1. html

<input [(ngModel)]="name" [ngStyle]="name.length % 2===0? evenStyle : oddStyle">  
<h1>Môn học:{{title1}}</h1>

<input [value]="title1" >

<input [placeholder]="title1">

<div \*ngIf="title1" [ngStyle]="currenStyle">

  time:{{title1}}

</div>

<ng-template #hello1> click on hello</ng-template>

<ng-template #nohello> click on hello1</ng-template>

<div class="circle"></div>

<div [class]="isTrue ? 'circle':'square' "></div> <!--neu true thi tron, nguoc lai vuong-->

<div [class.circle]="!isTrue " [class.square]="isTrue "></div>

<div [ngClass]="curenclass"></div>

<button (click)="isshow=!isshow">isshow</button>

<p \*ngIf="isshow">is show 123</p>

name='';

  isTrue=false;

  title1='chao mung ban den voi angular';

  hello='xin chao';

  currenStyle={color:'blue', fontSize:'50px'};

  evenStyle={color:'red',fontSize:'20px'};

  oddStyle={color:'black',fontSize:'30px'};

  curenclass={circle:!this.isTrue,square:this.isTrue};

  isshow=true;

3. css

.circle{

  height: 50px;

  width: 50px;

  border-radius: 25px; //cắt vào hình vuông thành tròn

  background-color: blue;

}

.square{

  height: 50px;

  width: 50px;

  border-radius: 2px;

  background-color: blue;

}

Bài 18:

<div >

  <ul>

    <li \*ngFor="let subj of arrsubject">

      <p>{{'subject: '+subj}}</p>

    </li>

  </ul>

</div>

Bài 19:

arrsubject=['angylar','react','vu'];

  newen='';

  newviet='';

  arrWords = [

    { id: 1, en: 'action', vn: 'hành động', memorized: true },

    { id: 2, en: 'actor', vn: 'diễn viên', memorized: false },

    { id: 3, en: 'activity', vn: 'hoạt động', memorized: true },

    { id: 4, en: 'active', vn: 'chủ động', memorized: true },

    { id: 5, en: 'bath', vn: 'tắm', memorized: false },

    { id: 6, en: 'bedroom', vn: 'phòng ngủ', memorized: true }

];

  addword(){

    this.arrWords.unshift({

      id:this.arrWords.length+1,

      en:this.newen,

      vn:this.newviet,

      memorized:false

    });

    this.newen='';

    this.newviet='';

    this.isshowform=false;

  }

  isshowform=false;

<div \*ngFor="let word of arrWords">

<h1 [ngStyle]="{color:word.memorized ? 'green':'red'}">

  {{word.en}}

</h1>

<p>{{word.vn}}</p>

</div>

Bài 20:

<div \*ngIf="isshowform">

  <input placeholder="english" [(ngModel)]="newen">

  <br>

  <input placeholder="Vietnam" [(ngModel)]="newviet">

  <br>

  <button (click)="addword()" >Them tu moi</button>

</div>

<button (click)="isshowform = true" \*ngIf="!isshowform">them tu moi 1</button>

Bài 21:

removeword(id:number){

const index=this.arrWords.findIndex(e=>e.id===id);

this.arrWords.splice(index,1)

}

<div \*ngFor="let word of arrWords" class="aword">

<h1 [ngStyle]="{color:word.memorized ? 'green':'red'}">  {{word.en}}</h1>

<p>{{word.vn}}</p>

<button (click)=removeword(word.id)>remove</button>

</div>

.aword{

  background-color: rgb(161, 163, 159);

  padding: 5px;

  margin: 5px;

  width: 150px;

}

Bài 22:

<select name="" id="" [(ngModel)]='filterStatus'>

  <option value='XEM\_TAT\_CA'>xem tất cả</option>

  <option value='XEM\_DA\_NHO'>xem đã nhớ</option>

  <option value='XEM\_DA\_KHONG\_NHO'>xem khong nho</option>

</select>

<div \*ngFor="let word of arrWords" >

  <div class="aword" \*ngIf="filterStatus==='XEM\_TAT\_CA'|| filterStatus==='XEM\_DA\_NHO' && word.memorized || filterStatus==='XEM\_DA\_KHONG\_NHO'&& !word.memorized ">

    <h1 [ngStyle]="{color:word.memorized ? 'green':'red'}">{{word.en}}</h1>

    <p>{{word.vn}}</p>

filterStatus='XEM\_TAT\_CA';

GETSHOWStatus(memorized:boolean){

const XEMTATCA=this.filterStatus==='XEM\_TAT\_CA';

const XEMCONHO=this.filterStatus==='XEM\_DA\_NHO'&& memorized;

const XEMKONHO=this.filterStatus==='XEM\_DA\_KHONG\_NHO'&& !memorized;

return XEMTATCA || XEMCONHO||XEMKONHO;

}

<div class="aword" \*ngIf="GETSHOWStatus(word.memorized) ">

Bài 23:

<div \*ngFor="let word of arrWords" >

  <div class="aword" \*ngIf="GETSHOWStatus(word.memorized) ">

    <h1 [ngStyle]="{color:word.memorized ? 'green':'red'}">{{word.en}}</h1>

    <p>{{word.vn}}</p>

    <button (click)=removeword(word.id)>remove</button>

    <button (click)="word.memorized=!word.memorized">{{word.memorized?'chưa thuộc':'đã thu'}}</button>

  </div>

</div>

Chỉ thêm dòng trạng thái này vào code trên như hình

<button (click)="word.memorized=!word.memorized">{{word.memorized?'chưa thuộc':'đã thu'}}</button>

Bài 24:

Khai báo biến tên trong app.component.ts

public bienten: String="";

Đưa ten input.ts và bienten từ app.ts vào trong app.component.html

<app-input [ten]="bienten"></app-input>

Thêm input và @input vào trong input.component.ts

import { Component, OnInit, Input } from '@angular/core';

thêm tên biến nhận từ app.component.html

@Input ('ten') name:string="";

Nếu truyền trực tiếp không cần khai báo bienten trong app.ts thì câu lệnh sẽ như thế này

<app-input ten="anh em "></app-input>

Input.html lấy tên biến name thể hiện ra màn hình

{{name}}

Bài 25:

App.ts

 arrperson=[

    {name:'Tí', age:'18'},

    {name:'Tèo', age:'15'},

    {name:'Tuấn', age:'14'}

  ];

App.html

<div \*ngFor="let item of arrperson">

<app-input [ten]="item.name" [tuoi]="item.age"></app-input>

</div>

Input.ts

@Input ('ten') names:string="";

@Input ('tuoi') ages:string="";

Input.html

{{names}}

{{ages}}

Bài 26, 27:

Trong ts cho biến value=0;

<p>parent works!</p>

{{value}}

<app-child (myclick)="value = value + 1 "></app-child>

<p>child works!</p>

<button (click)="addforparent()">add</button>

Import eventemitter và output vào và @output() và hàm click

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

@Component({

  selector: 'app-child',

  templateUrl: './child.component.html',

  styleUrls: ['./child.component.css']

})

export class ChildComponent implements OnInit {

  @Output() myclick=new EventEmitter();

  addforparent(){

    this.myclick.emit();

  }

Bài 28 out có tham số:

(myclick)="value = value + 1 " thành

<app-child (myclick)="changevalue() "></app-child>

Khai báo lại trong parent ts

  changevalue(){

    this.value = this.value + 1;

  }

Cách 2:

<app-child (myclick)="changevalue(false) "></app-child>

 changevalue(isadd:boolean){

    if(isadd)

    {

      this.value = this.value + 1;

    }

    else

    {

      this.value=this.value-1;

    }

  }

Cách 3:

<p>child works!</p>

<button (click)="addforparent()">add</button>

<button (click)="subforparent()">add</button>

 @Output() myclick=new EventEmitter <boolean>();

  addforparent(){

    this.myclick.emit(true);

  }

  subforparent(){

    this.myclick.emit(false);

  }

<p>parent works!</p>

{{value}}

<app-child (myclick)="changevalue($event) "></app-child>

 value=0;

  changevalue(isadd:boolean){

    if(isadd)

    {

      this.value = this.value + 1;

    }

    else

    {

      this.value=this.value-1;

    }

  }

Bài 29: bài tập

Bài 30: viewchild

Cách 1:

<p>parent works!</p>

<button (click)="child.value=child.value+1" >add of child</button>

<app-child #child></app-child>

Khai báo biến value tại child.ts

<p>child works!</p>

{{value}}

Cách 2:

<p>parent works!</p>

<button (click)="addchild()">add of child</button>

<app-child ></app-child>

Parent.ts

import { Component, OnInit,ViewChild } from '@angular/core';

import {ChildComponent} from '../child/child.component';

export class ParentComponent implements OnInit {

  @ViewChild(ChildComponent)  mychild: ChildComponent =new ChildComponent;

addchild(){

this.mychild.value++;

}

<p>child works!</p>

{{value}}

Và khai báo biến value=0; trong child.ts