



# Pipes

## Работа с pipes

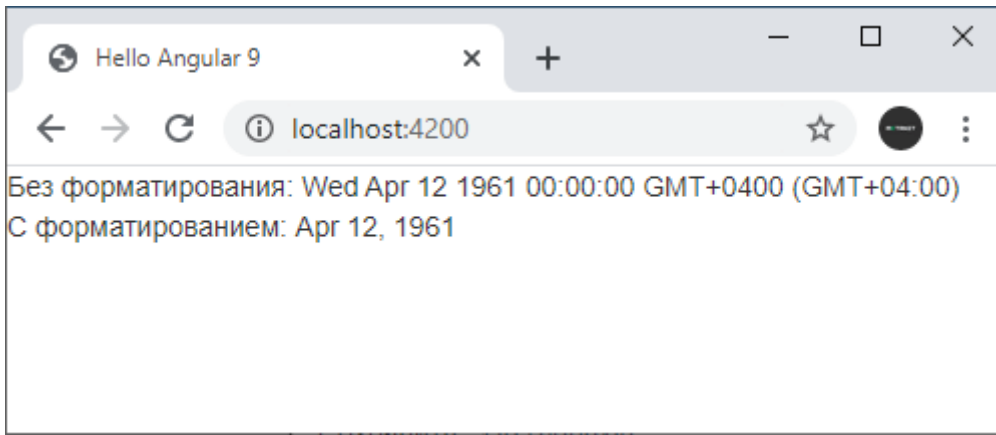
Последнее обновление: 05.06.2022



Pipes представляют специальные инструменты, которые позволяют форматировать отображаемые значения. Например, нам надо вывести определенную дату:

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'my-app',
5   template: `<div>Без форматирования: {{myDate}}</div>
6               <div>С форматированием: {{myDate | date}}</div>`
7 })
8 export class AppComponent {
9
10   myDate = new Date(1961, 3, 12);
11 }
```

Здесь создается дата, которая дважды выводится в шаблоне. Во втором случае к дате применяется форматирование с помощью класса DatePipe.



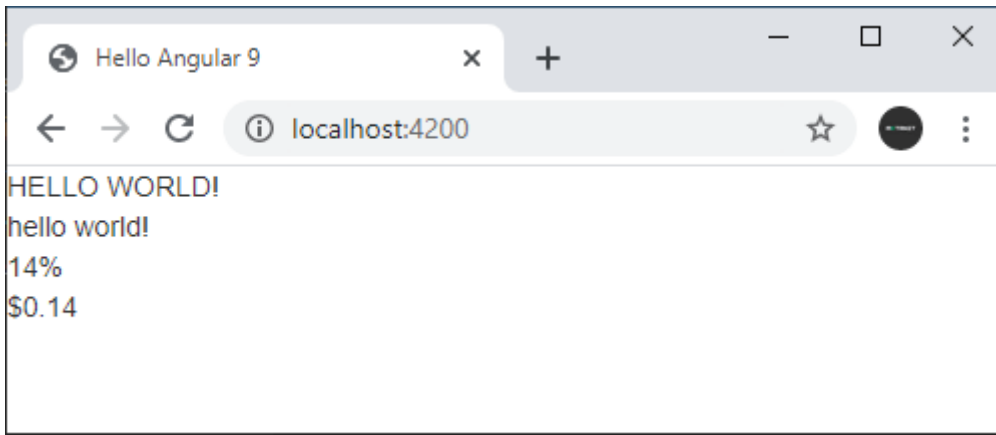
## Встроенные pipes

В Angular есть ряд встроенных pipes. Основные из них:

- **CurrencyPipe**: форматирует валюту
- **PercentPipe**: форматирует проценты
- **UpperCasePipe**: переводит строку в верхний регистр
- **LowerCasePipe**: переводит строку в нижний регистр
- **DatePipe**: форматирует дату
- **DecimalPipe**: задает формат числа
- **SlicePipe**: обрезает строку

При применении классов суффикс Pipe отбрасывается (за исключением DecimalPipe - для его применения используется название "number"):

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'my-app',
5   template: `<div>{{welcome | uppercase}}</div>
6               <div>{{welcome | lowercase}}</div>
7               <div>{{percentage | percent}}</div>
8               <div>{{percentage | currency}}</div>`
9 })
10 export class AppComponent {
11
12   welcome: string = "Hello World!";
13   percentage: number = 0.14;
14 }
```

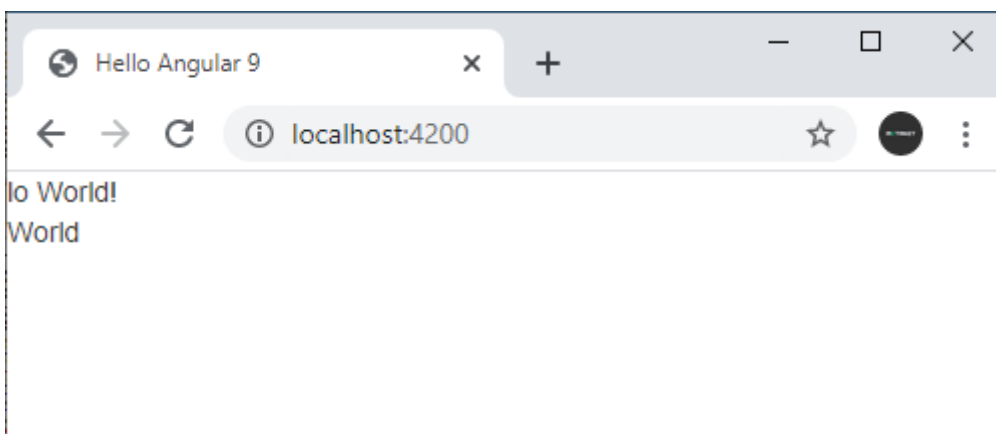


## Параметры в pipes

Pipes могут получать параметры. Например, пайп SlicePipe, который обрезает строку, может получать в качестве параметра, начальный и конечный индексы подстроки, которую надо вырезать:

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'my-app',
5   template: `<div>{{welcome | slice:3}}</div>
6             <div>{{welcome | slice:6:11}}</div>`
7 })
8 export class AppComponent {
9
10   welcome: string = "Hello World!";
11 }
```

Все параметры в пайп передаются через двоеточие. В данном случае `slice:6:11` вырезает подстроку, начиная с 6 до 11 индекса. При этом, если начала выреза строки обязательно передавать, то конечный индекс необязателен. В этом случае в качестве конечного индекса выступает конец строки.



## Форматирование дат

DatePipe в качестве параметра может принимать шаблон даты:

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'my-app',
5   template: `<div>{{myDate | date:"dd/MM/yyyy"}}</div>`
6 })
7 export class AppComponent {
8
9   myDate = Date.now();
10 }
```

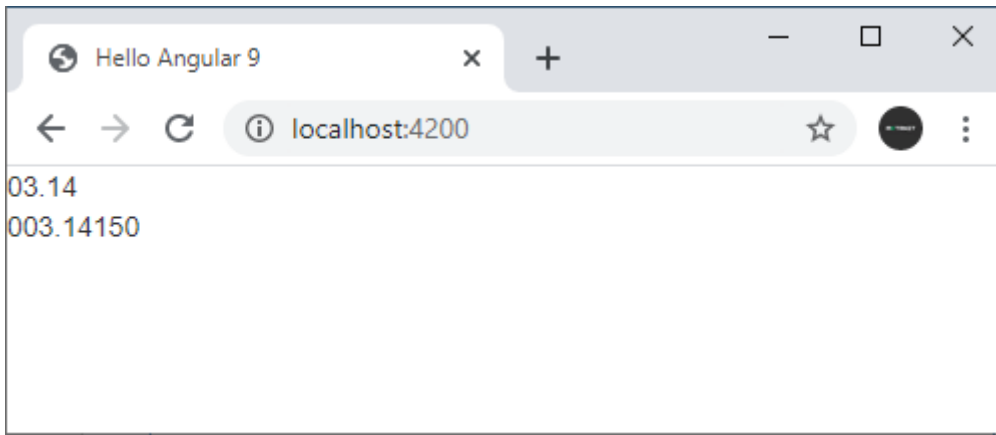
## Форматирование чисел

DecimalPipe в качестве параметра принимает формат числа в виде шаблона:

```
1 {{ value | number [ : digitsInfo [ : locale ] ] }}
```

- value: само выводимое значение
- digitsInfo: строка в формате "minIntegerDigits.minFractionDigits-maxFractionDigits", где
  - minIntegerDigits - минимальное количество цифр в целой части
  - minFractionDigits - минимальное количество цифр в дробной части
  - maxFractionDigits - максимальное количество цифр в дробной части
- locale: код применяемой культуры

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'my-app',
5   template: `<div>{{pi | number:'2.1-2'}}</div>
6   <div>{{pi | number:'3.5-5'}}</div>`
7 })
8 export class AppComponent {
9
10   pi: number = 3.1415;
11 }
```



## Форматирование валюты

CurrencyPipe может принимать ряд параметров:

```
1 {{ value | currency [ : currencyCode [ : display [ : digitsInfo [ : locale ]
```

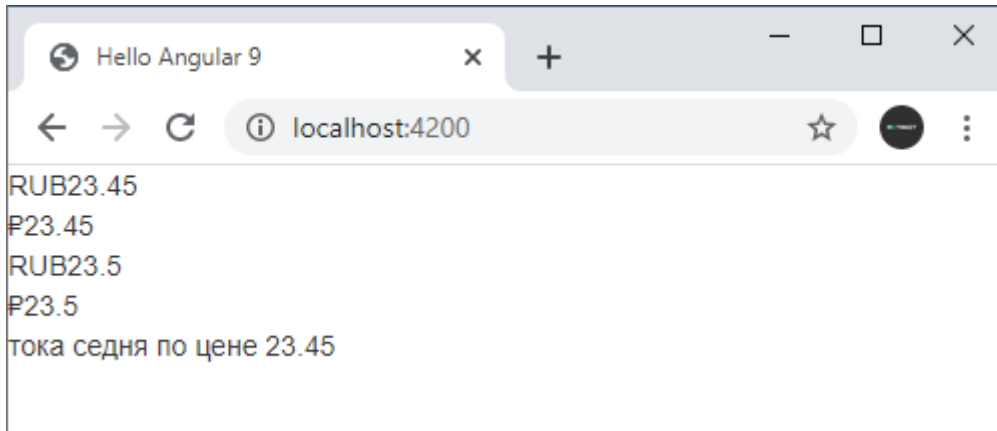
- **value:** выводимая сумма
- **currencyCode:** код валюты согласно спецификации ISO 4217. Если не указан, то по умолчанию применяется USD
- **display:** указывает, как отображать символ валюты. Может принимать следующие значения:
  - **code:** отображает код валюты (например, USD)
  - **symbol** (значение по умолчанию): отображает символ валюты (например, \$)
  - **symbol-narrow:** некоторые страны используют в качестве символа валюты несколько символов, например, канадский доллар - CA\$, данный параметр позволяет получить собственно символ валюты - \$
  - **string:** отображает произвольную строку
- **digitsInfo:** формат числа, который применяется в DecimalPipe
- **locale:** код используемой локали

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'my-app',
5   template: `
6     <div>{{money | currency:'RUB':'code'}}</div>
7     <div>{{money | currency:'RUB':'symbol-narrow'}}</div>
8     <div>{{money | currency:'RUB':'symbol':'1.1-1'}}</div>
```

```

9      <div>{{money | currency:'RUB':'symbol-narrow':'1.1-1'}}</div>
10     <div>{{money | currency:'RUB':'тока седня по цене '}}</div>`
11  })
12  export class AppComponent {
13      money: number = 23.45;
14  }

```



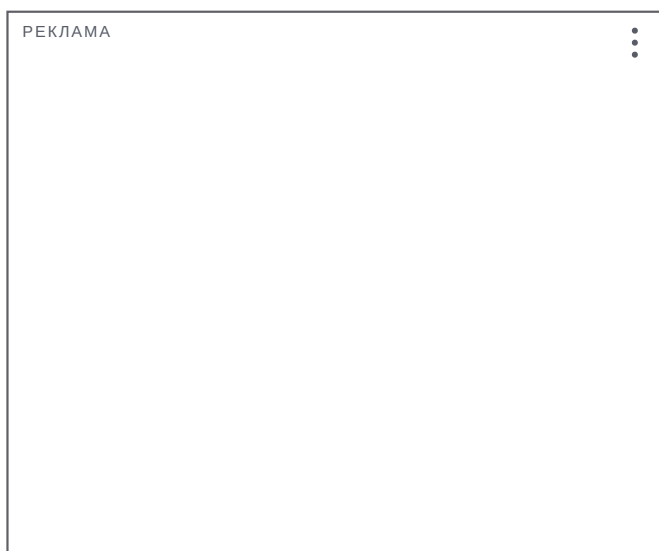
## Цепочки pipes

Вполне возможно, что мы захотим применить сразу несколько pipes к одному значению, тогда мы можем составлять цепочки выражений, разделенные вертикальной чертой:

```

1  import { Component } from '@angular/core';
2
3  @Component({
4      selector: 'my-app',
5      template: `<div>{{message | slice:6:11 | uppercase}}</div>`
6  })
7  export class AppComponent {
8
9      message = "Hello World!";
10 }

```



[Назад](#) [Содержание](#) [Вперед](#)

practicum.yandex.ru Реклама

## Курс: Математический анализ данных

За 4 месяца вы изучите принципы мат. анализа, линей

### ALSO ON METANIT.COM

#### Введение. Первый проект

3 месяца назад • 4 comments

Основы платформы Razor Pages в ASP.NET Core в связке с языком ...

#### Создание правил URL Rewriting

2 месяца назад • 1 comment

Создание правил URL Rewriting в приложении на ASP.NET Core и C#, ...

#### Введение в MSIL. Ildasm и ...

месяц назад • 3 comments

Введение в MSIL, компиляция кода на языке программирования C# в ...

#### Введение в MSIL. Ildasm и ...

3 месяца назад • 1 comment

Введение в MSIL, компиляция кода на языке программирования C# в ...

6 Comments

metanit.com

Политика конфиденциальности Disqus

Войти ▾

Favorite

Твитнуть

Поделиться

Лучшее ▾



Присоединиться к обсуждению...

ВОЙТИ С ПОМОЩЬЮ

ИЛИ ЧЕРЕЗ DISQUS

Имя



**Dmitry Babik** • 4 года назад

Если число отформатировать как валюту - сортироваться в табличке оно по прежнему будет как число или уже как строка?

1 ^ | v • Ответить • Поделиться >



**Михаил Макаров** • 2 года назад • edited

Жалко, что они позволяют только пропускать через себя значения и как-то их модифицировать, но не позволяют фильтровать их. Порой хочется вывести в цикле `ngFor` только те значения, которые удовлетворяют определённому критерию, и было бы здорово, если бы можно было их отобрать в синтаксисе `pipes`.

^ | v • Ответить • Поделиться >



**Роман Кириев** ➔ Михаил Макаров • год назад • edited

Pipe не предназначен для работы с `ngFor`. Да и не логично это. Отфильтровать можно и нужно до того как передаешь в шаблон.

^ | v • Ответить • Поделиться >



**Sergey Blokhov** • 3 года назад • edited

В разделе Встроенные `pipes` ещё один пайп можно добавить: `KeyValuePipe`. С использованием этого пайпа объект или `Map` могут быть итерированы в `ngFor`.  
<https://angular.io/api/comm...>

```
@Component({
  selector: 'keyvalue-pipe',
  template: `< span>
< p>Object< /p>
<div *ngfor="let item of object | keyvalue">
  {{item.key}}:{{item.value}}
< /div>
< p>Map< /p>
<div *ngfor="let item of map | keyvalue">
  {{item.key}}:{{item.value}}
< /div>
< /span>`
})
export class KeyValuePipeComponent {
  object: {[key: number]: string} = {2: 'foo', 1: 'bar'};
  map = new Map([[2, 'foo'], [1, 'bar']]);
}
```

^ | v • Ответить • Поделиться >



**Андрей Телешев** • 3 года назад • edited

Пайпы несколько шире в применении, теперь `RxJs` операторы вызываются внутри `pipe`.

^ | v • Ответить • Поделиться >



**Александр Семенов** • 5 лет назад • edited

Касаемо `DecimalPipe` дополню, что если не указана длина левой части - можно использовать следующий формат:  
`{{num | number:'.1-2'}}`


При 100 выведет 100,0

При 1 выведет 1,0

^ | v • Ответить • Поделиться >



РЕКЛАМА



practicum.yandex.ru

Помощь са

YooMoney: 410011174

Перевод н

Номер кар 404841502

Номер кар 489049475

Курс:

Математический анализ данных в Практикуме с нуля!

За 4 месяца вы изучите принципы мат. анализа, линейной алгебры и статистики.

Вконтакте

Узнать больше

Канал сайта на youtube | Помощь сайту

mail.ru