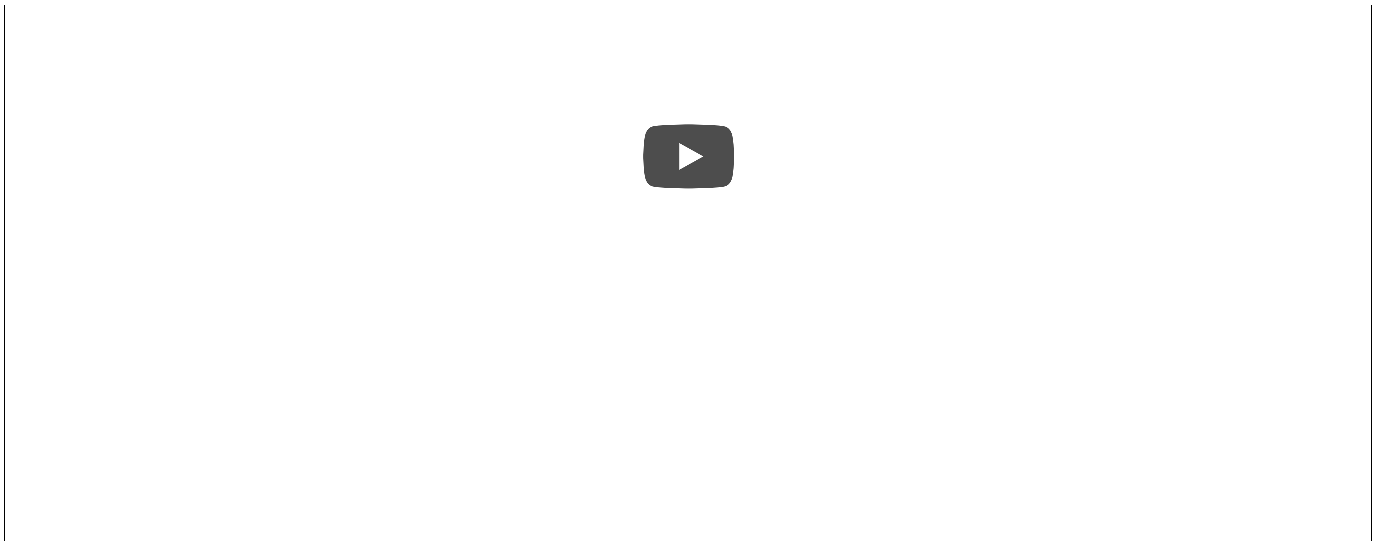
Built-In Pipes

EP 8.2 - Angular / Pipes / Built-In Pipes



In this video I'm using an online editor called Plunker (https://plnkr.co/) to write and run Angular code. The book and code has since been updated to use StackBlitz (https://stackblitz.com) instead. To understand more about why and the differences between read this (/courses/angular/quickstart/overview/#_plunker_vs_stackblitz).

(https://github.com/codecraft-tv/angular-course/tree/current/8.pipes/2.built-in-pipes/code)(https://stackblitz.com/github/codecraft-tv/angular-course/

OVERVIEW (/COURSES/ANGULAR/PIPES/OVERVIEW/) ASYNC PIPE

TABLE OF CONTENT

In this lecture we will cover all of the built-in pipes provided by Angular apart from the *async pipe* which we will cover in detail in a later lecture.

Learning Objectives

- Know the different built-in pipes provided by Angular and how to use them.

Pipes Provided by Angular

Angular provides the following set of built-in pipes.

CurrencyPipe

This pipe is used for formatting currencies. Its first argument is an abbreviation of the currency type (e.g. "EUR", "USD", and so on), like so:

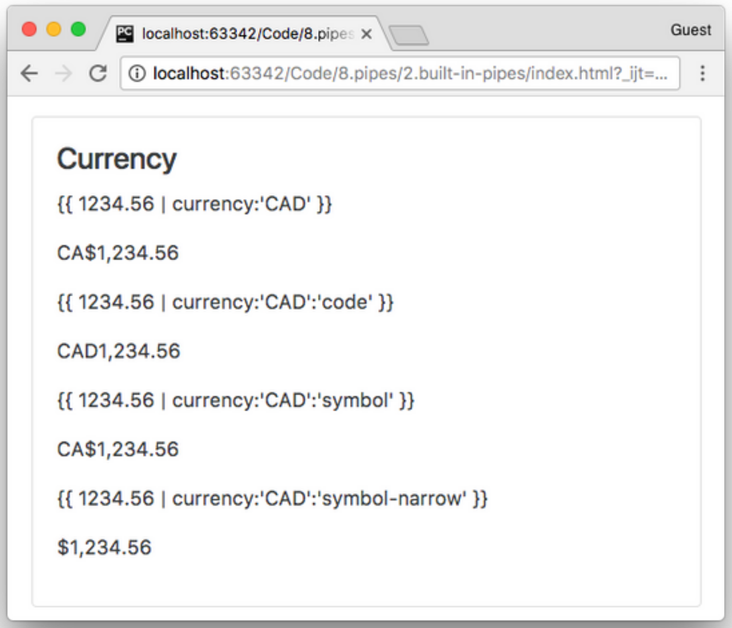
```
{{ 1234.56 | currency:'CAD' }}
```

The above prints out `CA$1,234.56`.

If instead of the abbreviation of `CA$` we want the currency symbol to be printed out we pass as a second parameter the string `symbol-narrow`, like so:

```
{{ 1234.56 | currency:"CAD":"symbol-narrow" }}
```

The above prints out `$1,234.56`.


localhost:63342/Code/8.pipes
Guest
localhost:63342/Code/8.pipes/2.built-in-pipes/index.html?_ijt=...

Currency

{{ 1234.56 | currency:'CAD' }}

CA$1,234.56

{{ 1234.56 | currency:'CAD':'code' }}

CAD1,234.56

{{ 1234.56 | currency:'CAD':'symbol' }}

CA$1,234.56

{{ 1234.56 | currency:'CAD':'symbol-narrow' }}

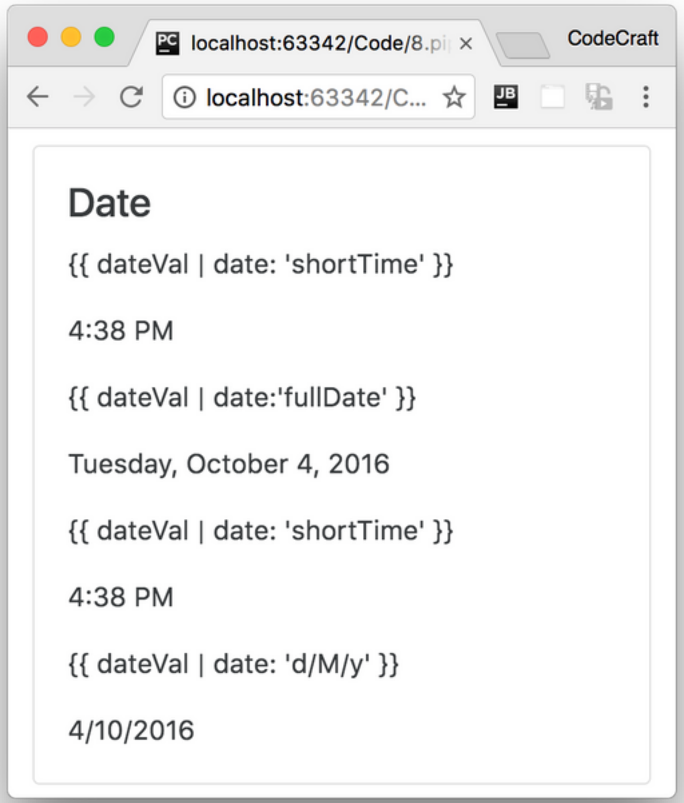
$1,234.56

HTML

```
<div class="card card-block">
  <h4 class="card-title">Currency</h4>
  <div class="card-text">
    <p ngNonBindable>{{ 1234.56 | currency:'CAD' }}</p>
    <p>{{ 1234.56 | currency:"CAD" }}</p>

    <p ngNonBindable>{{ 1234.56 | currency:'CAD':'code' }}</p>
    <p>{{ 1234.56 | currency:'CAD':'code'}}</p>

    <p ngNonBindable>{{ 1234.56 | currency:'CAD':'symbol' }}</p>
    <p>{{ 1234.56 | currency:'CAD':'symbol'}}</p>

    <p ngNonBindable>{{ 1234.56 | currency:'CAD':'symbol-narrow' }}</p>
    <p>{{ 1234.56 | currency:'CAD':'symbol-narrow'}}</p>
  </div>
</div>
```

DatePipe

This pipe is used for the transformation of dates. The first argument is a format string, like so:



Date

{{ dateVal | date: 'shortTime' }}

4:38 PM

{{ dateVal | date:'fullDate' }}

Tuesday, October 4, 2016

{{ dateVal | date: 'shortTime' }}

4:38 PM

{{ dateVal | date: 'd/M/y' }}

4/10/2016

HTML

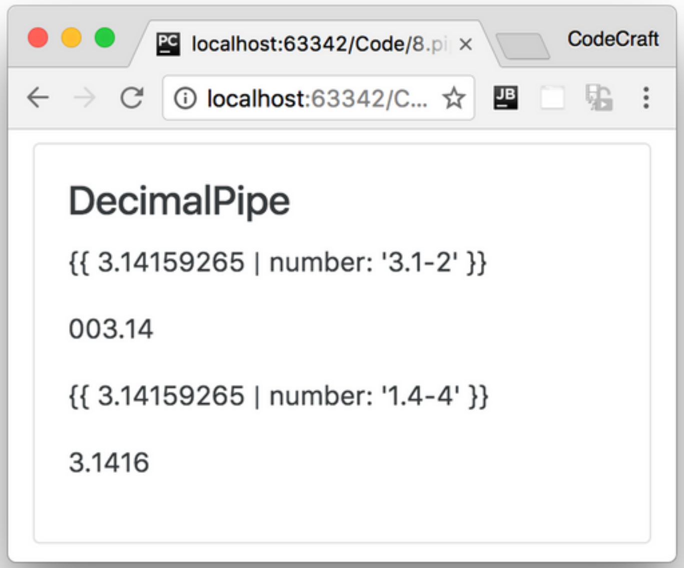
```
<div class="card card-block">
  <h4 class="card-title">Date</h4>
  <div class="card-text">
    <p ngNonBindable>{{ dateVal | date: 'shortTime' }}</p> (1)
    <p>{{ dateVal | date: 'shortTime' }}</p>

    <p ngNonBindable>{{ dateVal | date:'fullDate' }}</p>
    <p>{{ dateVal | date: 'fullDate' }}</p>

    <p ngNonBindable>{{ dateVal | date: 'd/M/y' }}</p>
    <p>{{ dateVal | date: 'd/M/y' }}</p>
  </div>
</div>
```

1 `dateVal` is an instance of `new Date()`.

DecimalPipe

This pipe is used for transformation of decimal numbers.

The first argument is a format string of the form "{minIntegerDigits}.{minFractionDigits}-{maxFractionDigits}", like so:



DecimalPipe

{{ 3.14159265 | number: '3.1-2' }}

003.14

{{ 3.14159265 | number: '1.4-4' }}

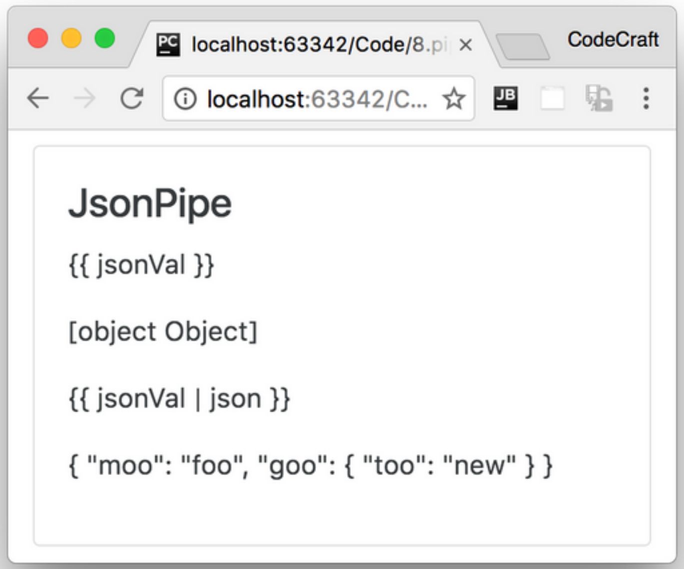
3.1416

HTML

```
<div class="card card-block">
  <div class="card-text">
    <h4 class="card-title">DecimalPipe</h4>
    <p ngNonBindable>{{ 3.14159265 | number: '3.1-2' }}</p>
    <p>{{ 3.14159265 | number: '3.1-2' }}</p>

    <p ngNonBindable>{{ 3.14159265 | number: '1.4-4' }}</p>
    <p>{{ 3.14159265 | number: '1.4-4' }}</p>
  </div>
</div>
```

JsonPipe

This transforms a JavaScript object into a JSON string, like so:



JsonPipe

{{ jsonVal }}

[object Object]

{{ jsonVal | json }}

{ "moo": "foo", "goo": { "too": "new" } }

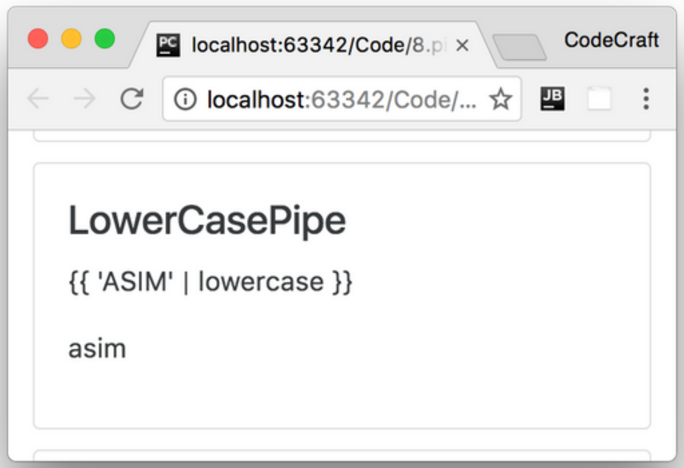
HTML

```html
<div class="card card-block">
  <h4 class="card-title">JsonPipe</h4>
  <div class="card-text">
    <p ngNonBindable>{{ jsonVal }}</p> (1)
    <p>{{ jsonVal }}</p>

    <p ngNonBindable>{{ jsonVal | json }}</p>
    <p>{{ jsonVal | json }}</p>
  </div>
</div>
```

1 `jsonVal` is an object declared as `{ moo: 'foo', goo: { too: 'new' }}`.

LowerCasePipe

This transforms a string to lowercase, like so:



LowerCasePipe

{{ 'ASIM' | lowercase }}

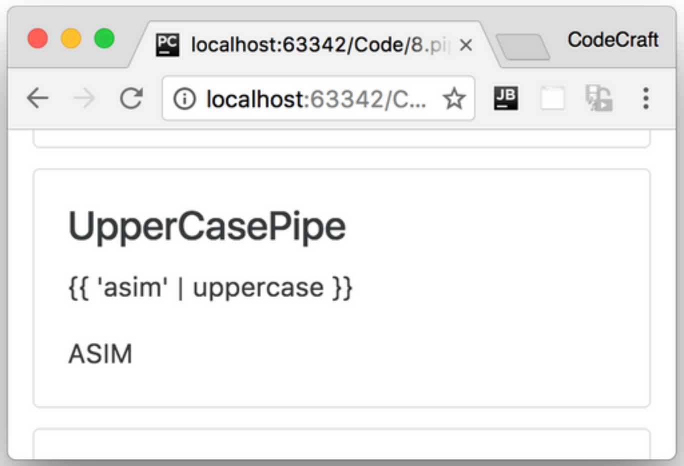
asim

HTML

```
<div class="card card-block">
  <h4 class="card-title">LowerCasePipe</h4>
  <div class="card-text">
    <p ngNonBindable>{{ 'ASIM' | lowercase }}</p>
    <p>{{ 'ASIM' | lowercase }}</p>
  </div>
</div>
```

UpperCasePipe

This transforms a string to uppercase, like so:



UpperCasePipe

{{ 'asim' | uppercase }}

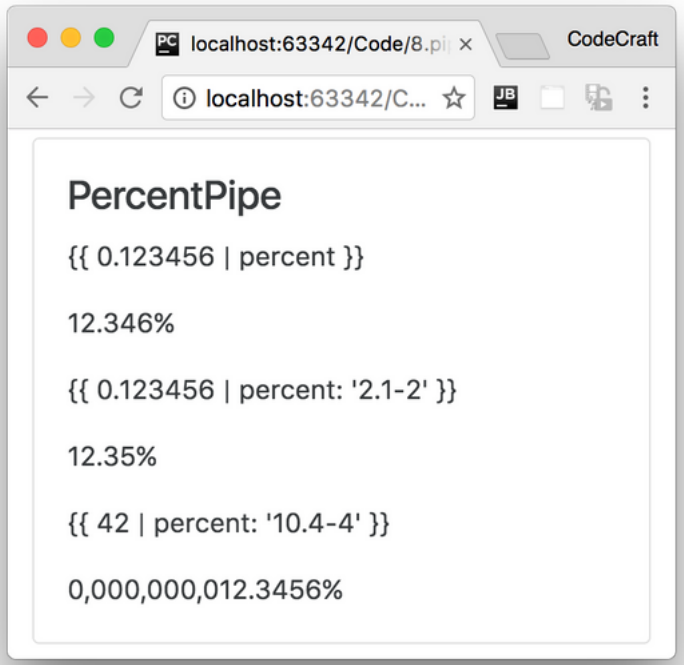
ASIM

HTML

```
<div class="card card-block">
  <h4 class="card-title">UpperCasePipe</h4>
  <div class="card-text">
    <p ngNonBindable>{{ 'asim' | uppercase }}</p>
    <p>{{ 'asim' | uppercase }}</p>
  </div>
</div>
```

PercentPipe

Formats a number as a percent, like so:



PercentPipe

{{ 0.123456 | percent }}

12.346%

{{ 0.123456 | percent: '2.1-2' }}

12.35%

{{ 42 | percent: '10.4-4' }}

0,000,000,012.3456%

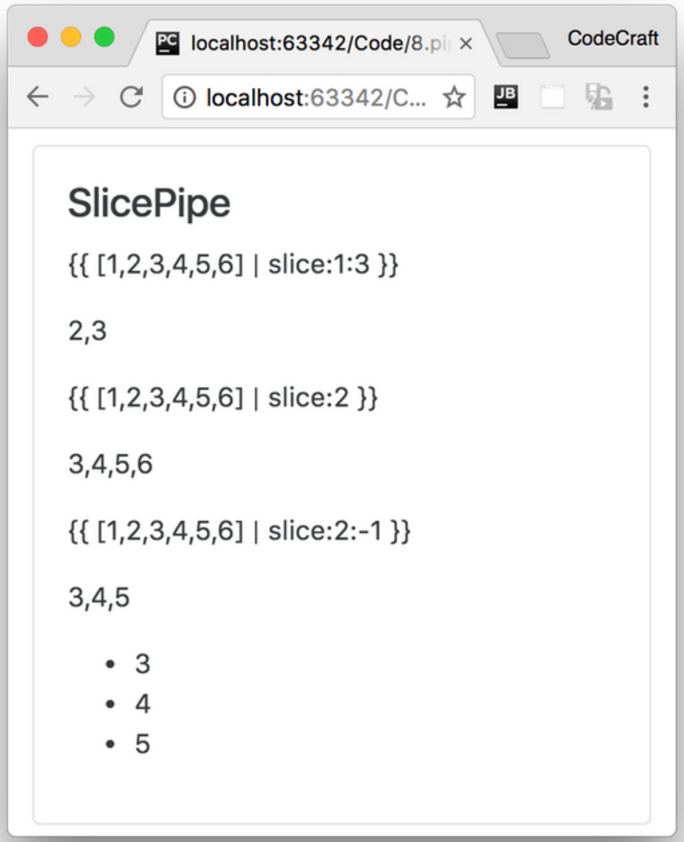
HTML
```
<div class="card card-block">
  <h4 class="card-title">PercentPipe</h4>
  <div class="card-text">
    <p ngNonBindable>{{ 0.123456 | percent }}</p>
    <p>{{ 0.123456 | percent }}</p>

    <p ngNonBindable>{{ 0.123456 | percent: '2.1-2' }}</p> (1)
    <p>{{ 0.123456 | percent: '2.1-2' }}</p>

    <p ngNonBindable>{{ 42 | percent: '10.4-4' }}</p>
    <p>{{ 0.123456 | percent : "10.4-4" }}</p>
  </div>
</div>
```

1 Percent can be passed a format string similar to the format passed to the `DecimalPipe`.

SlicePipe

This returns a *slice* of an array. The first argument is the start index of the slice and the second argument is the end index.

If either indexes are not provided it assumes the start or the end of the array and we can use negative indexes to indicate an offset from the end, like so:



SlicePipe

{{ [1,2,3,4,5,6] | slice:1:3 }}

2,3

{{ [1,2,3,4,5,6] | slice:2 }}

3,4,5,6

{{ [1,2,3,4,5,6] | slice:2:-1 }}

3,4,5

- 3
- 4
- 5

HTML

```
<div class="card card-block">
  <h4 class="card-title">SlicePipe</h4>
  <div class="card-text">
    <p ngNonBindable>{{ [1,2,3,4,5,6] | slice:1:3 }}</p> (1)
    <p>{{ [1,2,3,4,5,6] | slice:1:3 }}</p>

    <p ngNonBindable>{{ [1,2,3,4,5,6] | slice:2 }}</p> (2)
    <p>{{ [1,2,3,4,5,6] | slice:2 }}</p>

    <p ngNonBindable>{{ [1,2,3,4,5,6] | slice:2:-1 }}</p> (3)
    <p>{{ [1,2,3,4,5,6] | slice:2:-1 }}</p>

    <pre ngNonBindable>
&lt;ul&gt;
  &lt;li *ngFor=&quot;let v of [1,2,3,4,5,6] | slice:2:-1&quot;&gt;
    {{v}}
  &lt;/li&gt;
&lt;/ul&gt;
    </pre>

    <ul>
      <li *ngFor="let v of [1,2,3,4,5,6] | slice:2:-1"> (4)
        {{v}}
      </li>
    </ul>
  </div>
</div>
```

1 `slice:1:3` means return the items from the 1st to the 3rd index inclusive (indexes start at 0).

2 `slice:2` means return the items from the 2nd index to the end of the array.

3 `slice:2:-1` means return the items from the 2nd index to one from the end of the array.

4 We can use slice inside for loops to only loop over a subset of the array items.

AsyncPipe

This pipe accepts an observable or a promise and lets us render the output of an observable or promise without having to call `then` or `subscribe`.

We are going to take a much deeper look at this pipe at the end of this section.

Summary

Pipes enables you to easily transform data for display purposes in templates.

Angular comes with a very useful set of pre-built pipes to handle most of the common transformations.

One of the more complex pipes to understand in Angular is the async pipe that's what we'll cover next.

Listing

Listing 1. main.ts

TypeScript

```
import {NgModule, Component} from '@angular/core';
import {BrowserModule} from '@angular/platform-browser';
import {platformBrowserDynamic} from '@angular/platform-browser-dynamic';

@Component({
  selector: 'pipe-builtins',
  template: `<div class="card card-block">
  <h4 class="card-title">Currency</h4>
  <div class="card-text">
    <p ngNonBindable>{{ 1234.56 | currency:'CAD' }}</p>
    <p>{{ 1234.56 | currency:"CAD" }}</p>

    <p ngNonBindable>{{ 1234.56 | currency:'CAD':'code' }}</p>
    <p>{{ 1234.56 | currency:'CAD':'code'}}</p>

    <p ngNonBindable>{{ 1234.56 | currency:'CAD':'symbol' }}</p>
    <p>{{ 1234.56 | currency:'CAD':'symbol'}}</p>

    <p ngNonBindable>{{ 1234.56 | currency:'CAD':'symbol-narrow' }}</p>
    <p>{{ 1234.56 | currency:'CAD':'symbol-narrow'}}</p>
  </div>
</div>

<div class="card card-block">
  <h4 class="card-title">Date</h4>
  <div class="card-text">
    <p ngNonBindable>{{ dateVal | date: 'shortTime' }}</p>
    <p>{{ dateVal | date: 'shortTime' }}</p>

    <p ngNonBindable>{{ dateVal | date:'fullDate' }}</p>
    <p>{{ dateVal | date: 'fullDate' }}</p>

    <p ngNonBindable>{{ dateVal | date: 'shortTime' }}</p>
    <p>{{ dateVal | date: 'shortTime' }}</p>

    <p ngNonBindable>{{ dateVal | date: 'd/M/y' }}</p>
    <p>{{ dateVal | date: 'd/M/y' }}</p>
  </div>
</div>

<div class="card card-block">
  <div class="card-text">
    <h4 class="card-title">DecimalPipe</h4>
    <p ngNonBindable>{{ 3.14159265 | number: '3.1-2' }}</p>
    <p>{{ 3.14159265 | number: '3.1-2' }}</p>

    <p ngNonBindable>{{ 3.14159265 | number: '1.4-4' }}</p>
    <p>{{ 3.14159265 | number: '1.4-4' }}</p>
  </div>
</div>

<div class="card card-block">
  <h4 class="card-title">JsonPipe</h4>
  <div class="card-text">
    <p ngNonBindable>{{ jsonVal }}</p>
    <p>{{ jsonVal }}</p>

    <p ngNonBindable>{{ jsonVal | json }}</p>
    <pre>{{ jsonVal | json }}</pre>
  </div>
</div>


<div class="card card-block">
  <h4 class="card-title">LowerCasePipe</h4>
  <div class="card-text">
    <p ngNonBindable>{{ 'ASIM' | lowercase }}</p>
    <p>{{ 'ASIM' | lowercase }}</p>
  </div>
</div>
```

```
<div class="card card-block">
  <h4 class="card-title">UpperCasePipe</h4>
  <div class="card-text">
    <p ngNonBindable>{{ 'asim' | uppercase }}</p>
    <p>{{ 'asim' | uppercase }}</p>
  </div>
</div>

<div class="card card-block">
  <h4 class="card-title">PercentPipe</h4>
  <div class="card-text">
    <p ngNonBindable>{{ 0.123456 | percent }}</p>
    <p>{{ 0.123456 | percent }}</p>

    <p ngNonBindable>{{ 0.123456 | percent: '2.1-2' }}</p>
    <p>{{ 0.123456 | percent: '2.1-2' }}</p>

    <p ngNonBindable>{{ 42 | percent: '10.4-4' }}</p>
    <p>{{ 0.123456 | percent : "10.4-4" }}</p>
  </div>
</div>

<div class="card card-block">
  <h4 class="card-title">SlicePipe</h4>
  <div class="card-text">
    <p ngNonBindable>{{ [1,2,3,4,5,6] | slice:1:3 }}</p>
    <p>{{ [1,2,3,4,5,6] | slice:1:3 }}</p>

    <p ngNonBindable>{{ [1,2,3,4,5,6] | slice:2 }}</p>
    <p>{{ [1,2,3,4,5,6] | slice:2 }}</p>

    <p ngNonBindable>{{ [1,2,3,4,5,6] | slice:2:-1 }}</p>
    <p>{{ [1,2,3,4,5,6] | slice:2:-1 }}</p>

    <pre ngNonBindable>
&lt;ul&gt;
  &lt;li *ngFor=&quot;let v of [1,2,3,4,5,6] | slice:2:-1&quot;&gt;
    {{v}}
  &lt;/li&gt;
&lt;/ul&gt;
    </pre>

    <ul>
      <li *ngFor="let v of [1,2,3,4,5,6] | slice:2:-1">
        {{v}}
      </li>
    </ul>
  </div>
</div>

 `
})
class PipeBuiltinsComponent {
  private dateVal: Date = new Date();
  private jsonVal: Object = {moo: 'foo', goo: {too: 'new'}};

}

@Component({
  selector: 'app',
  template: `
<pipe-builtins></pipe-builtins>
 `
})
class AppComponent {
}


@NgModule({
  imports: [BrowserModule],
  declarations: [AppComponent,
    PipeBuiltinsComponent
  ],
  bootstrap: [AppComponent],
})
class AppModule {

}

platformBrowserDynamic().bootstrapModule(AppModule);
```

Caught a mistake or want to contribute to the book? Edit this page on GitHub! (https://github.com/codecraft-tv/angular-course/tree/current/8.pipes/2.built-in-pipes/index.adoc)

‹ OVERVIEW (/COURSES/ANGULAR/PIPES/OVERVIEW/) ASYNC PIPE ›

Home (/) Terms (/terms/) Privacy (/privacy/)