Pipe Parameters

In this lecture, we will learn how to **Customize the output of a Pipe through parameters**. At the end of the day, **pipes are functions**.

Pipes can have arguments to modify the behaviour or output.

https://v17.angular.io/api/common/DatePipe#custom-format-options

https://angular.dev/api/common/DatePipe

Not all pipes are configurable.

For example, the **titlecase pipe** does not have parameters or have to explore a **different pipe.**

For this demonstration, we are going to learn about a new **pipe** called **date.**

The **date pipe** will **format a valid date object.** This **pipe** can be useful for rendering friendlier dates to users. It gives us complete control over the formatting of a date.

Unlike the **titlecase pipe,** the **date pipe has a parameter for modifying the date format.**

To get started, let's create a date object. Open the app component class file, we will create a property called **currentDate**, its value will be a new instance of a **date object**.

The **date pipe** can work with **three types of values**, we can use **numbers, strings, and date objects**.

I think working with a **date object** will be the easiest.

By default, new instances of the **date object** will the **current date and time.** We can avoid trying to guess the **current date** wit a **string or number**. In addition, it's clearly impossible for **date object** to store a value other than a **date**. Whereas a **string** may have a **typo**. If we attempt to pass an **invalid value** to the **date pipe**, we may encounter **unexpected behaviour.**

In most cases, I recommend applying **date pipes to date objects for the most accurate results**.

Let's switch over to the app template file.

At the bottom of the template, we will add a paragraph tagged with the following expression:
{{ currentDate }}

app.html M

basics > src > app > app.html > ...

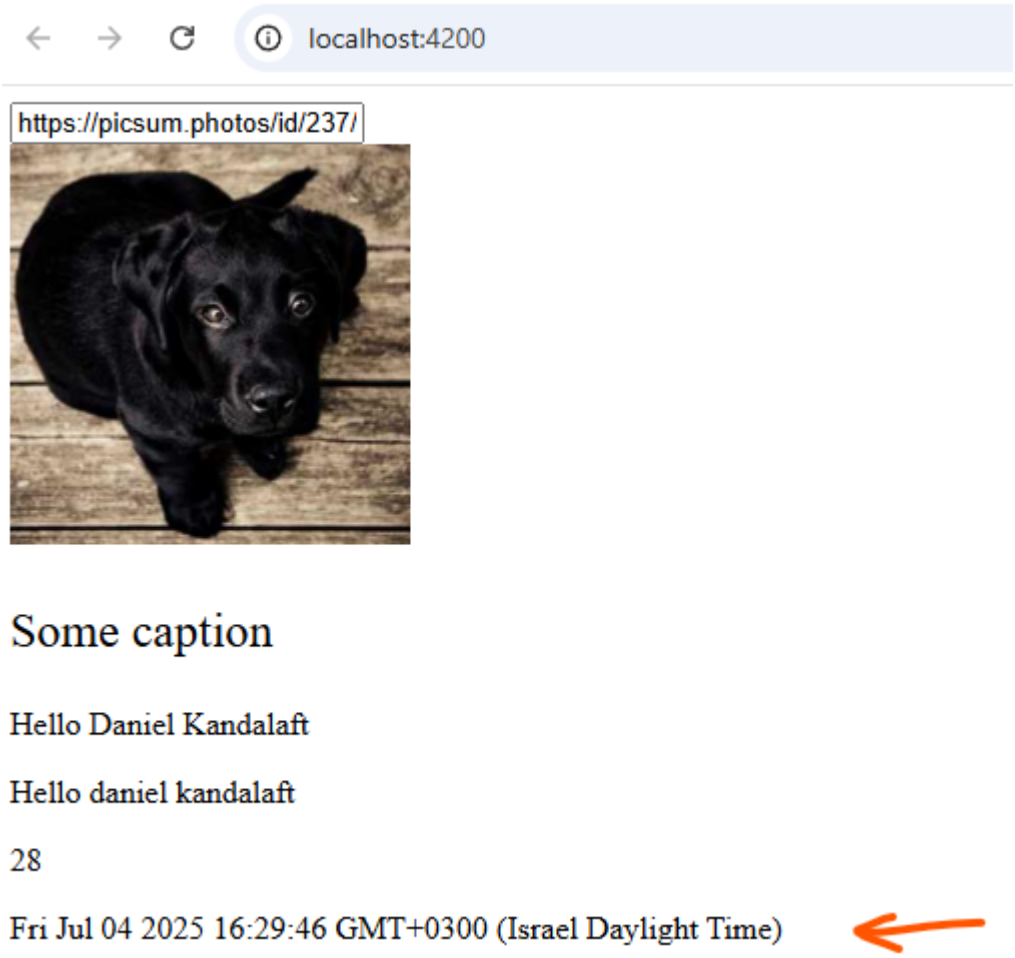
Go to component

```
<input (keyup)="changeImage($event)" [value]="imgURL" />

<app-post [img]="imgURL" (imgSelected)="logImg($event)">
  <p>Some caption</p>
</app-post>

<p>Hello {{ name | titlecase }}</p>
<p>Hello {{ getName() }}</p>
<p>{{ 15 + 13 }}</p>
<p>{{ currentDate }}</p>
```

Let's switch over to the browser:



Date objects can be rendered in a template. However, the formatting isn't appealing. Ideally, the format of a date should be the month, day and year.

At the same time, we shouldn't modify the **date object**. In some cases, you may want to work with **date objects for sorting.** JavaScript allows for arrays to be sortable with dates.

In this scenario, applying the **date pipe** will allow us to format the date without directly modifying the **date object**.

Let's fix the formatting with the **date pipe.**

In app template, we will add the **date pipe** after the **currentDate expression.** By default, the **date pipe** will format the date by displaying the **month, day and year.**

** {{ currentDate | date }} **

Everything else will be removed. It's possible, you may see a different format based on your current environment, or when you're taking this course. For a consistent format, I recommend passing in a format.

The **date pipes first parameter is the format for the date, we can add a parameter by adding a colon : after the name of the pipe, followed by the value. A parameter for a pipe is the equivalent of passing in a value to a function's argument list.**

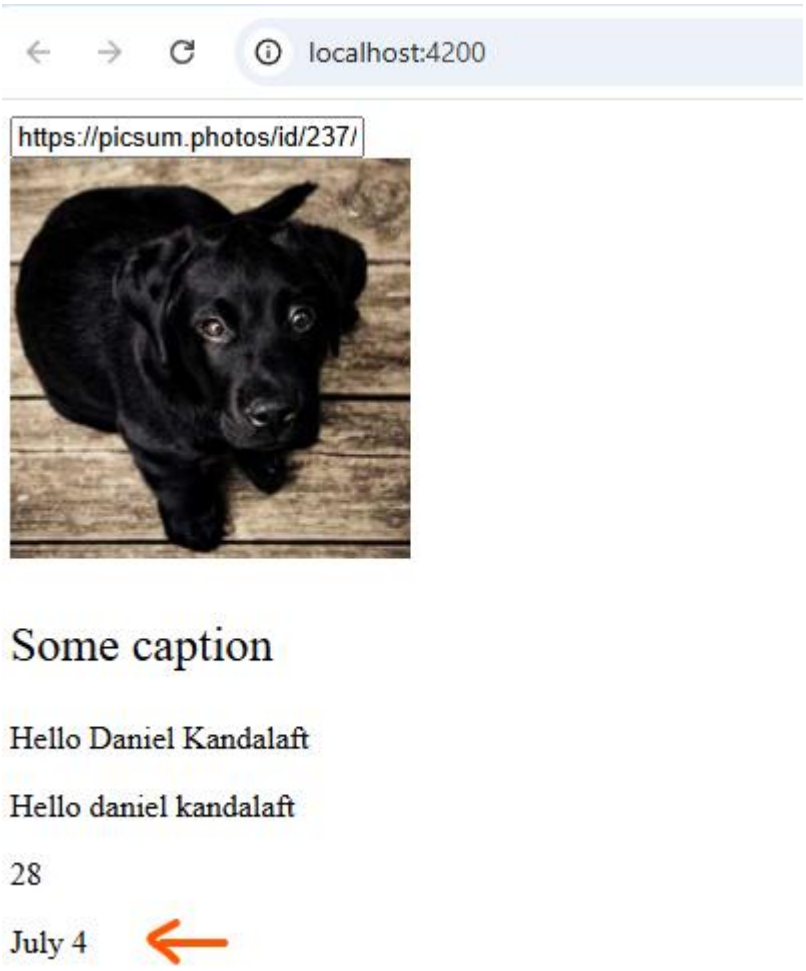
Let's say we want to output the month and day. According to the table below [Custom format options], we have five formats for displaying the month. Let's use the **MMMM** placeholder. As for the day, we will stick with the **d** placeholder.

We will pass in the placeholders:

** {{ currentDate | date: 'MMMM d'}} **

The value of the placeholders must be passed in as a string. Be sure to wrap the values with quotes. As for the arrangement, the month should appear before the day. If you'd like, you can reverse the arrangement for a different format. Feel free to explore the various formats at your disposal.

Switch over to the browser:



The date has been properly formatted, according to our specifications. Choosing a format is a powerful feature, but there is an alternative solution for those who prefer to let Angular format dates. The value of the format does not need to be a series of placeholders. Alternatively, Angular provides a set of placeholder's formats.

Instead of passing in a format, we can pass in a predefined format. Angular will format the date based on the rules in the equivalent column [See below under pre-defined format options]. The Angular team provide these formats for the most popular date formats in the community. You may prefer to use predefined formats as a shortcut. It'll save you time from scrolling around the documentation for outputting the desired format.

For example, we can use the **short** option instead of typing the same format with placeholders. Angular will be able to format the date regardless of the absence of placeholders. According to this table, the format renders the **month, day, year and time.**

Custom format options

You can construct a format string using symbols to specify the components of a date-time value, as described in the following table. Format details depend on the locale. Fields marked with (*) are only available in the extra data set for the given locale.

Field type	Format	Description	Example Value
Era	G, GG & GGG	Abbreviated	AD
	GGGG	Wide	Anno Domini
	GGGGG	Narrow	A
Year	y	Numeric: minimum digits	2, 20, 201, 2017, 20173
	yy	Numeric: 2 digits + zero padded	02, 20, 01, 17, 73
	yyy	Numeric: 3 digits + zero padded	002, 020, 201, 2017, 20173
	yyyy	Numeric: 4 digits or more + zero padded	0002, 0020, 0201, 2017, 20173
ISO Week-numbering year	Y	Numeric: minimum digits	2, 20, 201, 2017, 20173
	YY	Numeric: 2 digits + zero padded	02, 20, 01, 17, 73
	YYY	Numeric: 3 digits + zero padded	002, 020, 201, 2017, 20173
	YYYY	Numeric: 4 digits or more + zero padded	0002, 0020, 0201, 2017, 20173
Month	M	Numeric: 1 digit	9, 12
	MM	Numeric: 2 digits + zero padded	09, 12
	MMM	Abbreviated	Sep
	MMMM	Wide	September
	MMMMM	Narrow	S
Month standalone	L	Numeric: 1 digit	9, 12
	LL	Numeric: 2 digits + zero padded	09, 12
	LLL	Abbreviated	Sep
	LLLL	Wide	September
	LLLLL	Narrow	S
ISO Week of year	w	Numeric: minimum digits	1... 53
	ww	Numeric: 2 digits + zero padded	01... 53
Week of month	W	Numeric: 1 digit	1... 5
Day of month	d	Numeric: minimum digits	1
	dd	Numeric: 2 digits + zero padded	01
Week day	E, EE & EEE	Abbreviated	Tue
	EEEE	Wide	Tuesday
	EEEEE	Narrow	T
	EEEEEE	Short	Tu
Week day standalone	c, cc	Numeric: 1 digit	2

Field type	Format	Description	Example Value
	ccc	Abbreviated	Tue
	cccc	Wide	Tuesday
	ccccc	Narrow	T
	cccccc	Short	Tu
Period	a, aa & aaa	Abbreviated	am/pm or AM/PM
	aaaa	Wide (fallback to a when missing)	ante meridiem/post meridiem
	aaaaa	Narrow	a/p
Period*	B, BB & BBB	Abbreviated	mid.
	BBBB	Wide	am, pm, midnight, noon, morning, afternoon, evening, night
	BBBBB	Narrow	md
Period standalone*	b, bb & bbb	Abbreviated	mid.
	bbbb	Wide	am, pm, midnight, noon, morning, afternoon, evening, night
	bbbbb	Narrow	md
Hour 1-12	h	Numeric: minimum digits	1, 12
	hh	Numeric: 2 digits + zero padded	01, 12
Hour 0-23	H	Numeric: minimum digits	0, 23
	HH	Numeric: 2 digits + zero padded	00, 23
Minute	m	Numeric: minimum digits	8, 59
	mm	Numeric: 2 digits + zero padded	08, 59
Second	s	Numeric: minimum digits	0... 59
	ss	Numeric: 2 digits + zero padded	00... 59
Fractional seconds	S	Numeric: 1 digit	0... 9
	SS	Numeric: 2 digits + zero padded	00... 99
	SSS	Numeric: 3 digits + zero padded (= milliseconds)	000... 999
Zone	z, zz & zzz	Short specific non location format (fallback to O)	GMT-8
	zzzz	Long specific non location format (fallback to OOOO)	GMT-08:00
	Z, ZZ & ZZZ	ISO8601 basic format	-0800
	ZZZZ	Long localized GMT format	GMT-8:00

Field type	Format	Description	Example Value
	ZZZZZ	ISO8601 extended format + Z indicator for offset 0 (= XXXXX)	-08:00
	O, OO & OOO	Short localized GMT format	GMT-8
	OOOO	Long localized GMT format	GMT-08:00

Pre-defined format options

Option	Equivalent to	Examples (given in `en-US` locale)
`'short'`	`'M/d/yy, h:mm a'`	`6/15/15, 9:03 AM`
`'medium'`	`'MMM d, y, h:mm:ss a'`	`Jun 15, 2015, 9:03:01 AM`
`'long'`	`'MMMM d, y, h:mm:ss a z'`	`June 15, 2015 at 9:03:01 AM GMT+1`
`'full'`	`'EEEE, MMMM d, y, h:mm:ss a zzzz'`	`Monday, June 15, 2015 at 9:03:01 AM GMT+01:00`
`'shortDate'`	`'M/d/yy'`	`6/15/15`
`'mediumDate'`	`'MMM d, y'`	`Jun 15, 2015`
`'longDate'`	`'MMMM d, y'`	`June 15, 2015`
`'fullDate'`	`'EEEE, MMMM d, y'`	`Monday, June 15, 2015`
`'shortTime'`	`'h:mm a'`	`9:03 AM`
`'mediumTime'`	`'h:mm:ss a'`	`9:03:01 AM`
`'longTime'`	`'h:mm:ss a z'`	`9:03:01 AM GMT+1`
`'fullTime'`	`'h:mm:ss a zzzz'`	`9:03:01 AM GMT+01:00`