

# Fsd internal 1

**Create a simple multiplier Angular JS application which will multiply two numbers and display the result**

**Sol:**

**Source code**

```
<!DOCTYPE html>

<html>

<head>

  <title>First AngularJS Application</title>

  <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.7.9/angular.min.js"></script>

</head>

<body ng-app >

  <h1>First AngularJS Application</h1>

  Enter Numbers to Multiply:

  <input type="text" ng-model="Num1" /> x <input type="text" ng-model="Num2" />

</br>

  <h1 ng-model="Num1 * Num2">{{Num1 * Num2}}</h1>

</body>

</html>
```

**2Q) Write an angular program to share data from parent to child component via @Input Decorator.**

**Sol:**

**To create new component “ng g c child”**

**App.component.html:**

```
<app-child></app-child>
<app-child [cdata] = "pdata"></app-child>
```

### App.comoponent.ts

```
import { Component } from '@angular/core';
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'week3';
  pdata: String = "gamma";
}
```

### Child.componet.html:

```
<p>child works!</p>
<p>{{cdata}}</p>
```

### Child.component.ts:

```
import { Component,Input} from '@angular/core';
@Component({
  selector: 'app-child',
  templateUrl: './child.component.html',
  styleUrls: ['./child.component.css']
})
export class ChildComponent {
  @Input()
  cdata:String = "";
}
```

**3q) Write an angular program to share data from child to parent component via @Output Decorator.**

### App.component.html

```
<app-child (cdata) = "pdata = $event"></app-child>
{{pdata}}
```

### Child.component.html

```
<button (click)="passtoparent()">send</button>
```

### App.component.ts

```
import { Component } from '@angular/core';
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'week3';
  pdata: String = "";
}
```

### Child.component.ts

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

@Component({
  selector: 'app-child',
  templateUrl: './child.component.html',
  styleUrls: ['./child.component.css']
})
export class ChildComponent {
  @Output()
  cdata = new EventEmitter();
  message = "gamma";
  passtoparent(){
    this.cdata.emit(this.message);
  }
}
```

**5q) Write an inline template to display college details(name ,department name,and address etc).**

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

@Component({
  selector: 'app-root',
  template: `<h1> My Location Details</h1>
    <div>
      <div class="name">
```

```

        Place:{{ user.name }}
      </div>
      <div class="address">
        Name:{{user.address}}
      </div>
    </div>`,
    styles: [`
      h1 { font-weight: normal; color:red;}
      .name {font-weight: bold; color:blue; }
      .address {font-weight: bold; color:green;}

    `]
  })
export class AppComponent {
  title = 'myapp';
  user={name:"kkk",
    address:"goa"};
}

```

**6q) Write an external template to display student details(id and name.etc).**

**App.component.ts:**

```

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'myapp';
  student_detail={id:101, name:"Sanjay Paul"}
}

```

**App.component.html:**

```

<h1> This is External Template Example</h1>
<div>
  <div class="stud_id">
    Student ID:{{student_detail.id}}
  </div>

```

```
<div class="stud_name">
Student Name:{{student_detail.name}}
</div>

</div>
```

### App.component.css:

```
h1 { font-weight: normal; color:red;}
```

**q7)** Write an angular program to use built-in pipes: Uppercase, Lowecase, Date, Currency, Json and Slice

### app.component.html

```
<div>
  <h2>{{ name | uppercase }}</h2>
  <p>{{ name | lowercase }}</p>
  <p>{{ today | date }}</p>
  <p>{{ price | currency:'USD':true }}</p>
  <pre>{{ object | json }}</pre>
  <p>{{ text | slice:0:10 }}</p>
</div>
```

### App.component.ts

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
})
export class AppComponent {
  name = 'Angular';
  today = new Date();
  price = 123.456;
  object = { foo: 'bar', baz: 42 };
  text = 'Lorem ipsum dolor sit amet';
}
```

## 8q) Create custom PIPES to multiply two numbers.

Use command `ng g pipe multiply` to create new pipe page

App.component.html

```
<div>
  <h2>{{ 2 | multiply:3 }}</h2>
  <p>{{ 4.5 | multiply:2 }}</p>
</div>
```

Multiple.pipe.ts

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

@Pipe({
  name: 'multiply'
})
export class MultiplyPipe implements PipeTransform {

  transform(value: number, multiplier: number): number {
    return value * multiplier;
  }

}
```

## 9q) Write an angular example to demonstrates ng-if, ng-readonly, and ng-disabled directives.

App.component.html

```
<div>
  <h2>Conditional Display</h2>
  <input type="checkbox" [(ngModel)]="showText"> Show text
  <div *ngIf="showText">
    <input type="text" [(ngModel)]="text" [ngReadonly]="readonly">
    <button (click)="toggleReadonly()">Toggle Readonly</button>
    <button (click)="toggleDisabled()">Toggle Disabled</button>
  </div>
</div>
```

App.component.ts

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

@Component({
```

```

    selector: 'app-root',
    templateUrl: './app.component.html',
  })
  export class AppComponent {
    showText = false;
    text = 'Editable text';
    readonly = false;
    disabled = false;

    toggleReadonly() {
      this.readonly = !this.readonly;
    }

    toggleDisabled() {
      this.disabled = !this.disabled;
    }
  }
}

```

**10q) Write an angular program to perform Arithmetical operations using ngSwitch.**

```

<div>
  <h2>Arithmetic Operations</h2>
  <input type="number" [(ngModel)]="a">
  <select [(ngModel)]="op">
    <option value="+">+</option>
    <option value="-">-</option>
    <option value="*">*</option>
    <option value="/">/</option>
  </select>
  <input type="number" [(ngModel)]="b">
  <button (click)="calculate()">Calculate</button>
  <div [ngSwitch]="op">
    <div *ngSwitchCase="+">Result: {{ a + b }}</div>
    <div *ngSwitchCase="-">Result: {{ a - b }}</div>
    <div *ngSwitchCase="*">Result: {{ a * b }}</div>
    <div *ngSwitchCase="/">Result: {{ a / b }}</div>
    <div *ngSwitchDefault>Please select an operation</div>
  </div>
</div>

```

**App.component.ts**

```

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

```

```
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
})
export class AppComponent {
  a: number;
  b: number;
  op = '+';

  calculate() {
    // do nothing, the result is displayed using ngSwitch
  }
}
```