Data Binding

Agenda

01 What is Data Binding

02 Attribute versus Property

03 Types of Data Binding

04 Interpolation

05 Property Binding

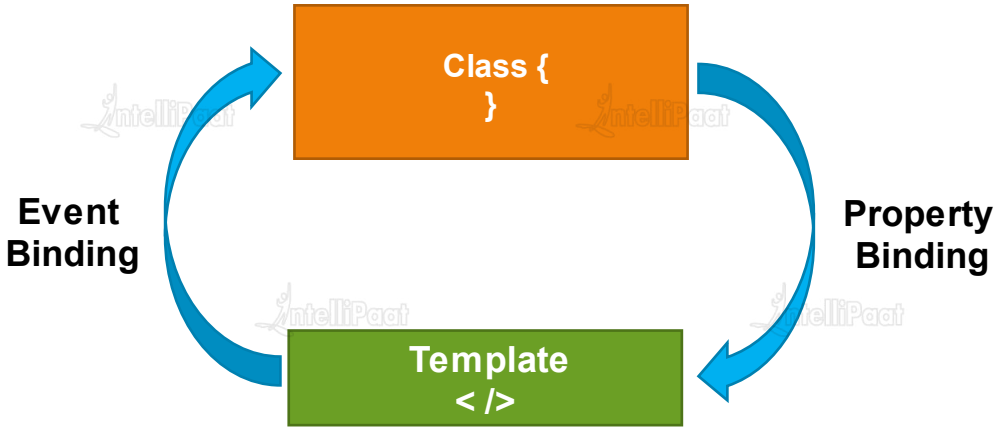
06 Event Binding

07 Two way data binding

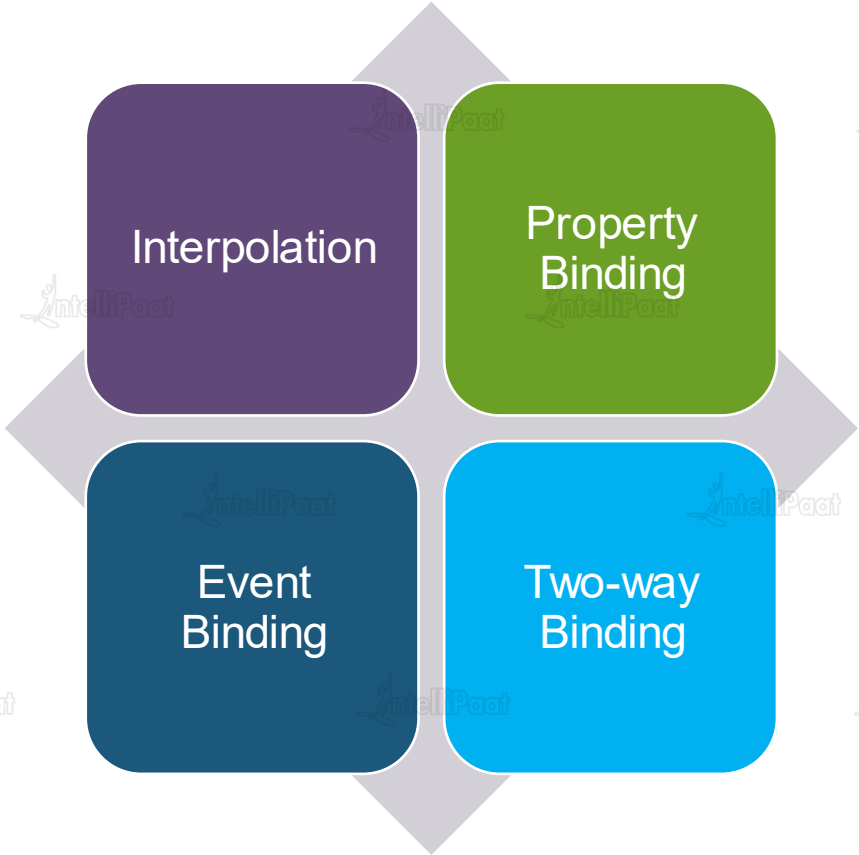
08 Demo

Data Binding

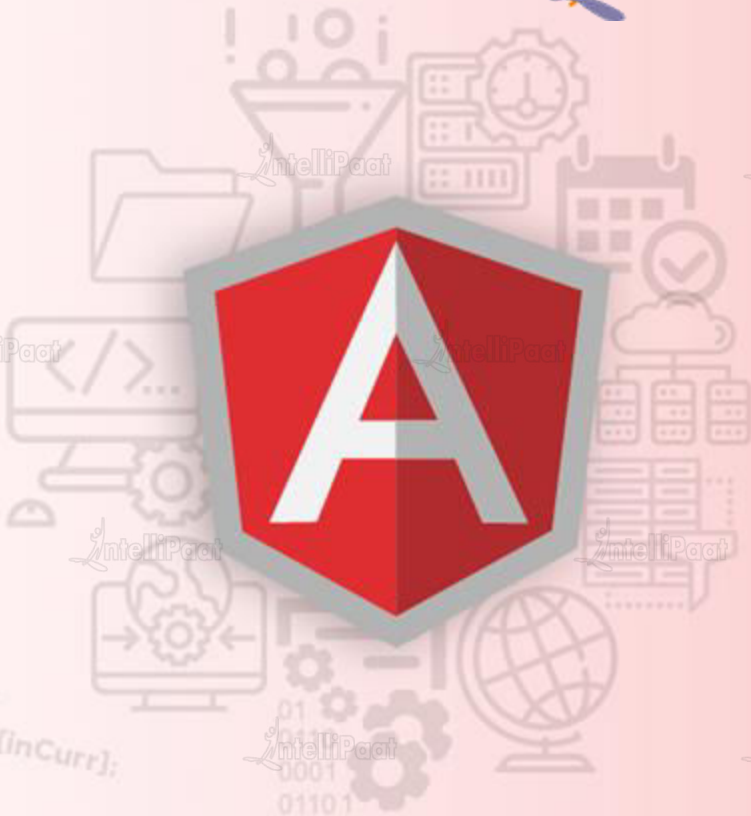
- Data Binding is a process that creates a connection between the application's UI and the data.
- When the data changes its value, the UI elements that are bound to the data, will also change.
- Angular handles data binding by synchronizing the state of the view, with the data in the component.
- Data-binding can be one-way, where a change in the state affects the view, or two-way, where a change from the view can also change the model.



Types of Data Binding



Attribute vs Property



Attribute vs Property

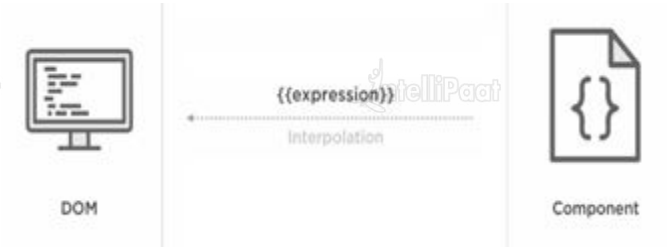
- Attributes and properties are not same
- Attributes belong to HTML element
- Properties belong to DOM
- Attributes initialize DOM properties and then they are done.
- Attributes once initialized then they can not be changed

  whereas Property values can be changed.

Interpolation

Interpolation

- Interpolation is a technique that allows the user to bind a value to a UI element.
- Interpolation binds the data one-way. This means that when value of the field bound using interpolation changes, it is updated in the page as well.




```
< input type="text"  value={{name}}/>
```

Interpolation

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  template: `
    <h1>{{title}}</h1>
    <h2>My favorite hero is: {{myHero}}</h2>
    `
})
export class AppComponent {
  title = 'Tour of Heroes';
  myHero = 'Windstorm';
}
```

Property Binding

Property Binding

- Property binding is used to bind values to the DOM properties of the HTML elements.
- It sends information from the component class to the view.
- Property bindings are evaluated on every browser event and any changes made to the objects in the event, are applied to the properties.


```
<p><img [src]="heroImageUrl"> is the <i>property bound</i> image.</p>

<p><img bind-src="heroImageUrl"> is the <i>property bound</i> image.</p>
```

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  template: `
    <h1>{{title}}</h1>
    <h2>My favorite hero is: {{myHero}}</h2>
     <input type=text [disabled]= isUnchanged
value={{myHero}}/>

})
export class AppComponent {
  title = 'Tour of Heroes';
  myHero = 'Windstorm';
   isUnchanged = false;
}
```

Property Binding

Interpolation vs Property Binding

Property Binding or Interpolation

You often have a choice between interpolation and property binding. The following binding pairs do the same thing:

```
<p><img src="{{heroImageUrl}}"> is the <i>interpolated</i> image.</p>

<p><img [src]="heroImageUrl"> is the <i>property bound</i> image.</p>

<p><img bind-src="heroImageUrl"> is the <i>property bound</i> image.</p>
```

- When rendering data values as strings, *Interpolation is convenient option as it increases readability.*
- When setting an element property to a non-string data value, you must use *property binding*.

Event Binding

Event Binding

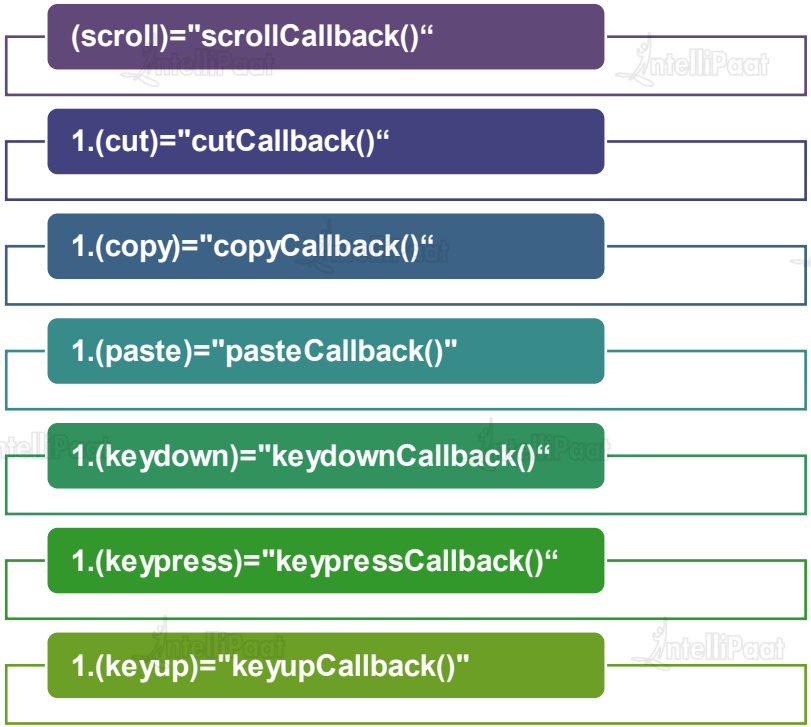
What is an event : A user expects a UI to respond to her/his actions on the page. Every such action would trigger an event on the page and the page has to respond by listening to these events.

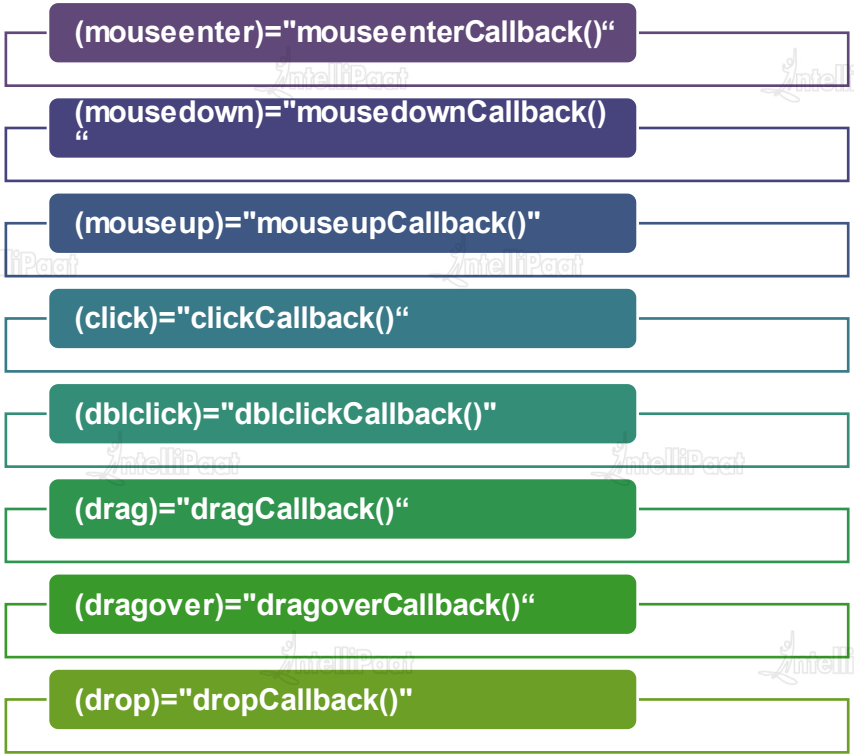
What is event binding : The event binding system provides us the way to attach a method defined in a component with an event.

```
<button class='btn' (click)='submit()'>Submit</button>
<button class='btn' on-click='submit()'>Submit</button>
```

Here, the method submit has to be defined in the component class. Any DOM event can be either prefixed with *on-* or can be enclosed inside parentheses to bind it with a method in the component class.

Types of Events





```
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  template: `
    <h1>{{title}}</h1>
    <h2>My favorite hero is: {{myHero}}</h2>
    <button (click)= myClickFunction($event)>Click  Me </button>
 })
export class AppComponent {
  title = 'Tour of Heroes';
  myHero = 'Windstorm';

myClickFunction(event):void{
console.log(event);
}
}
```

Event Binding

Two way data Binding

Two way data Binding

- The feature two-way binding in Angular is derived from the property and event bindings.
- The property and event bindings are directed one way with the former receiving data into view from the component object and the later sending data from the view to the component.
- The two-way binding is a combination of these two bindings; it gets the data from the component object to the view and sets the data from view to the component object.

Two way data Binding

The following snippet shows an example of a directive, *ngModel* to show how two-way binding can be used:

```
<input type="text" [(ngModel)]="name" />
     <div>{{name}}</div>
```

Here, the field name is two-way bound on the input box. When it is rendered on a page, it shows the existing value of the field and when the value is modified on the screen, it updates the value in the field.

```
import { NgModule }      from '@angular/core';
import { BrowserModule }  from '@angular/platform-browser';
import { FormsModule }    from '@angular/forms';  // Importing forms
module to this file

import { AppComponent }         from './app.component';
import { DemoComponent } from './demo.component';

@NgModule({
  imports: [
    BrowserModule,
    FormsModule // Importing forms module to application module
  ],

  declarations: [
    AppComponent,
    DemoComponent
  ],

  bootstrap: [ AppComponent ]
})

export class AppModule { }
```

Two way Data Binding

Demo