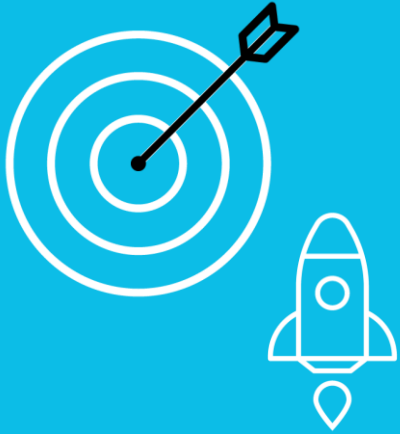# Angular 8

Lesson 3: Routing, Pipes & Forms
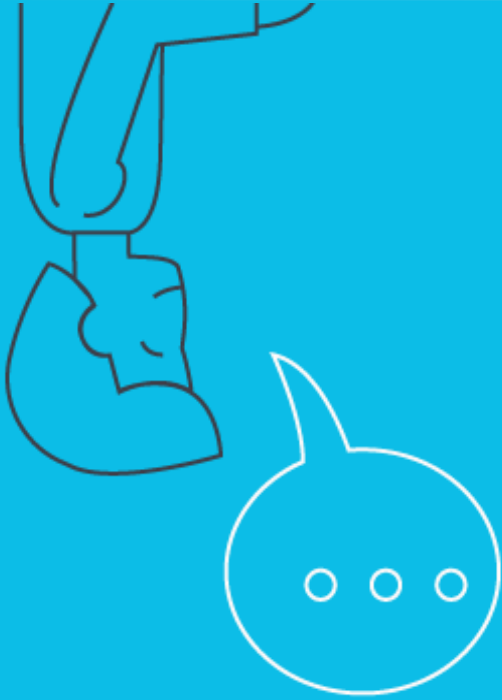
KEY OBJECTIVES

At the end of this lesson, you will be able to:

- Discuss what is router

- Discuss the features and functions of router

- List the types of parameter

- Discuss what is pipes

- Reveal the Key points of Pipes

- Discuss the Introduction of Forms

- List and discuss the types of forms

- Differentiate between the types of Form
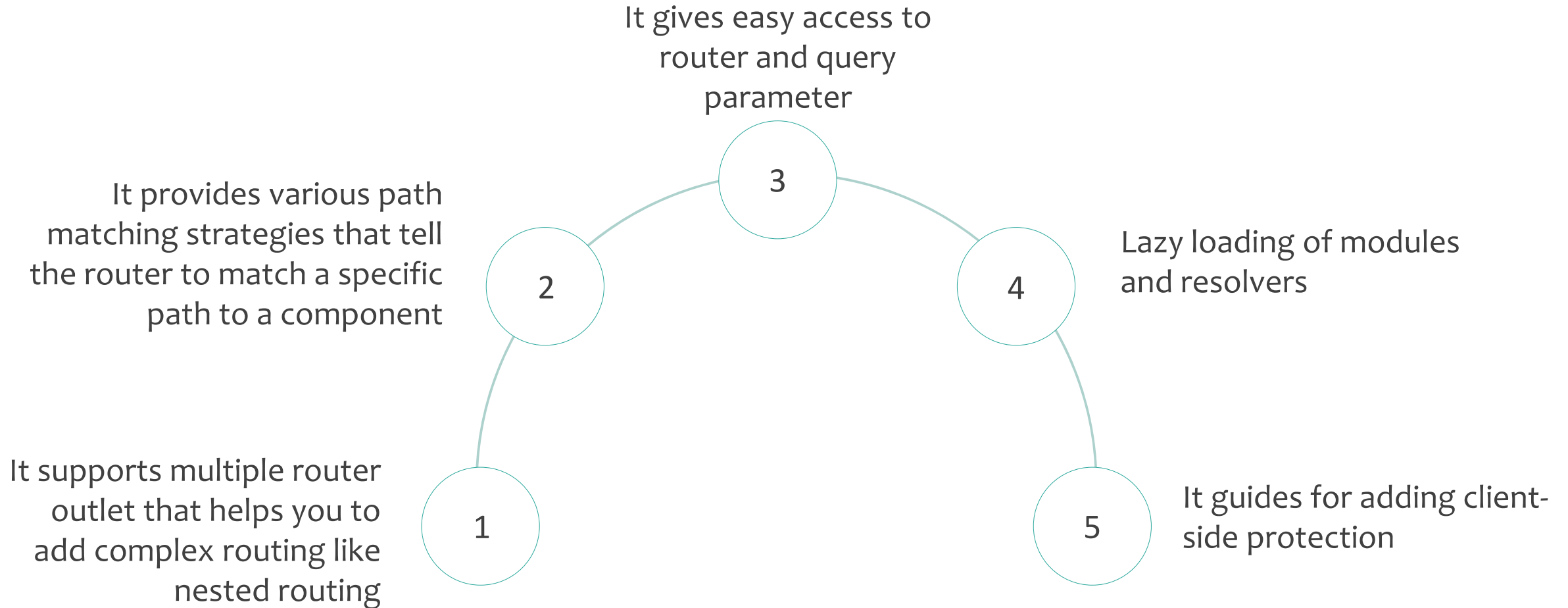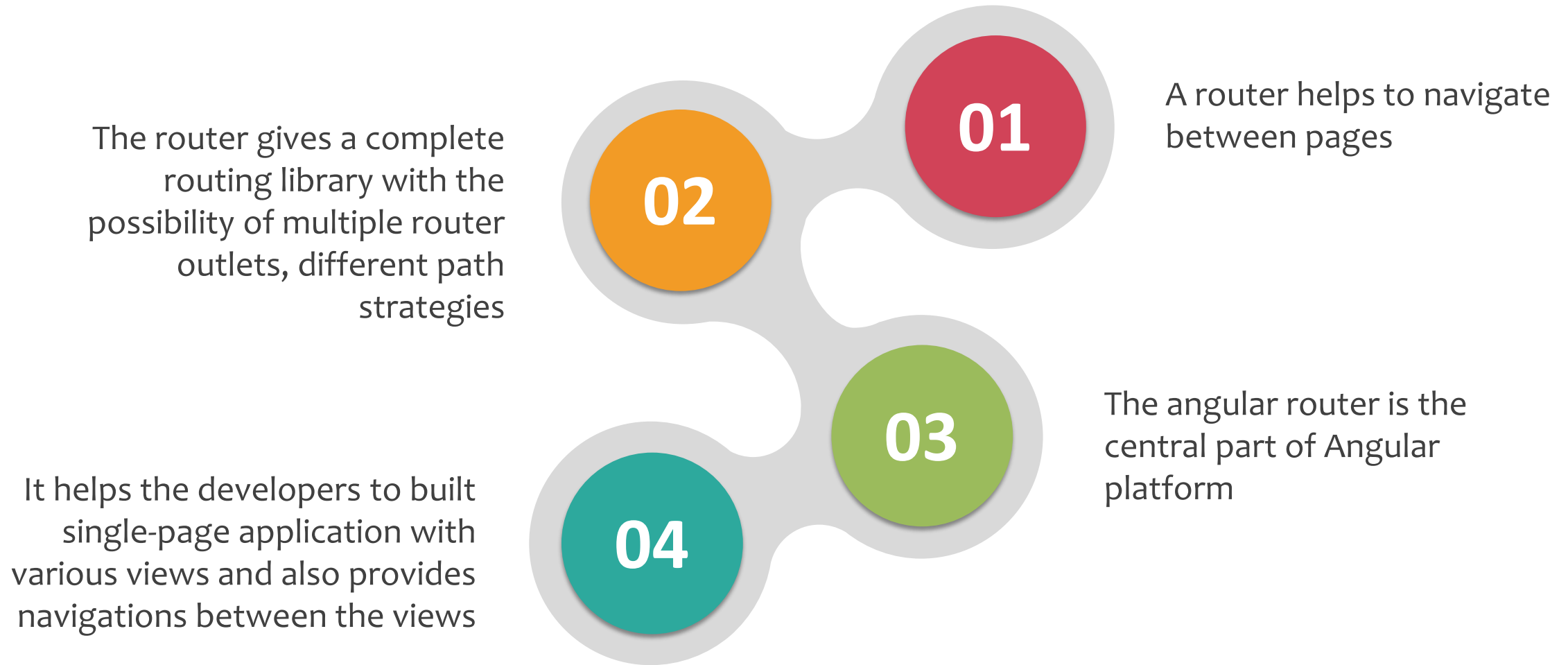
Routing, Pipe & Forms

Routers

What is Router?

The Angular router is the powerful JavaScript router developed and maintained by Angular team, and it is installed from the package@angular/router.

Features of Router

It gives easy access to router and query parameter

3

It provides various path matching strategies that tell the router to match a specific path to a component

2

Lazy loading of modules and resolvers

4

It supports multiple router outlet that helps you to add complex routing like nested routing

1

It guides for adding client-side protection

5

Basic Functions of Router

01 A router helps to navigate between pages

The router gives a complete routing library with the possibility of multiple router outlets, different path strategies **02**

03 The angular router is the central part of Angular platform

It helps the developers to built single-page application with various views and also provides navigations between the views **04**
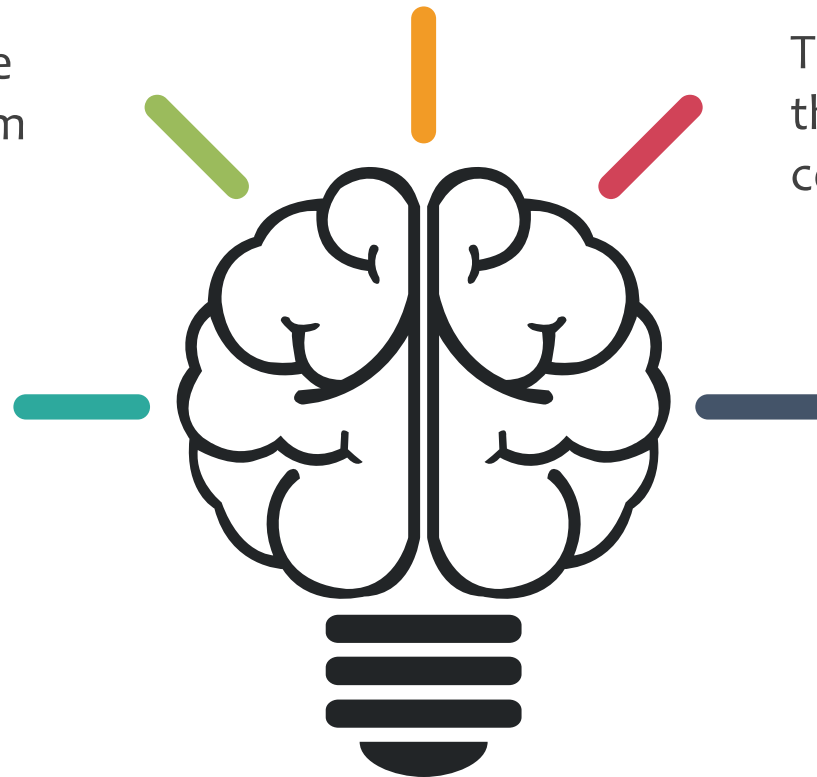
Router parameter

- Route parameter allows passing values in the URL to the component so that it can dynamically change the content view

- The component that views the details of the product will need a route parameter

- The angular router provides properties to get router parameter and the properties are:

PROPERTIES	DESCRIPTION
Snapshot	The current snapshot of this route
url	An observable of the URL segments matched by this route.
Params	An observable of the matrix parameters scoped to this route.
queryParams	An observable of the query parameters shared by all the routes.
fragment	An observable of the URL fragment shared by all the routes.
Data	An observable of the static and resolved data of this route.
outlet	string The outlet name of the route, a constant.
component	Type<any> | string | null The component of the route, a constant.

Wildcard Parameter

It consists of two asterisks and matches every URL

Wild card route intercept the invalid URLs and checks them easily

The router uses this route if the previous route configuration does not match

If any URL causes the router to throw an error and crash the app, the wildcard route is used to solve the problem

The wildcard route navigates to a custom '404 not found' components or redirects it to an existing route

Redirecting Parameter
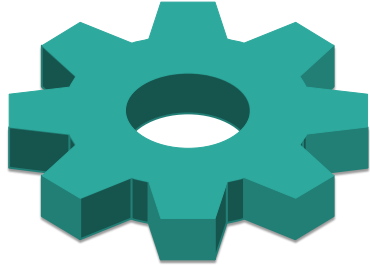
When an application is launched the original URL in the browser bar displays "localhost: '4200' and this does not match the concrete configured route.

This happens when the router falls through the wildcard route and displays the PageNotFoundComponent.

To solve this problem the applications need a default route to a valid page and the redirect route is used to translate the initial relative URL to the desires default path

Component router

- Angular has many improved modules to the Angular ecosystem, and this includes a new router called component router

- The component router is a highly configured and feature-packed router

- The concept linked to the router are:

1. The router outlet

2. Routes and path

3. Navigation
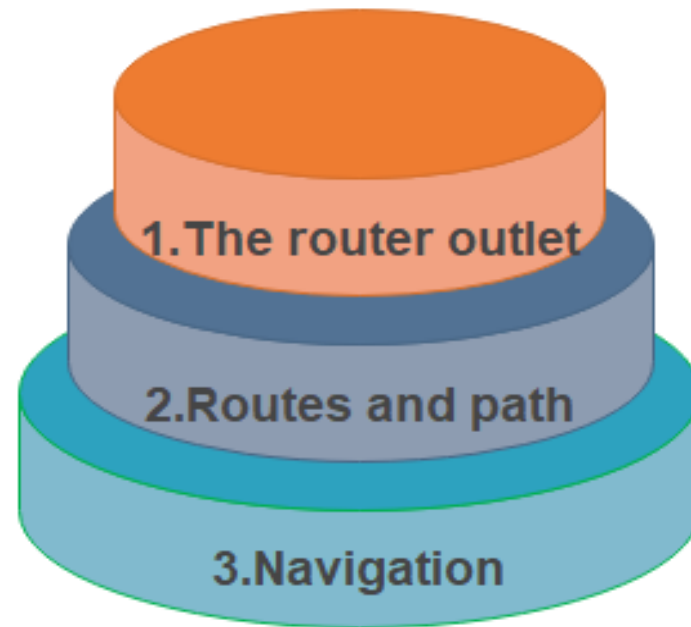
Component Router

THE ROUTER OUTLET

- The router-outlet: It is directive accessible from the router library where the router inserts the component and gets matched based on the current browsers URL
- SYNTAX:
  ```
  <router-outlet>
  </touter-outlet>
  ```

ROUTES AND PATHS

- Routes and paths: Routes are comprised of a path and a component attributes
- The path suggested to the URL that determines a unique view that can be displayed
- It also indicates to the Angular component that requires to be associated with a path
- In component, each route maps a URL path
- SYNTAX:
  ```
  {path:'contacts',
  component:
  ContactListComponent}
  ```

NAVIGATION

- The Angular router gives a router link directive to build the navigation links
- This directive creates the path associated with the component to navigate
- SYNTAX:
  ```
  <a[routerLink]= "
  '/contacts'
  ">Contacts</a>.
  ```

CHECK POINT

The component that views the details of the product will need _____ parameter?

1. Wildcard Parameter

2. Static parameter

3. Redirecting parameter

4. Router Parameter

CHECK POINT

The component that views the details of the product will need _____ parameter

1. Wildcard Parameter

2. Static parameter

3. Redirecting parameter

4. Router Parameter

Routing, Pipes & Forms

Pipes

Pipes

Pipes are one of the useful features in angular

They are used to transform values in an angular template

2

1

3

It takes integer, strings, array, and dates as input and divides it with | to convert it in the format as needed and displayed same as the browser

4

- Pipes are almost similar to styles, and it can also be applied in the HTML template as a form of Style
- SYNTAX:
    {{title | uppercase}}

Pipe Definition reveals a key Points
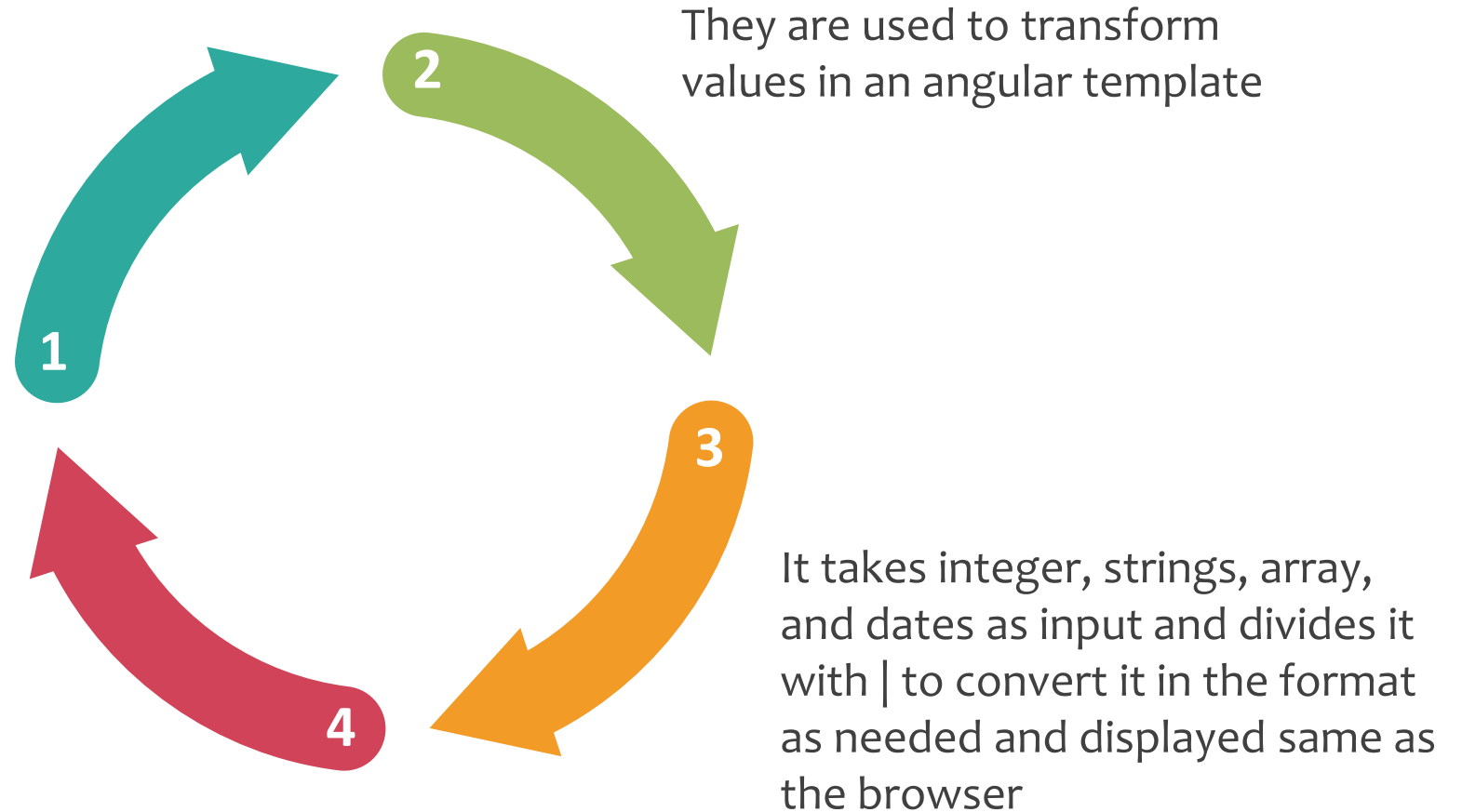
A pipe is a class and is decorated with pipe metadata.

The pipe class performs the PipeTransform interface's that accepts an input value supported by optional parameters and delivers the transformed value

There will be one extra argument to the transform method for each parameter moved to the pipe. The pipe has one so parameter: the exponent

To notify Angular that this is a pipe, you use the @Pipe decorator, and it can be imported from the Angular core library

The @Pipe decorator allows defining the pipe name that is used within template expressions. It must be a valid JavaScript identifier

Custom Pipes

- A user can write his own custom pipe.

- Example: Here is a custom pipe named as '`ExponentialPowerPipe`' below

```
import { Pipe, PipeTransform } from '@angular/core';

/* * hike the value exponentially
  Assign an exponent argument that defaults to 1.
  Usage:
        value | exponentialPower:exponent
  Example:
        {{ 2 | exponentialPower:10 }}
        formats to: 1024
*/

@Pipe ({name: 'exponentialPower'})
export class ExponentialPowerPipe implements PipeTransform {
    transform(value: number, exponent?: number): number {
    return Math.pow(value, isNaN(exponent) ? 1 : exponent);
    }
    }
```

CHECK POINT

One of the below symbol are used divide the input and convert the format needed in pipes

1. /

2. ~

3. |

4. ||

CHECK POINT

One of the below symbol are used divide the input and convert the format needed in pipes

1. /

2. ~

3. |

4. ||

Routing, Pipe & Forms

Forms

springpeople
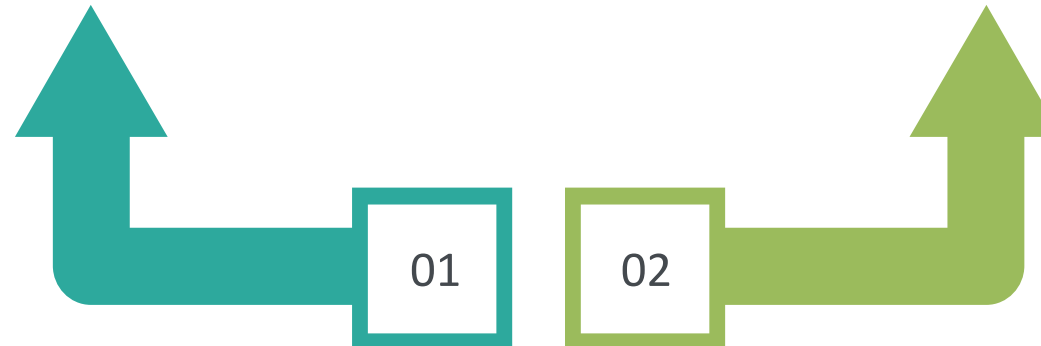
01 In Angular forms are used to handle user inputs

02 In an application form are used to approve a user to log in, update profile, to enter information and to perform many other data-entry tasks
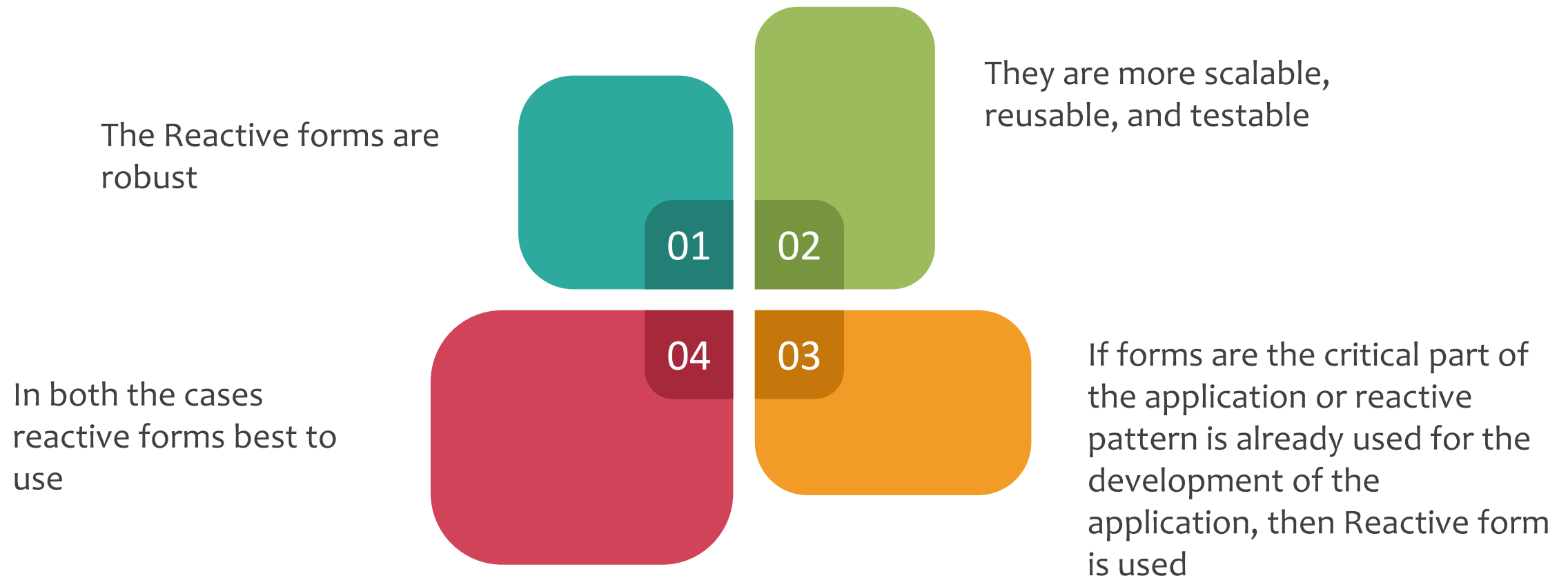
Types of Forms

Here both the methods are used to get user input event from the view, verify the user input, create a form model to update and provide a way to track changes, they also process and manage form data differently

The types of form are:

Reactive forms

01

02

Template-driven forms

Reactive Form

The Reactive forms are robust

They are more scalable, reusable, and testable

01

02

04

03

In both the cases reactive forms best to use

If forms are the critical part of the application or reactive pattern is already used for the development of the application, then Reactive form is used

Template-Driven Form

If the application requires fundamental form and logic, then it can be managed in the template

03

01
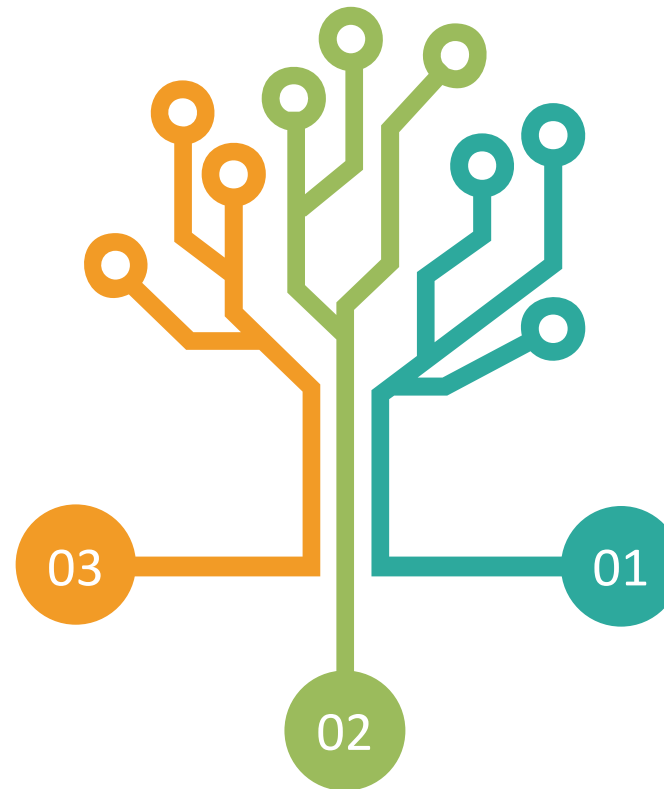
It is used to add a form to an app, such as email list signup form to an app

02

It is easy to add to an app but does not scale well as reactive form

Difference between Type of Forms

INDEX	Reactive Forms	Template-Driven Forms
Setup	They are more explicit and are created in the compound class	They are less explicit and directives create them
Data model	Structured	Unstructured
Form validation	Functions	Directives
Predictability	Synchronous	Asynchronous
Scalability	Low-level API access	Abstraction on top of APIs
Mutability	Immutable	Mutable

Developing Bootstrap Form

- Create an Angular form app named angular form and run the server by using the commands

```
ng new angularform
cd angularform
ng serve
```

- ```
  npm install bootstrap --
  ```
  given

  save

- ```
  "styles": [
      "./node_modules/bootstrap/dist/css/bootstrap.min.css
  ",
      "src/styles.css"
  ],
  ```

Form Validation

- Validation is a fundamental part of managing any set of forms

- Angular 8 gives a set of built-in validators besides creating a custom validator

- There are two validator functions, and they are:

01

02

Sync Validators

- It is a type of service that controls the situation and instantly returns either a set of validation error or null

- The second argument is entered when we instantiate a Form Control

Async Validator

- The functions take a control instance and return a promise that later releases a set of validation error or null

- The third argument is entered when we can instantiate a Form Control

CHECK POINT

Which methods are used to get user input event from the view, verify, create, update the input and also process and manage form data differently?

1. Template-driven Form

2. Reactive Form
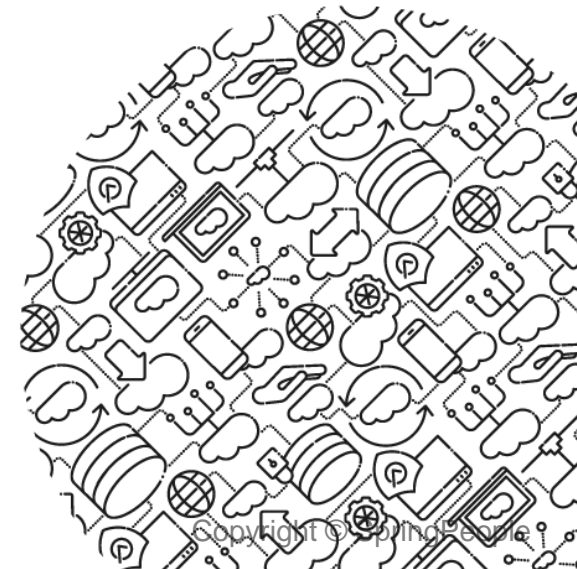
3. Both 1 & 2

4. None

CHECK POINT

Which methods are used to get user input event from the view, verify, create, update the input and also process and manage form data differently?

1. Template-driven Form

2. Reactive Form

3. Both 1 & 2

4. None

DOUBT'S

We wish you to take nothing except "Pure Learning." Leave the doubt's to us.

LET'S SUMMARIZE

- The Angular router is the powerful JavaScript router developed and maintained by the Angular team, and it is installed from the package@angular/router

- They are used to transform values in an angular template

- In an application form are used to approve a user to log in, update profile, to enter information and to perform many other data-entry tasks

THANK YOU