**Components**

**Root Component**

**What?**

beginning of component, the very first component that is referenced and hosted in the main index. Everything else in Angular builds off of this root.

**When?**

import all components in one component, view

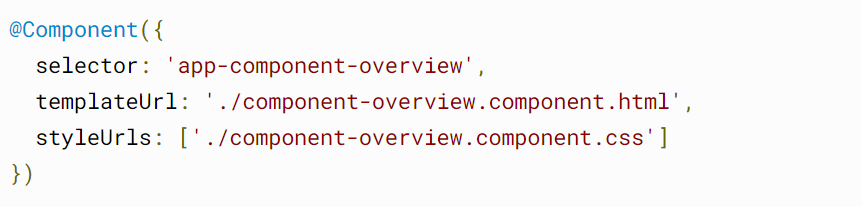
**Component Syntax**

Run the **ng generate component <component-name>** command, And Then add **<component-name>** to another component That you need.

**Selectors, Templates, and Styles**

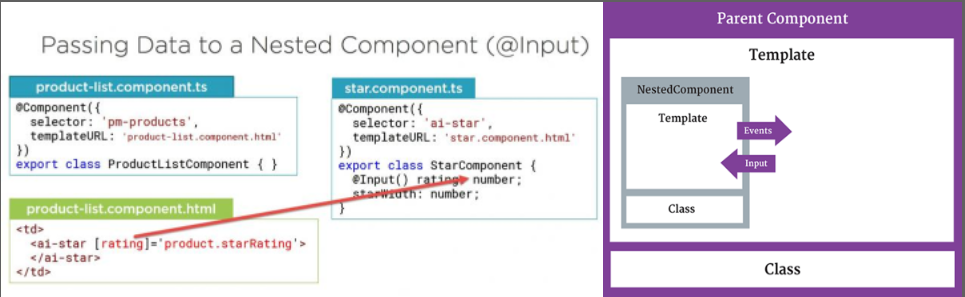


**Import File Base Selectors, Templates, and Styles**



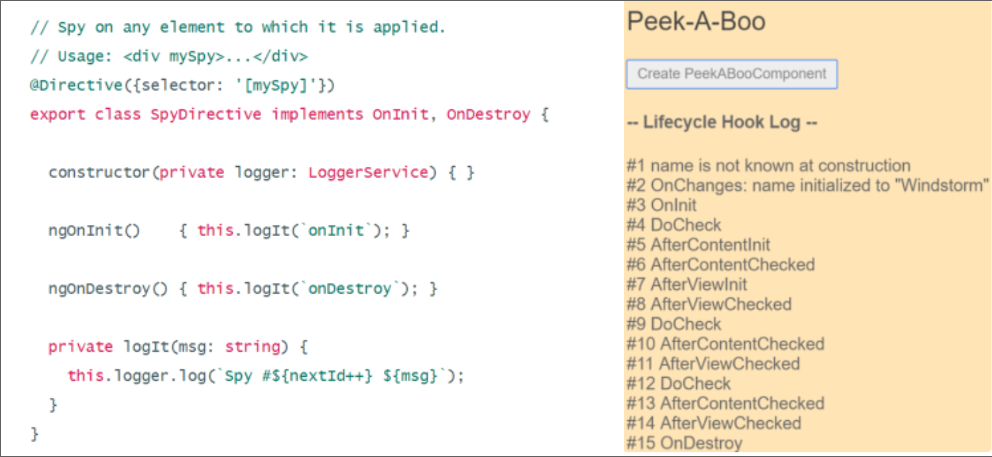
**Nested Components**

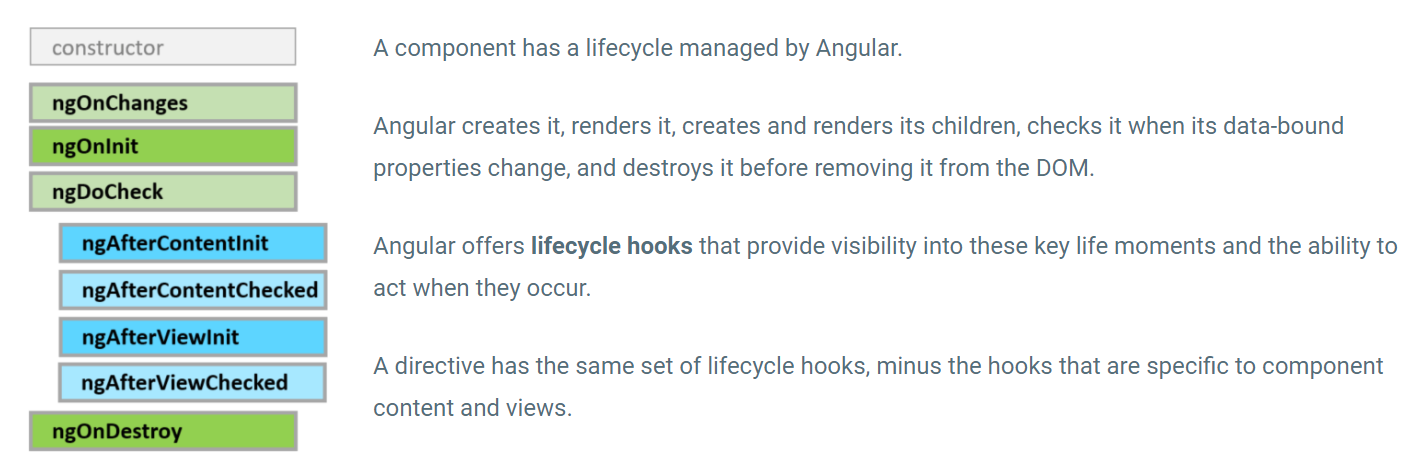
What is a Nested Component? Angular allows us to have a different child, or nested component, which is the same component we normally use in an Angular application. The difference between them is that this child component **contains the logic which can be used into the parent component as a single unit**.



**Component Life Cycle**

The sneaky spy directive is simple, consisting almost entirely of ngOnInit() and ngOnDestroy() hooks that log messages to the parent via an injected LoggerService.



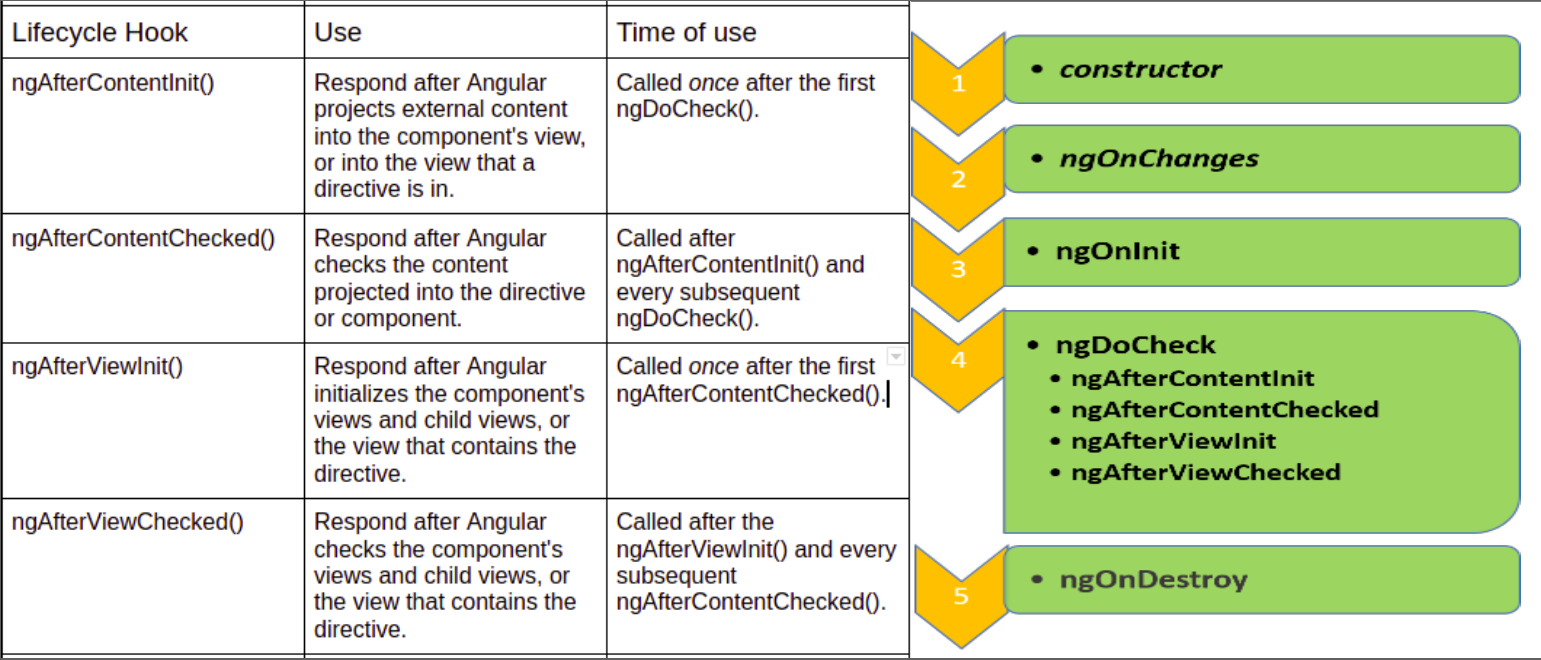
**­**

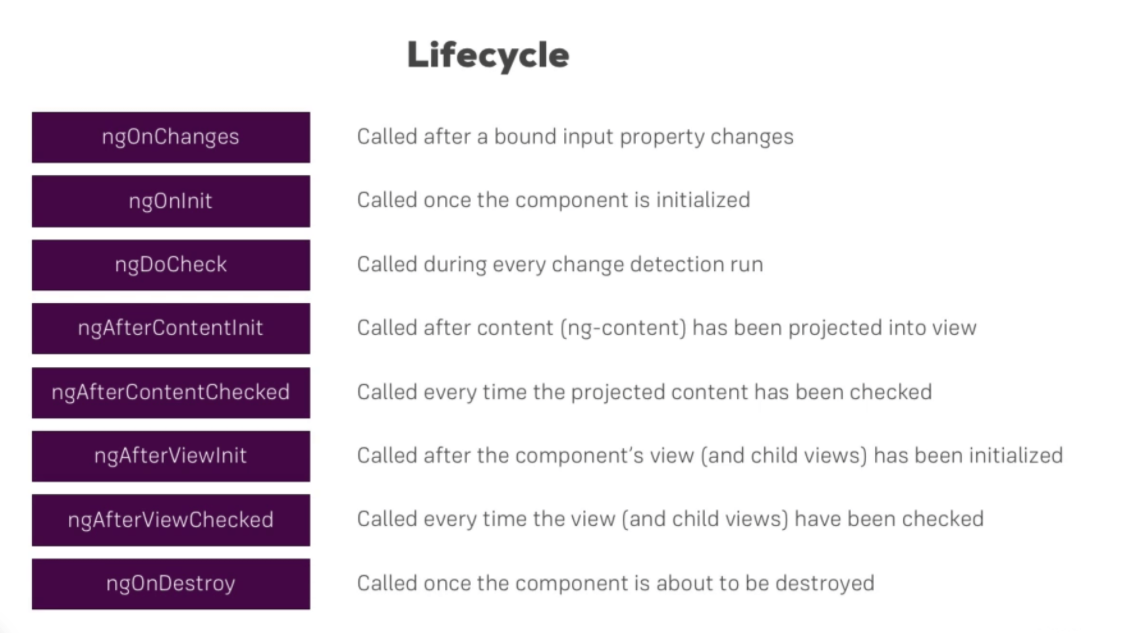
|  |  |
| --- | --- |
| Constrocter | In class-based object-oriented programming, a constructor (abbreviation: ctor) is a special type of subroutine called to create an object |
| ngOnChanges() | Called before ngOnInit() (if the component has bound inputs) and whenever one or more data-bound input properties change. |
| ngOnInit() | Called once, after the first ngOnChanges(). ngOnInit() is still called even when ngOnChanges() is not (which is the case when there are no template-bound inputs). |
| ngDoCheck() | Called immediately after ngOnChanges() on every change detection run, and immediately after ngOnInit() on the first run. |
| ngAfterContentInit() | Called once after the first ngDoCheck() |
| ngAfterContentChecked() | Called after ngAfterContentInit() and every subsequent ngDoCheck(). |
| ngAfterViewInit() | Called once after the first ngAfterContentChecked() |
| ngAfterViewChecked() | Called after the ngAfterViewInit() and every subsequent ngAfterContentChecked(). |
| ngOnDestroy() | Called immediately before Angular destroys the directive or component. |

­­

**Initialization Hooks to Destroy Hooks**

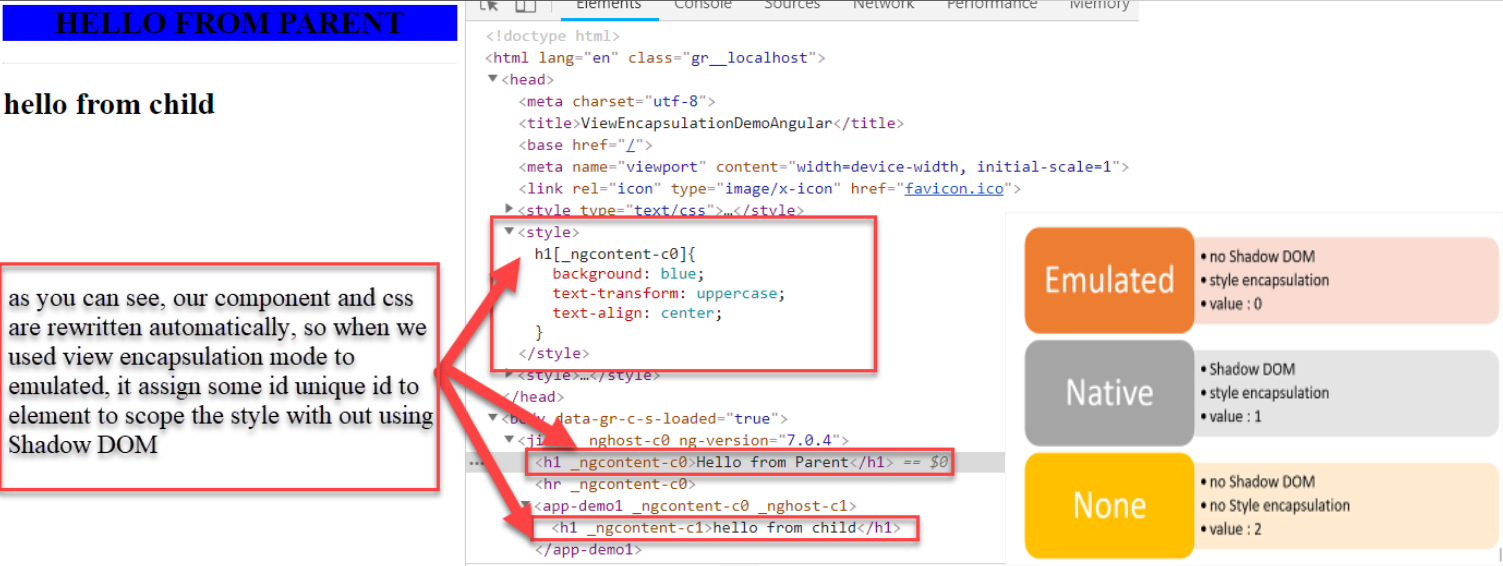
Hooks in the parent component are called **when the child component's content has been initialized**, when the UI or view of the child component has been initialized and rendered, or when the child component's change detection has been run





**View Encapsulation**

View encapsulationlink. In Angular, **a component's styles can be encapsulated within the component's host element** so that they don't affect the rest of the application. The Component 's decorator provides the encapsulation option which can be used to control how the encapsulation is applied on a per component basis.



**Web Components**

Angular elements are Angular components packaged as **custom elements** (also called Web Components), a web standard for defining new HTML elements in a framework-agnostic way. ... Transforming a component to a custom element makes all of the required Angular infrastructure available to the browser.

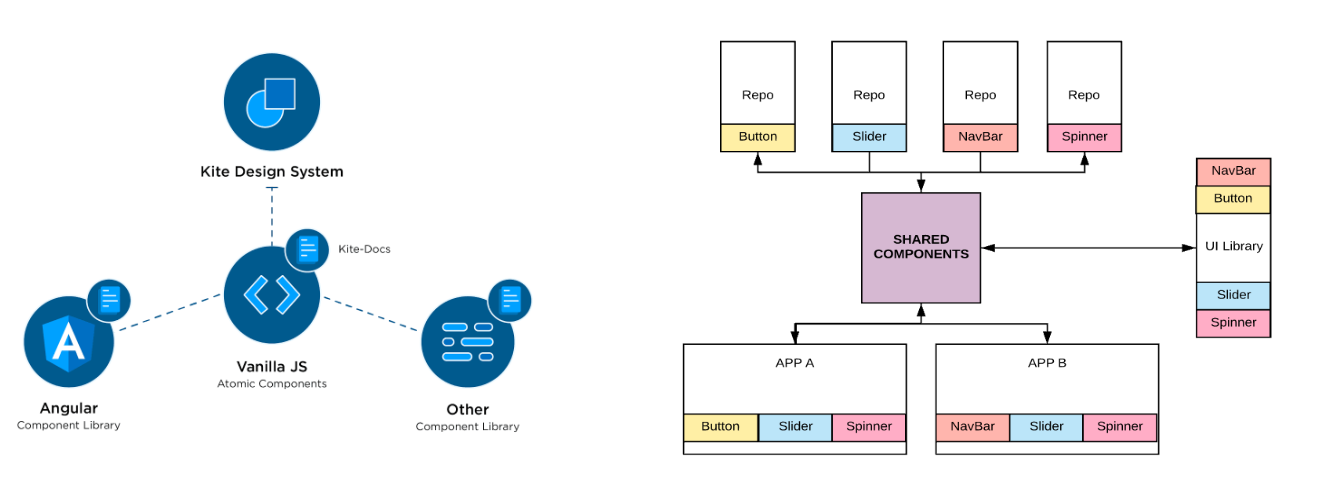
Web Components essentially allow us to create new HTML tags/elements using JavaScript. Let’s see a small example of how this is done, purely with JavaScript.



**Depth Explationation Web Components** [**https://indepth.dev/posts/1116/angular-web-components-a-complete-guide**](https://indepth.dev/posts/1116/angular-web-components-a-complete-guide)

**Re-usable Components**

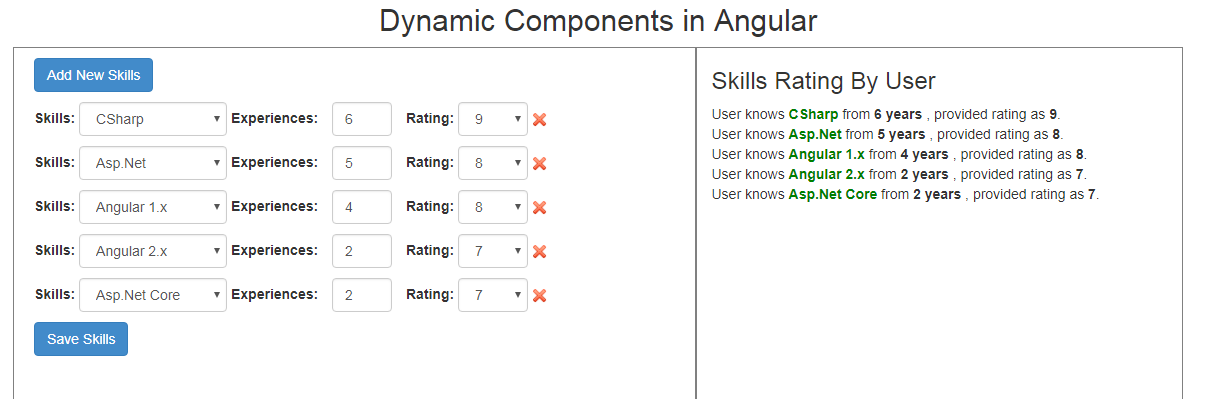
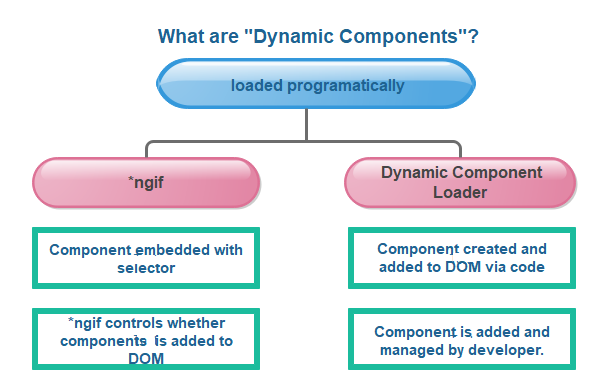
There are two main ways to create reusable components in Angular: **Pass inputs to the component, passing the necessary data to the component used for rendering and configuring the component**. This normally involves iterating over the provided data and follow a convention for how to render the data.



**Adding Components Dynamically**

Dynamic component means, that **the location of the component in the application is not defined at build time**. That means, that it is not used in any angular template. Instead, the component is instantiated and placed in the application at runtime.

**How to add dynamically add component https://angular.io/guide/dynamic-component-loader**

****

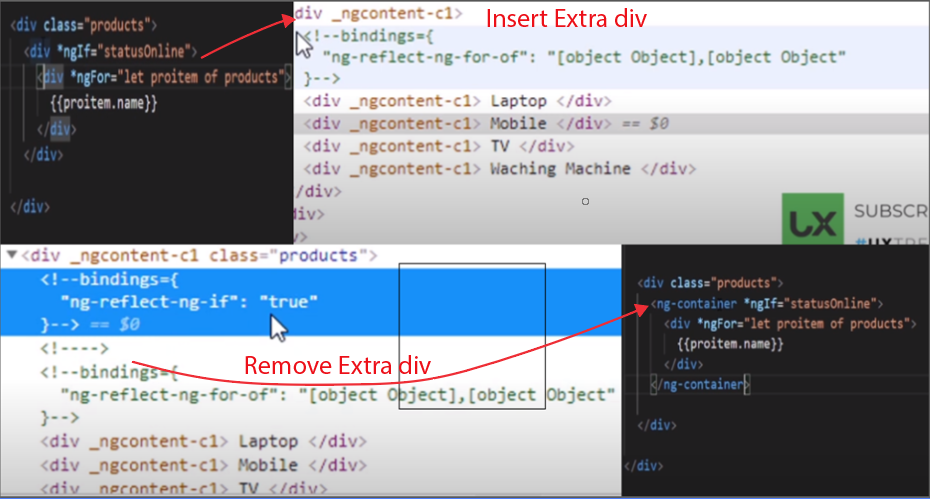
**Entry Components**

A bootstrapped component is an entry component, All router components must be entry components. Because this would require you to add the component in two places (router and entryComponents) the

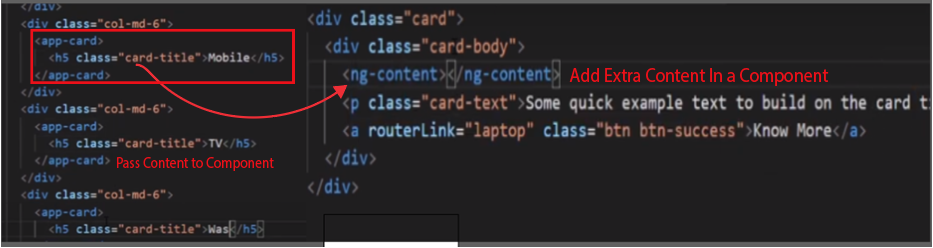
Compiler is smart enough to recognize that this is a router definition and automatically add the router component into entryComponents.



**Ng-Container**



**Ng-Content**



**Ng-template**

