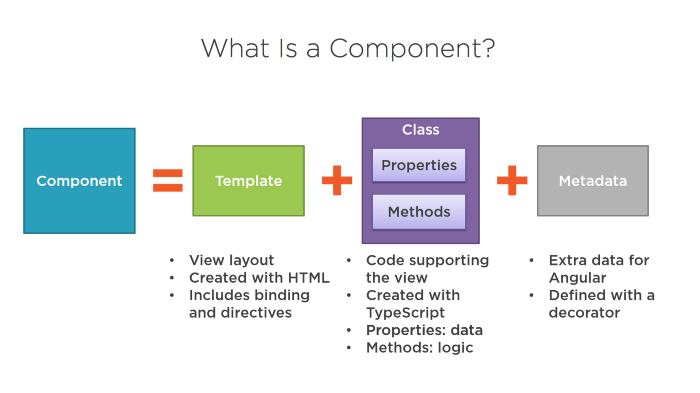
**Smart component:** also known sometimes as application-level or container components are the components having application level logic.

**Presentation Components:** also known sometimes as pure or dumb components are the component having view/display logic for rendering.

Before we jump into architecture lets have a look at component.



As depicted above we have 2 major things in a component

1. business logic
2. view logic.

This Architecture strategy focuses on separation of the same.

**Presentation or dumb components:**A component which does the job of presentation based on data input from the smarter component. These component doesn’t handle any kind business logic and be dependent on other component for data. Few points to remember:

1. Doesn’t have access to service
2. Doesn’t store data in backend
3. Focuses on presentation
4. Have changeDectection as onPush

Example:

As shown in above example the component receives the data as input and only contains the display logic. This component can be reused anytime we need similar listing.

**Smart Components:** A component which is responsible for business logic interacting with services, managing state generating/retrieving data for presentation component. Few points to remember:

1. Have states
2. Implements services
3. any kind of business logic happening for that component/feature
4. Not much focus on presentation

Example:

As shown in the above example app-medicines connects with rest API using HttpClient and retrieves the data applies the business logic and share the data with presentation component. This helps in keeping the business logic and view separate.

**Benefits**:

* Components are easily shared across our application, If configured properly
* Loose coupling
* Consistent Experience
* Reductions in active observables (better performance)