SVG in templates

It is possible to use SVG as valid templates in Angular. All of the template syntax below is applicable to both SVG and HTML. Learn more in the SVG 1.1 \square and 2.0 \square specifications.

See the live example / download example for a working example containing the code snippets in this quide.

Why would you use SVG as template, instead of simply adding it as image to your application?

When you use an SVG as the template, you are able to use directives and bindings just like with HTML templates. This means that you will be able to dynamically generate interactive graphics.

Refer to the sample code snippet below for a syntax example:

```
src/app/svg.component.ts
 import { Component } from '@angular/core';
@Component({
   selector: 'app-svg',
  templateUrl: './svg.component.svg',
   styleUrls: ['./svg.component.css']
})
 export class SvgComponent {
   fillColor = 'rgb(255, 0, 0)';
   changeColor() {
     const r = Math.floor(Math.random() * 256);
     const g = Math.floor(Math.random() * 256);
     const b = Math.floor(Math.random() * 256);
     this.fillColor = `rgb(${r}, ${g}, ${b})`;
  }
}
```

Add the following code to your svg.component.svg file:

Here you can see the use of a click() event binding and the property binding syntax ([attr.fill]="fillColor").

RESOURCES	HELP	COMMUNITY	LANGUAGES
About	Stack Overflow	Events	简体中文版
Resource Listing	Gitter	Meetups	正體中文版
Press Kit	Report Issues	Twitter	日本語版
Blog	Code of Conduct	GitHub	한국어
Usage Analytics		Contribute	

Super-powered by Google ©2010-2020. Code licensed under an MIT-style License. Documentation licensed under CC BY 4.0.

Version 10.0.10-local+sha.b32126c335.