

Angular2 Pipe

Pipe

어떠한 값을 받아 원하는 형태의 출력 결과를 작성하여, View에 반영하도록 처리하는 Class 파일이다.

사용법

```
{{ value | pipeName [: option, : option..] }}
```

사용 예

```
{{ new Date | date : ' yyyy-MM-dd hh:mm:ss a EEEE ' }} => 2019-12-25 12:35:25 FM Sunday
```

```
{{ 12345678 | currency | ' USD ' : true : ' 1.1-1 ' }}      => $12,234,678.0
```

Pipe 장점

- 가독성 향상
적절한 파이프 사용은 데이터를 읽기 쉽게 만들어 가독성을 향상 시킨다.
- 코드 생산성 증가:
데이터 형식을 결정하기 위해 컴퍼넌트마다 로직을 추가할 필요가 없다
- 유지보수 용의
파이프를 사용하면 데이터 출력에 일관성이 보장되며 유지 보수가 간결해 진다

AngularJS Filter & Angular Pipe

AngularJS의 Filter가 이름만 Pipe로 변경되었다.

Filter와 Pipe 비교

기능	AngularJS 지원	Angular 지원
date, currency, uppercase, lowercase, json	지원 함	지원 함
Number, limitTo, filter, orderBy	지원 함	지원 안함
decimal, percent, slice	지원 안함	지원 함
Custom Filter & Pipe	지원 함	지원 함

내장 Pipe

- 내장 Pipe를 사용하기 위해서는 필요한 모듈을 Component에서 import 할 필요가 있다.
- 필터 사용을 위해 providers: [PipeConstructor, ...] 형태로 Component에 등록이 필요하다.
- 내장 파이프는 Module의 declarations에 등록할 필요가 없이 바로 사용 가능하다. (Custom 파이프는 등록 필요)

DatePipe

날짜를 표시하기 위한 Pipe.

사용법

```
{{ value | date [: option, : option..] }}
```

Value는 Date Type, String Type, Number Type을 지원한다

사용 예

```
{{ new Date | date : ' yyyy-MM-dd hh:mm:ss a EEEE ' }}
```

=> 2019-12-25 12:35:25 FM Sunday

```
{{ 78451282 | date : ' yyyy-MM-dd hh:mm:ss a EEEE ' }}
```

=> 2019-12-25 12:35:25 FM Sunday

DatePipe

날짜를 설정을 위한 심볼

구분	단축표현	전체 표현
연도	y (2017)	yy (17)
월	M (3)	MM (03)
일	d (9)	dd (09)
시간	j / h / H (2PM / 5 / 5)	jj / hh / HH (02PM / 05 / 05)
분	m (5)	mm (05)
초	s (5)	ss (05)
요일	EE (Sun)	EEEE (Sunday)
AM PM	a	a

DatePipe

날짜를 설정을 위한 심볼

단축어	표현	단축어	표현
날짜			
fullDate	Friday, September 8, 2017	longDate	September 8, 2017
mediumDate	Sep 8, 2017	shortDate	9/8/2017
시간			
mediumTime	3:15:28 PM	shortTime	3:15 PM
전체			
medium	Sep 8, 2017, 3:15:28 PM	short	9/8/2017, 3:15 PM

CurrencyPipe

통화 표시를 표시하기 위한 Pipe. 통화 코드 ISO 4217 규격에 따름.

https://ko.wikipedia.org/wiki/ISO_4217

사용법

```
{{ value | currency [: option, : option..] }}
```

Value는 Number Type을 지원한다

사용 예

```
{{123456789 | currency }}
```

=> USD123,456,789.00

```
{{1234.56789 | currency : 'USD' : true : '1.2-3'}}
```

=> \$1,234.568

PercentPipe

value * 100 + ' % ' 형태로 연산 후 적용. 기본 자리 수는 3자리

사용법

{{ value | percent [: option] }}

Value는 Number Type을 지원한다

사용 예

{{ 1.23 | percent }} => 123%

{{1.2345678 | percent : '1.1-1'}} => 123.5%

DecimalPipe

value 의 값을 3자리 단위로 , 로 구분해 준다. 소수점 이하는 .로 구분

사용법

```
{{ value | number [: option] }}
```

Value는 Number Type을 지원한다

사용 예

```
{{ 123456789 | number }}           => 123,456,789
```

```
{{ 1234.56789 | number : '1.1' }} => 1,234.568
```

SlicePipe

JavaScript의 substr과 동일한 동작을 수행한다.

사용법

```
{{ value | slice [: option] }}
```

Value는 String Type을 지원한다

사용 예

```
{{ '동해물과 백두산이 마르고 닳도록' | slice }}      => 동해물과 백두산이 마르고 닳도록
```

```
{{ '동해물과 백두산이 마르고 닳도록' | slice : 0:3}}  => 동해물
```

```
{{ '동해물과 백두산이 마르고 닳도록' | slice : -5:-1}} => 고 닳도
```

Pipe

Class 내부에서 Pipe를 사용하기 위해서는 생성자를 이용하여 DI(의존성 주입)을 받는다.
주입 받은 파이프의 transform 함수를 호출하여 값을 변환하여 사용한다.

사용법

```
constructor(private slicePipe: SlicePipe){ }
```

사용 예

```
private getString(): string{  
    return this.slicePipe.transform(this.str, 0:3);  
}
```

Custom Pipe

- Custom Pipe는 PipeTransform 인터페이스를 implemen하여 구현한다. 따라서 Custom Pipe Class 내부에는 PipeTransform 인터페이스의 transform 함수를 구현 해야 한다.
- transform의 매개변수 중 첫 번째 매개변수는 변환 할 데이터 값이다.
- transform의 매개변수 중 두 번째 매개변수부터는 필터 이름 뒤에 오는 option 값이다.

Custom Pipe

{{ 1.2345678 | percent : '1.1-1' }}

```
@Pipe({
  name: 'percent'
})
export class PercentPipe implements PipeTransform{
  public transform( data: string, value:string ) : string{
    return 처리 후 전달 값
  }
}
```

