



**One framework.
Mobile & desktop.**

8강. Pipe



DEVELOP ACROSS ALL PLATFORMS

Learn one way to build applications with Angular and reuse your code and abilities to build apps for any deployment target. For web, mobile web, native mobile and native desktop.

8 강.

Pipe

□ 1) Pipe 개요

* 파이프 (pipe)

- Angular의 파이프는 문자열을 입력 받아 읽기 쉬운 형태로 포맷을 변경하여 출력한다.

```
{{ 값 | 파이프:표현식 }}
```

* 파이프의 장점

- 가독성 향상 : 적절한 파이프 사용은 데이터를 읽기 쉽게 만들어 가독성이 향상됨.
- 코드 생산성 증가: 데이터 형식을 결정하기 위해 컴포넌트마다 로직이 불필요.
- 유지 보수 용이: 데이터 출력에 일관성이 생기고 유지보수가 쉬워짐.

<https://angular.io/guide/pipes>

□ 2) 날짜 Pipe

가. 날짜 Pipe

```
{{ 값 | date:'y/M/dd' }}
```

이름	1자리 표현	2자리 표현
년도	y (2020)	yy (20)
월	M(6)	MM(06)
일	d(12)	dd(12)
시간	h/H	hh/HH
분	m	Mm
초	s	ss

□ 2) 날짜 Pipe

app.component.html X

y-app > src > app > <> app.component.html > ...

```

1 <h1>{{title}}</h1>
2 <h2>날짜 포맷 지정</h2>
3 <ul>
4   <li>01:{{currentDate}}</li>
5   <li>02:{{currentDate|date}}</li>
6   <li>03:{{currentDate|date:'yy y'}}</li>
7   <li>04:{{currentDate|date:'MM M dd d'}}</li>
8   <li>05:{{currentDate|date:'y/MM/dd HH:mm:ss'}}</li>
9   <li>05:{{currentDate|date:'y년 MM월 dd일 HH:mm:ss'}}</li>
10 </ul>

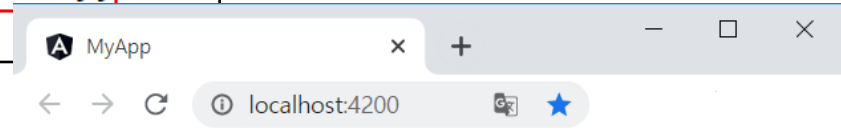
```

```

export class AppComponent {
  title = '날짜 pipe 실습';

  currentDate:Date = new Date();
}

```



날짜 pipe 실습

날짜 포맷 지정

- 01:Fri Jun 12 2020 09:16:39 GMT+0900 (대한민국 표준시)
- 02:Jun 12, 2020
- 03:20 2020
- 04:06 6 12 12
- 05:2020/06/12 09:16:39
- 05:2020년 06월 12일 09:16:39

□ 3] 통화 Pipe

나. 통화 파이프

```
{{ 값 | currency:'통화코드':true:'숫자형식' }}
```

app.component.html ✕

/-app > src > app > <> app.component.html > ...

```
1 <h1>{{title}}</h1>
2 <h2>통화 포맷 지정</h2>
3 <ul>
4   <li>01:{{1000.125}}</li>
5   <li>02:{{1000.125|currency}}</li>
6   <li>03:{{1000.125|currency:'USD':false}}</li>
7   <li>04:{{1000.125|currency:'USD':true}}</li>
8   <li>04:{{1000.125789|currency:'KRW':true:'4.1-4'}}</li>
9 </ul>
```

통화 포맷 지정

- 01:1000.125
- 02:\$1,000.13
- 03:USD1,000.13
- 04:\$1,000.13
- 04:₩1,000.1258

❑ 4) decimal Pipe



다. decimal 파이프

- 파이프에 입력된 숫자 값에 대한 자리 수 형식을 지정.

{{ 값 | number:'자리수' }}

```
app.component.html ×
/-app > src > app > <> app.component.html > ...
1  <h1>{{title}}</h1>
2  <h2>decimal 포맷 지정</h2>
3  <ul>
4    <li>01:{{2.7188}}</li>
5    <li>02:{{2.7188|number}}</li>
6    <li>03:{{2.7188|number:'3.1-1'}}</li>
7    <li>04:{{2.7188|number:'3.1-2'}}</li>
8    <li>05:{{2.7188|number:'3.1-3'}}</li>
9    <li>06:{{2.7188|number:'3.1-4'}}</li>
10 </ul>
```

decimal 포맷 지정

- 01:2.7188
- 02:2.719
- 03:002.7
- 04:002.72
- 05:002.719
- 06:002.7188

□ 5) 문자열 Pipe

라. 문자열 파이프

- 파이프에 입력된 문자열에 대한 함수 적용.

{{ 값 | 함수 }}

app.component.html ×

/-app > src > app > <> app.component.html > ...

```
1 <h1>{{title}}</h1>
2 <h2>문자열 포맷 지정</h2>
3 <ul>
4   <li>01:{{"Happy New Year!!!"}}</li>
5   <li>02:{{"Happy New Year!!!"|lowercase}}</li>
6   <li>03:{{"Happy New Year!!!"|uppercase}}</li>
7   <li>04:{{"Happy New Year!!!"|slice:0:3}}</li>
8   <li>05:{{"Happy New Year!!!"|slice:-7:-3}}</li>
9 </ul>
```

문자열 포맷 지정

- 01:Happy New Year!!!
- 02:happy new year!!!
- 03:HAPPY NEW YEAR!!!
- 04:Hap
- 05:Year

❑ 6] JSON Pipe

마. i18nSelect 파이프

JSON 객체의 키에 대응하는 값을 찾아 출력한다.

```
{{ 키값 |i18nSelect:객체}}
```

```
export class AppComponent {  
  title = 'i18n JSON pipe 실습';  
  
  items={  
    apple:'사과',  
    melon:'멜론',  
    banana:'바나나',  
    orange:'오렌지',  
  }  
  
  best_item='orange'  
}
```

```
app.component.html X  
y-app > src > app > <> app.component.html > ...  
1 <h1>{{title}}</h1>  
2 <ul>  
3   <li>01:{{'apple'|i18nSelect:items}}</li>  
4   <li>02:{{'melon'|i18nSelect:items}}</li>  
5   <li>03:{{best_item|i18nSelect:items}}</li>  
6 </ul>
```

- 01:사과
- 02:멜론
- 03:오렌지

바. JSON 파이프

- JSON객체를 문자열로 변환하고자 할 때 사용한다.
- JSON은 template에 바로 출력하면 [object Object] 형식으로 출력되기 때문에 문자열로 변환이 필요하다.

```
{{객체|json}}
```

```
export class AppComponent {  
  title = 'JSON pipe 실습';
```

```
  items={  
    apple:'사과',  
    melon:'멜론',  
    banana:'바나나',  
    orange:'오렌지',  
  }
```

app.component.html ×

```
/-app > src > app > <> app.component.html > ...  
1 | <h1>{{title}}</h1>  
2 | <ul>  
3 |   <li>01:{{items}}</li>  
4 |   <li>02:{{items|json}}</li>  
5 | </ul>
```

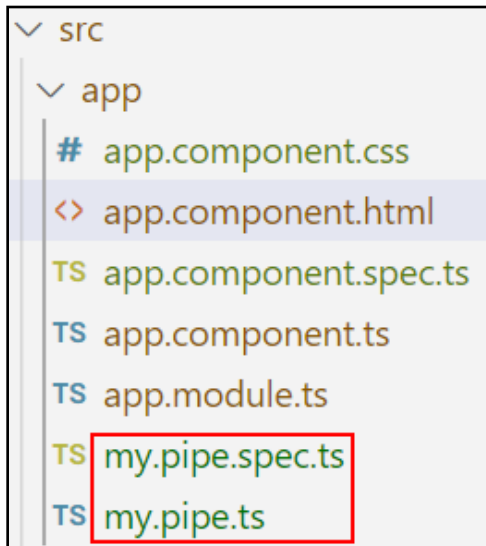
□ 8] 사용자 정의 Pipe

* 사용자 정의 pipe 생성 방법

-내장 파이프로 원하는 출력 데이터를 만들 수 없을 때 직접 정의 가능하다.

```
ng g pipe My // Creates MyPipe
```

1. 사용자 pipe를 생성한다.



```
C:\angular\my-app>ng g pipe My
CREATE src/app/my.pipe.spec.ts (171 bytes)
CREATE src/app/my.pipe.ts (209 bytes)
UPDATE src/app/app.module.ts (362 bytes)
```

□ 8] 사용자 정의 Pipe

2. Transform 메서드를 구현한다.

```
export class AppComponent {
  title = '사용자정의 pipe 실습';
  currentDate = "20200612";
}
```

```
현재날짜:{{currentDate|my}}<br>
현재날짜2:{{currentDate|my:"/"}}<br>
현재날짜2:{{currentDate|my:"-"}}<br>
```

```
@Pipe({
  name: 'my'
})
export class MyPipe implements PipeTransform {

  transform(value: string, ...args: string[]): string {
    if(value.length==8 && args != undefined){
      return value.substring(0,4)+args+value.substring(4,6)+args+value.substring(6,8);
    }
  }
}
```

현재날짜:20200612
 현재날짜2:2020/06/12
 현재날짜2:2020-06-12

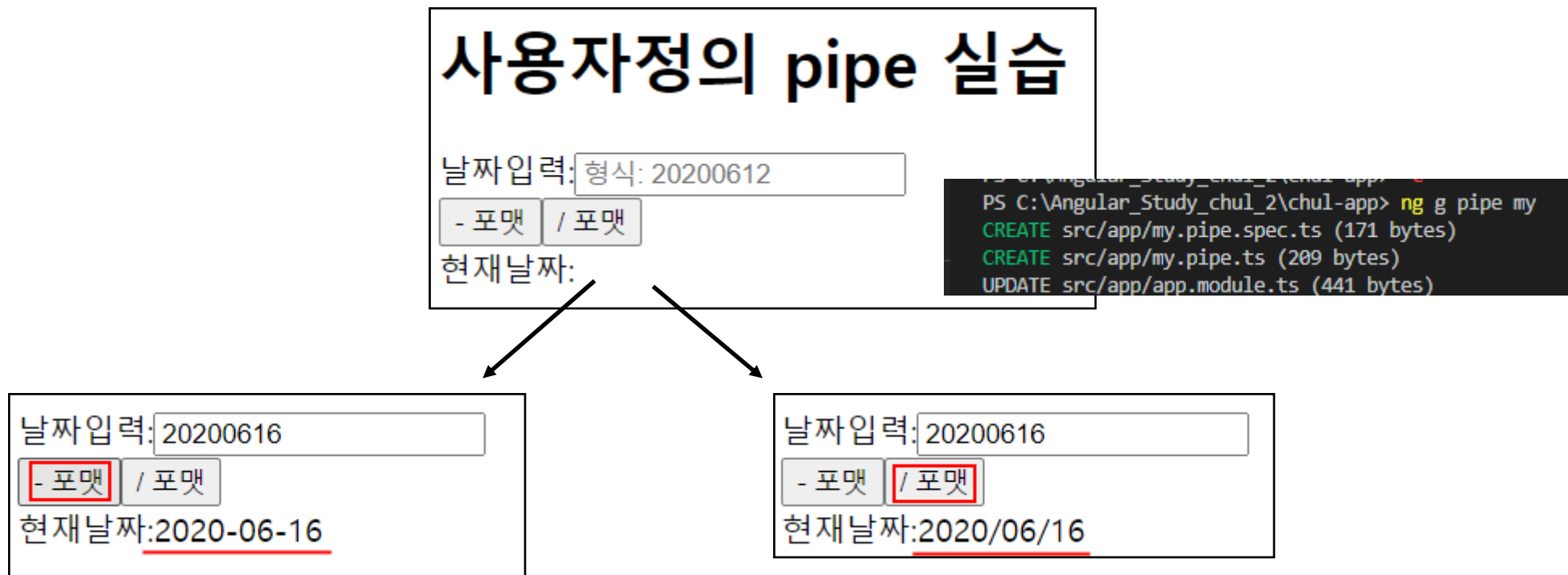
□ 실습 예제 1

```
PS C:\Angular_Study_chul_2\chul-app> ng serve
```



다음과 같은 초기화면에서 형식에 맞게 날짜를 입력하고 버튼을 클릭했을 때, 포맷을 적용한 날짜 정보를 출력하는 코드를 구현 하시오.

- “- 포맷” 버튼을 클릭하면 2020-06-15 형식으로 출력하고
“/ 포맷” 버튼을 클릭하면 2020/06/15 형식으로 출력한다.



```
c > app > TS app.module.ts > AppModule
1  import { BrowserModule } from '@angular/platform-browser'
2  import { NgModule } from '@angular/core';
3  import {FormsModule} from '@angular/forms';
4
5  import { AppRoutingModule } from './app-routing.module';
6  import { AppComponent } from './app.component';
7  import { MyPipe } from './my.pipe';
8
9  @NgModule({
10   declarations: [
11     AppComponent,
12     MyPipe
13   ],
14   imports: [
15     BrowserModule,
16     AppRoutingModule,
17     FormsModule
18   ],
19   providers: []
```



수고하셨습니다.
