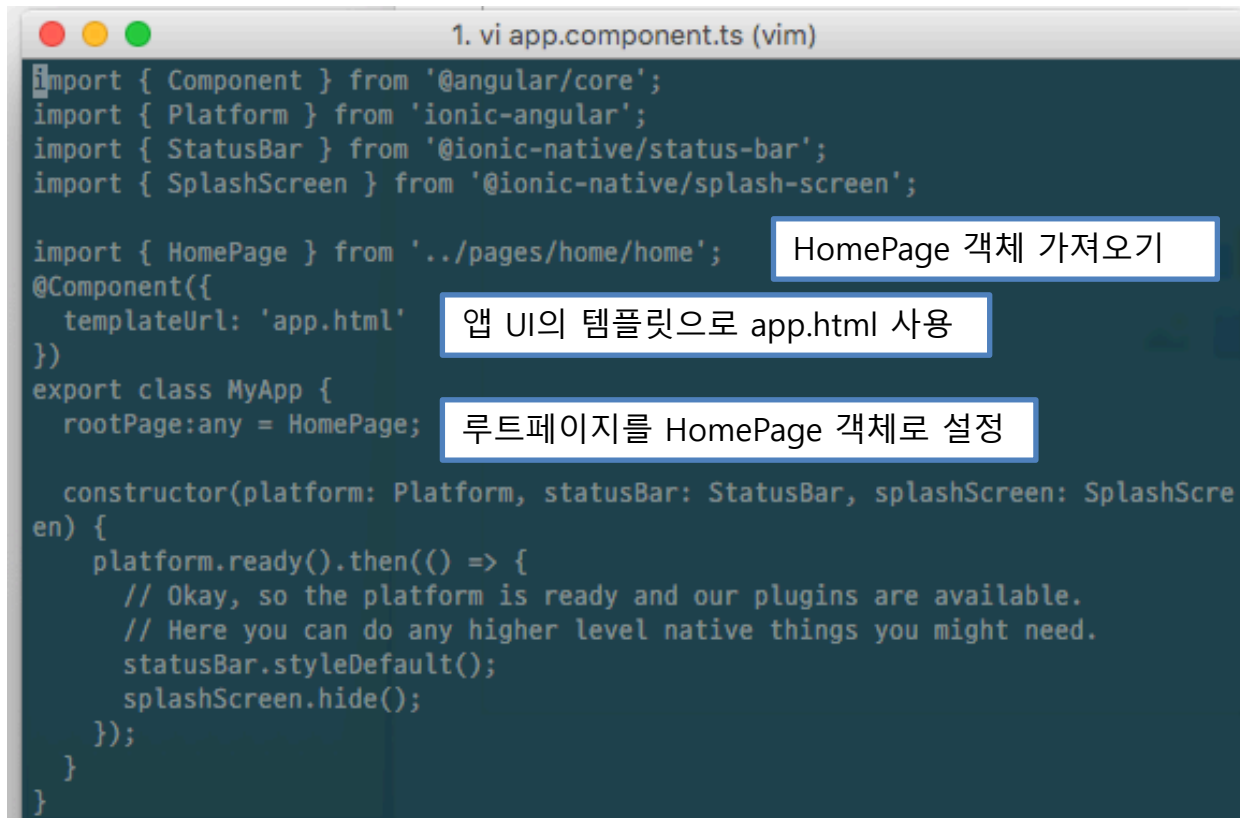
- ionic start navigation blank
 - navigation이라는 프로젝트명을 생성
 - 템플릿은 blank 템플릿을 사용



```
1. kimtibber@KimTibbers-MacBook-Pro: ~/Desktop/sample (zsh)
kimtibber@KimTibbers-MacBook-Pro ~/Desktop/sample$ ionic start navigation blank
```

화면 네비게이션

- app.component.ts



```
1. vi app.component.ts (vim)
import { Component } from '@angular/core';
import { Platform } from 'ionic-angular';
import { StatusBar } from '@ionic-native/status-bar';
import { SplashScreen } from '@ionic-native/splash-screen';

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

@Component({
  templateUrl: 'app.html'
})
export class MyApp {
  rootPage:any = HomePage;

  constructor(platform: Platform, statusBar: StatusBar, splashScreen: SplashScreen) {
    platform.ready().then(() => {
      // Okay, so the platform is ready and our plugins are available.
      // Here you can do any higher level native things you might need.
      statusBar.styleDefault();
      splashScreen.hide();
    });
  }
}
```

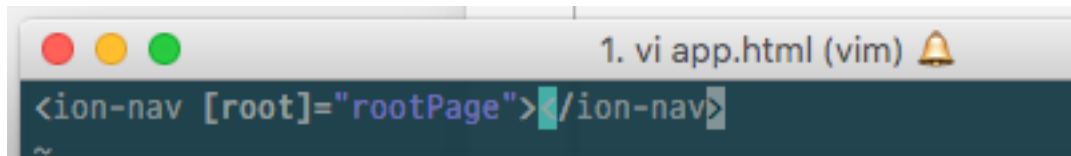
HomePage 객체 가져오기

앱 UI의 템플릿으로 app.html 사용

루트페이지를 HomePage 객체로 설정

화면 네비게이션

- app.html

A screenshot of a vim editor window. The title bar shows "1. vi app.html (vim)" and a bell icon. The editor content shows the HTML tag `<ion-nav [root]="rootPage"></ion-nav>` on a single line. The cursor is positioned at the end of the closing tag. A tilde (~) is visible on the line below.

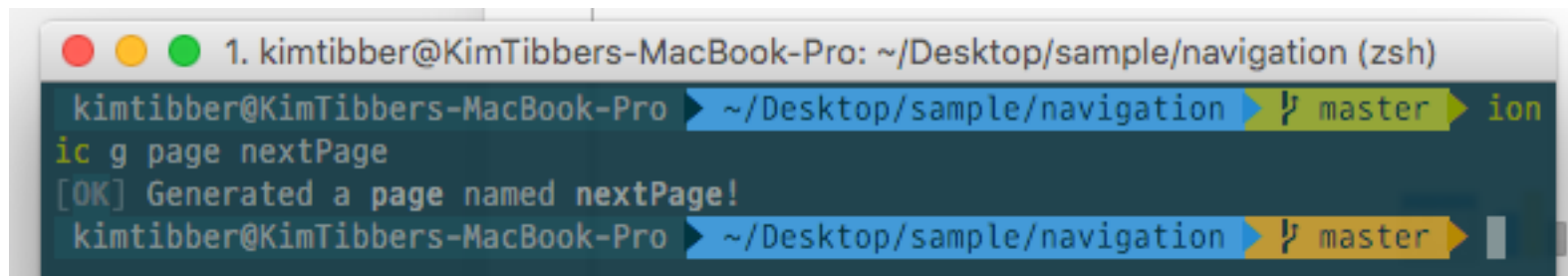
```
1. vi app.html (vim) 🔔  
<ion-nav [root]="rootPage"></ion-nav>  
~
```

NavController

- 페이지의 히스토리를 관리하는 페이지 객체형의 배열
- 페이지의 생성자 파라미터에 전달이 가능한 객체
- 페이지 이동이 가능하며 각각 push, pop 함수 사용
 - push : 새로운 페이지로의 전환 시 사용
 - pop : 현재 페이지를 나가 이전 페이지로 전환 시 사용

NavController

- navigation 프로젝트에서 계속 진행
- ionic g page nextPage
 - 현재 프로젝트에 신규 페이지를 추가하는 명령 (g: generate의 alias)



```
1. kimtibber@KimTibbers-MacBook-Pro: ~/Desktop/sample/navigation (zsh)
kimtibber@KimTibbers-MacBook-Pro > ~/Desktop/sample/navigation > master > ion
ic g page nextPage
[OK] Generated a page named nextPage!
kimtibber@KimTibbers-MacBook-Pro > ~/Desktop/sample/navigation > master >
```


NavController

- src/pages/home/home.html

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

<ion-content padding>
  <button ion-button (click)="moveToNextPage()">다음 페이지로 </button>
</ion-content>
```

NavController

- src/pages/home/home.ts

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

@Component({
  selector: 'page-home',
  templateUrl: 'home.html'
})
export class HomePage {

  constructor(public navCtrl: NavController) {

  }

  moveToNextPage() {
    this.navCtrl.push(NextPage);
  }
}
```

NavController

- src/app/app.module.ts

```
9 | import { NextPageModule } from '../pages/next/next.module';
10 |
11 | @NgModule({
12 |   declarations: [
13 |     MyApp,
14 |     HomePage
15 |   ],
16 |   imports: [
17 |     BrowserModule,
18 |     IonicModule.forRoot(MyApp),
19 |     NextPageModule
20 |   ]
21 | })
22 | export class AppModule {}
```

NavController

- src/pages/next/next.html

```
9      <ion-navbar [hideBackButton]="true">
10        <ion-title>nextPage</ion-title>
11      </ion-navbar>
12
13    </ion-header>
14
15
16    <ion-content padding>
17      <button ion-button (click)="moveToHomePage()">홈 페이지로 고고씹~</button>
18    </ion-content>
```

hideBackButton 속성을 true로 주
게 되면 네비게이션 바의 Back 버튼
이 사라짐.

NavController

- src/pages/next/next.ts

```
16  export class NextPage {
17
18      constructor(public navCtrl: NavController) {}
19
20
21      ionViewDidLoad() {
22          console.log('ionViewDidLoad NextPage');
23      }
24
25      moveToHomePage() {
26          this.navCtrl.pop();
27      }
28
```

페이지 이벤트

- 페이지의 상태가 변하면 이벤트가 발생함
 - 이벤트 핸들러 (Event handler)
 - 8가지의 이벤트 핸들러 지원.
 - `ionViewDidLoad`, `ionViewWillEnter`,
`ionViewDidEnter`, `ionViewWillLeave`,
`ionViewDidLeave`, `ionViewWillUnload`,
`ionViewCanEnter`, `ionViewCanLeave`

NavController

- src/pages/home/home.ts (생성자 밑에 작성)

```
19  ionViewDidLoad() {  
20    console.log('[HomePage] ionViewDidLoad');  
21  }  
22  
23  ionViewWillEnter() {  
24    console.log('[HomePage] ionViewWillEnter');  
25  }  
26  
27  ionViewDidEnter() {  
28    console.log('[HomePage] ionViewDidEnter');  
29  }  
30  
31  ionViewWillLeave() {  
32    console.log('[HomePage] ionViewWillLeave');  
33  }  
34  
35  ionViewDidLeave() {  
36    console.log('[HomePage] ionViewDidLeave');  
37  }  
38  
39  ionViewWillUnload() {  
40    console.log('[HomePage] ionViewWillUnload');  
41  }  
42  
43  ionViewCanEnter() {  
44    console.log('[HomePage] ionViewCanEnter');  
45  }  
46  
47  ionViewCanLeave() {  
48    console.log('[HomePage] ionViewCanLeave');  
49  }
```

NavController

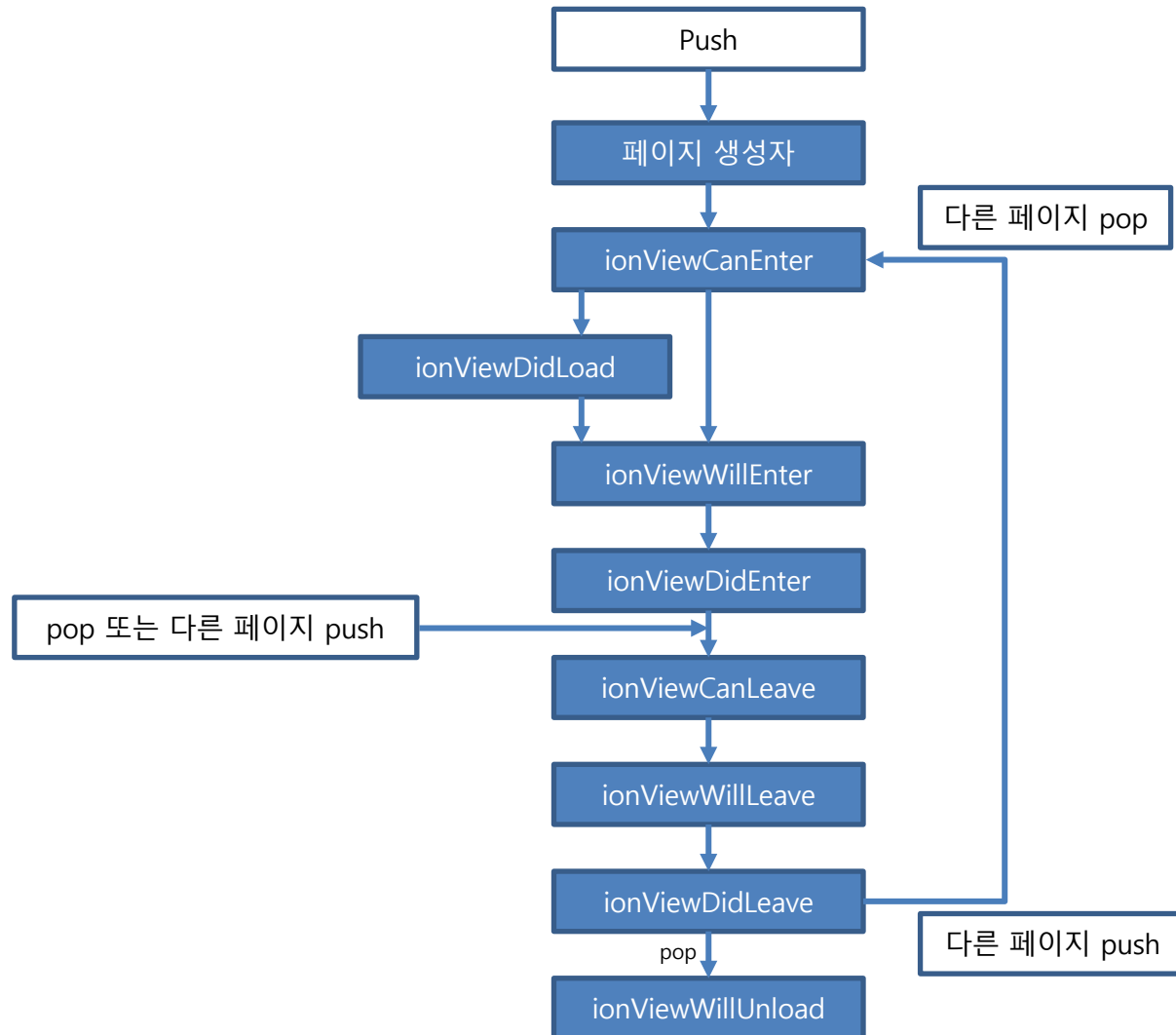
- src/pages/next/next.ts (생성자 밑에 작성)

```
29  ionViewWillEnter() {  
30    console.log('[NextPage] ionViewWillEnter');  
31  }  
32  
33  ionViewDidEnter() {  
34    console.log('[NextPage] ionViewDidEnter');  
35  }  
36  
37  ionViewWillLeave() {  
38    console.log('[NextPage] ionViewWillLeave');  
39  }  
40  
41  ionViewDidLeave() {  
42    console.log('[NextPage] ionViewDidLeave');  
43  }  
44  
45  ionViewWillUnload() {  
46    console.log('[NextPage] ionViewWillUnload');  
47  }  
48  
49  ionViewCanEnter() {  
50    console.log('[NextPage] ionViewCanEnter');  
51  }  
52  
53  ionViewCanLeave() {  
54    console.log('[NextPage] ionViewCanLeave');  
55  }
```


페이지 이벤트

- 이벤트 핸들러 설명
 - `ionViewDidLoad`
 - 페이지가 로드되면 호출
 - `ionViewWillEnter`
 - 페이지가 보여지기 전 호출되는 함수
 - `ionViewWillLeave`
 - 페이지가 가려지기 전 호출되는 함수
 - `ionViewDidLeave`
 - 페이지가 가려진 후 호출되는 함수
 - `ionViewWillUnload`
 - 페이지가 삭제되기 전 호출되는 함수
 - `ionViewCanEnter`
 - 페이지의 생성자가 호출된 이후 호출. 페이지가 보여지기 전, 페이지로 진입 가능한지 여부를 체크할 때 사용
 - `return true, undefined` : 진입 가능한 것으로 체크, `return false` : 진입 불가로 체크
 - `ionViewCanLeave`
 - 페이지가 사라지기 전, 페이지로 나갈 수 있는지 여부를 체크할 때 사용.
 - `return true, undefined` : 나갈 수 있는 것으로 체크, `return false` : 탈출 불가로 체크

이벤트 발생 Flow



루트 페이지 변경

- 필요에 따라 루트 페이지를 갈아치워야 하는 경우가 생김(!)
 - 상태에 따라 첫(루트) 페이지를 교체해야 하는 경우 요긴하게 사용.
 - NavController 객체의 `setRoot` 함수 사용.
 - 기존 루트 페이지는 메모리에서 제거됨
 - 덤으로 `onViewWillUnload` 이벤트 핸들러도 호출.

루트 페이지 변경

- src/pages/home/home.html
 - 앞서 작성한 코드의 10번째 줄 아래에 버튼 새로 추가

```
11 | <button ion-button (click)="moveToNextPageWithSetRoot()">넌 이제 필요없어!</button>
```

- src/pages/home/home.ts
 - 신규 멤버함수 추가

```
51 | moveToNextPageWithSetRoot() {  
52 |   this.navCtrl.setRoot(NextPage);  
53 | }  
54 |
```

페이지간 파라미터 전달

- 현재 페이지에 입력받은 사용자의 값을 다음 페이지로 전달할 때 사용.
- NavParams 클래스와 push, setRoot 함수의 파라미터를 이용하여 구현이 가능.

페이지간 파라미터 전달

- ionic start param blank
 - blank 템플릿으로 param 프로젝트 생성
- ionic g page nextPage
 - nextPage 추가

페이지간 파라미터 전달

- src/pages/home/home.html

```
9   <ion-content padding>
10   <ion-item>
11     <ion-label>숫자 입력</ion-label>
12     <ion-input type="number" [(ngModel)]="input" placeholder="숫자를 입력해주삼"></ion-input>
13   </ion-item>
14   <button ion-button (click)="moveToNextPage()">다음 페이지로 고고씹</button>
15 </ion-content>
```

페이지간 파라미터 전달

- src/pages/home/home.ts

```
1  import { Component } from '@angular/core';
2  import { NavController } from 'ionic-angular';
3  import { NextPage } from '../next/next';
4
5  @Component({
6    selector: 'page-home',
7    templateUrl: 'home.html'
8  })
9  export class HomePage {
10   input;
11
12   constructor(public navCtrl: NavController) {
13
14   }
15
16   moveToNextPage() {
17     this.navCtrl.push(NextPage, {number: this.input});
18   }
19
20 }
```


페이지간 파라미터 전달

- src/app/app.module.ts

```
9 | import { NextPageModule } from '../pages/next/next.module';
10 |
11 | @NgModule({
12 |   declarations: [
13 |     MyApp,
14 |     HomePage
15 |   ],
16 |   imports: [
17 |     BrowserModule,
18 |     IonicModule.forRoot(MyApp),
19 |     NextPageModule
```

페이지간 파라미터 전달

- src/pages/next/next.html

```
16 <ion-content padding>
17   Number from HomePage : {{number}}
18 </ion-content>
19
```

- src/pages/next/next.ts

```
15 //
16 export class NextPage {
17   number:number;
18
19   constructor(public navCtrl: NavController, public navParams: NavParams) {
20     this.number = navParams.get('number');
21   }
22
23   ionViewDidLoad() {
24     console.log('ionViewDidLoad NextPage');
25   }
26
27 }
```

오늘은 여기까지~

See you next day!