ng new hello-world 🡪Create new project

ng serve 🡪 compile, build and run on embedded development server

**Component**

**Type script**

**HTML**

**Menu Component**

**Page Component**

**Footer Component**

**Component** : Piece of display with some functionality behind it

Here <app-root> 🡪 Component selector

ng generate component Page1

ng g c Page1

It will create Page1 folder inside app.

constructor() – Runs when every class created

ngOnInit() – Runs after the constructor

Template expressions are dynamic

**page1.component.html**

<h2>{{pageName}}</h2>

**page1.component.ts**

pageName = 'Page One';

**Event Binding**

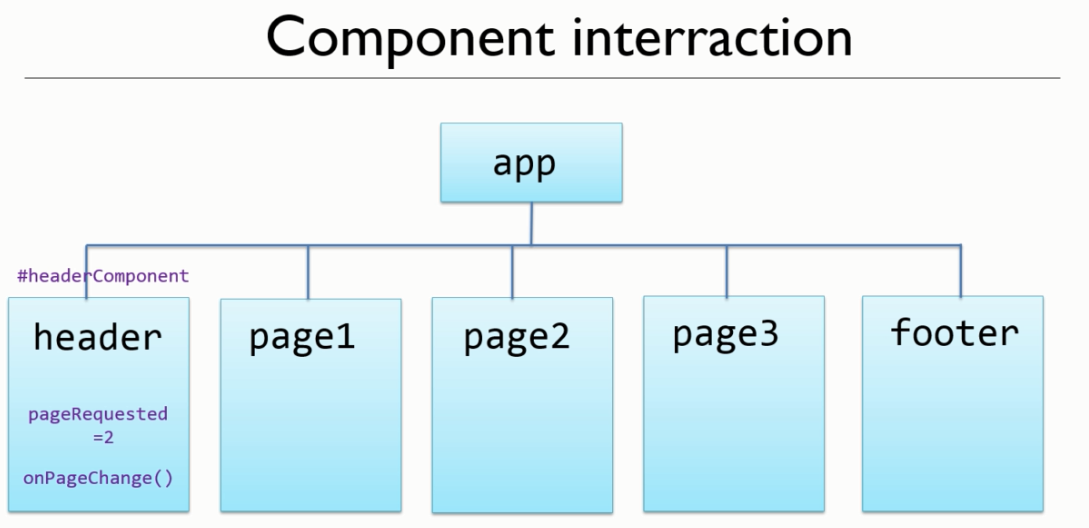
*page1.component.ts page1.component.html*

onButtonClick() { <button **(click)** ='onButtonClick()'> Click Me</button>

alert('Hello - the date is ' + new Date());

}

Here variable called with {{}} and method calls with ()



<app-header #headerComponent></app-header>

<app-page1 [hidden] = "headerComponent.pageRequested !== 1"></app-page1>

<app-page2 \*ngIf = "headerComponent.pageRequested === 2"></app-page2>

To bind a html attribute we need to use **[]**

**Selective Display**

<div [hidden] = “some expression”>

<div \*ngif = “some expression”>

\*ngif is structural directive in angular means it can change the DOM or change the structure of page.

Using structural directive we are manipulating the DOM.

**Component Interaction**

app.component.ts

import { FooterComponent } from './footer/footer.component';

@ViewChild('footer', {static: true})

footerComponent: FooterComponent;

updateLastAccess() {

this.footerComponent.lastAccessed = new Date().toDateString();

}

footer.componet.ts

lastAccessed = ‘’;

app.component.html

<app-footer #footer></app-footer>

<button (click)="updateLastAccess()">Update Last Access</button>

footer.component.html

Last Accessed {{lastAccessed}}

Template Reference

**Writing from HTML to CLASS**

Property Binding

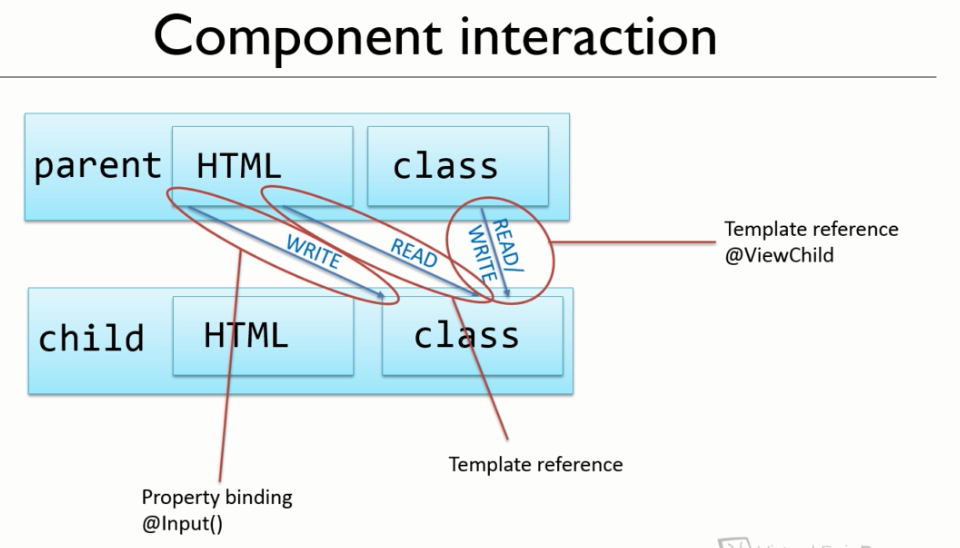
app.component.html

<app-footer #footer [lastAccessed]='startTime'></app-footer>

footer.componet.ts

@Input()

lastAccessed = '';



**Event Binding**

**header.component.ts**

@Output()

pageChangeEvent = new EventEmitter<number>();

onPageChange(page: number) {

    this.pageRequested = page;

    console.log(this.pageRequested);

    this.pageChangeEvent.emit(page);

  }

**app.componet.html**

<app-header (pageChangeEvent)="incrementHitCounter($event)" #headerComponent></app-header>

**app.component.ts**

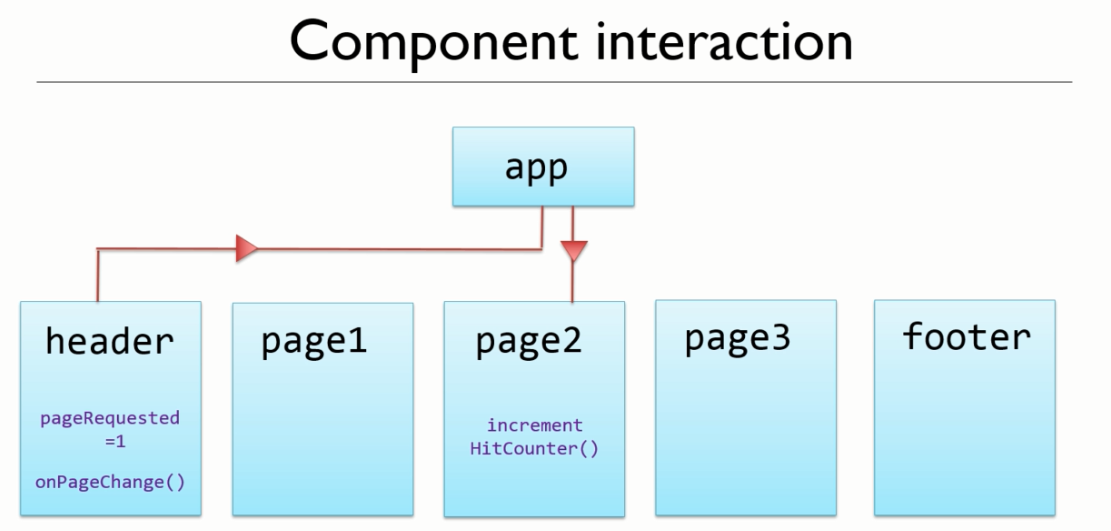
incrementHitCounter(page: number) {

    this.currentPage = page;

    if (page === 2) {

      this.page2Component.incrementHitCounter();

    }



**Typescript Data Types**

number Array

boolean Tuple

string any

symbol object

null void

undefined