

Angular2 Component

Form

Angular에서 폼을 처리하는 방식은 템플릿 기반 폼 방식과 반응형 폼 방식이 있다.

템플릿 기반의 폼은 템플릿에서 디렉티브를 사용하고 반응형 폼은 클래스 코드에서 TypeScript를 이용하여 폼 컨트롤을 사용한다.

템플릿 기반 => 단순한 체크. NgModel, NgModelGroup, NgForm 사용.

반응형 폼 => 복잡한 구조. 브라우저가 없어도 테스트 가능

모델은 @angular/forms 모듈의 FormControl, FormGroup, FormArray를 사용하는 개체이어야 한다.

NgForm

- `<form ngForm>` 형태로 사용
- `<form #f = ngForm>` 형태로 Alias 사용 가능
- NgForm이 추가된 form 내부에서 ngModel이 달린 요소를 Form model에 추가. `[(ngModel)]`도 사용 가능.
- submit 이벤트를 가로채서 서버로 전달을 막는다. 서버로 값 전달은 ngSubmit 사용.
- `<form #fm = "ngForm" (ngSubmit)="onSubmit(fm.value)">...</form>`
- 일반 HTML form으로 사용하기 위해서는 ngNoForm을 사용한다.
`<form ngNoForm>...</form>`

NgModel

ngModel 형태로 기술한다. 이는 바인딩 목적이 아닌 Form model 을 구성하는 용도로 사용. 이때 name 어트리뷰트는 값 참조를 위해 지정(필수).

```
<input name="name" ngMdoel>
```

ngForm과 마찬가지로 Alias 사용 가능. **ngModel**을 두 번 기술.

```
<input name="name" #n = ngModel ngModel> {{n.value}}
```

NgFormGroup

폼 필드를 묶어서 참조할 목적으로 사용.

```
<form #fm = ngForm>
```

```
  <div ngFormGroup="info">
```

```
    <input type="text" name="name" #n = ngModel>
```

```
    <input type="text" name="age" #a = ngModel>
```

```
  </div>
```

```
</form>
```

```
{{fm.value.info.n.value}} / {{fm.value.info.a.value}}
```

Component
