

ปฏิบัติการที่ 3

คะแนน

วิชา 523419 Advanced Web Application Development

รหัสนักศึกษา..... ชื่อ.....

วัตถุประสงค์

ไฟล์ที่จำเป็น

แบบฝึกปฏิบัติการที่ 3.1 Parent to Child via @Input

1. สร้างแอปพลิเคชันใหม่ด้วยคำสั่ง “ng new lab3-angular-app”
2. ติดตั้ง Bootstrap CSS Framework ด้วยการพิมพ์คำสั่ง “npm install bootstrap@4.6”
3. ติดตั้ง jQuery ด้วยการพิมพ์คำสั่ง “npm install jquery popper.js”
4. สร้าง Component ใหม่ ได้แก่ parent และ child โดยกรอกคำสั่งต่อไปนี้
 - a. “ng g c components/parent”
 - b. “ng g c components/child/child”
5. แก้ไขไฟล์ app.component.html โดยกำหนด sub-component ต่าง ๆ ที่อยู่ภายใต้ root component พร้อมกับแก้ไขไฟล์เทมเพลต parent.component.html และ child.component.html กรอกโค้ดต่อไปนี้ลงไป

```
parent.component.html
lab3-angular-app > src > app > components > parent > < parent.component.html
1 <div class="container p-3 my-3 bg-primary text-white" style="margin-top:60px">
2 <h2>Parent Component</h2>
3 <div class="row">
4 <div class="col-md-6">
5 <div class="form-group">
6 <input type="text" class="form-control">
7 <small id="helpId" class="form-text text-white">Help text</small>
8 </div>
9 </div>
10 <div class="col-md-6">Counter : </div>
11 </div>
12 <div class="row">
13 <div class="col-md-6">
14 <button type="button" class="btn btn-success">Send via @Input</button>
15 <button type="button" class="btn btn-danger">Send via @ViewChild</button>
16 </div>
17 <div class="col-md-6">
18 <button type="button" class="btn btn-light">Show Child</button>
19 </div>
20 </div>
21 <div class="row">
22 <div class="col-md-12">
23 <app-child></app-child>
24 </div>
25 </div>
26 </div>
```

```
child.component.html
lab3-angular-app > src > app > components > child > < child.component.html
1 <div class="container p-3 my-3 bg-warning text-white" style="margin-top:60px">
2 <h3>Child Component</h3>
3 <div class="row">
4 <div class="col-md-6">
5 Counter :
6 </div>
7 <div class="col-md-6">
8 <button type="button" class="btn btn-light">Send via @Output</button>
9 </div>
10 </div>
11 </div>
```

```
app.component.html
lab3-angular-app > src > app > < app.component.html
1 <app-parent></app-parent>
2
```

6. แก้ไขไฟล์ parent.component.ts โดยกำหนดพรีอเพอร์ติ parentData และฟังก์ชัน onClick2Child() ดังต่อไปนี้

```
9 export class ParentComponent implements OnInit {
10
11   parentData: number;
12
13   constructor() { }
14
15   ngOnInit(): void {
16     this.parentData = 0;
17   }
18
19   onClick2Child(){
20     this.parentData += 1;
21   }
22
23 }
```

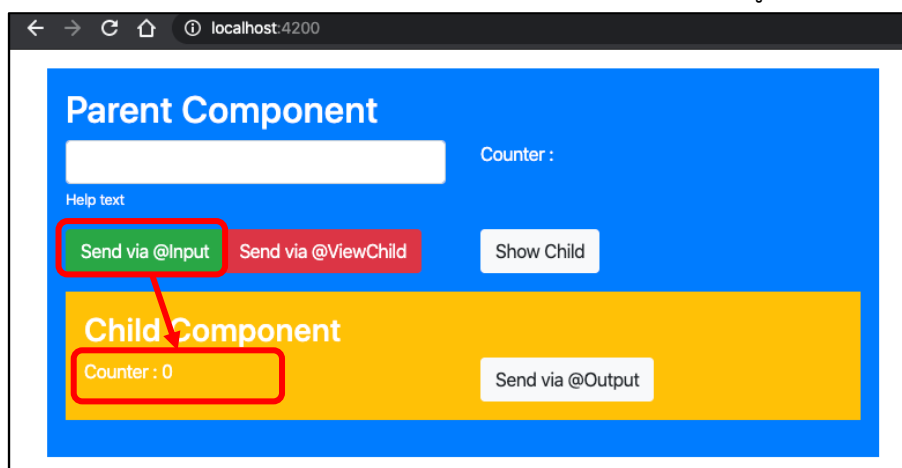
7. แก้ไขไฟล์ parent.component.html เพื่อส่งค่าพร็อพเพอร์ตี้ parentData ไปให้กับพร็อพเพอร์ตี้ parentData ซึ่งอยู่ที่ child component ดังต่อไปนี้

```
12 <div class="row">
13   <div class="col-md-6">
14     <button type="button" (click)="onClick2Child()"
15       class="btn btn-success">Send via @Input</button>
16     <button type="button" class="btn btn-danger">Send via @ViewChild</button>
17   </div>
18   <div class="col-md-6">
19     <button type="button" class="btn btn-light">Show Child</button>
20   </div>
21 </div>
22 <div class="row">
23   <div class="col-md-12">
24     <app-child [parentData]="parentData" ></app-child>
25   </div>
26 </div>
```

8. แก้ไขไฟล์ child.component.ts โดยกำหนด @Input หน้าพร็อพเพอร์ตี้ parentData เพื่อบอกให้ทราบว่าการรับค่าจาก parent component ดังต่อไปนี้

```
TS child.component.ts X
lab3-angular-app > src > app > components > child > TS child.component.ts > ...
1 import { Component, OnInit, Input } from '@angular/core';
2
3 @Component({
4   selector: 'app-child',
5   templateUrl: './child.component.html',
6   styleUrls: ['./child.component.css']
7 })
8 export class ChildComponent implements OnInit {
9
10   @Input() parentData: number;
11
12   constructor() { }
13
14   ngOnInit(): void {
15
16   }
17
18 }
```

9. แก้ไขไฟล์ child.component.html โดยทำ Data Binding กับพร็อพเพอร์ตี้ parentData เพื่อแสดงผลลัพธ์ที่หน้า view ของ child component
10. หน้าต่าง Command Prompt ทดสอบการทำงานของแอปพลิเคชันด้วยการพิมพ์คำสั่ง “ng serve --open”
11. เปิดเบราว์เซอร์ จากนั้นกรอก url คือ “http://localhost:4200” จะปรากฏที่หน้าเว็บเพจ โดยคลิกที่ “Send via @Input” จะแสดงผลลัพธ์ของการนับเลขที่ child component ดังรูปต่อไปนี้



แบบฝึกปฏิบัติการที่ 3.2 Child to Parent via @Output

1. แก้ไขไฟล์ child.component.ts โดยจะต้องกำหนด @Output เพื่อบอกว่านี่คือข้อมูลที่ถูกส่งออกไปจาก child component และใช้ EventEmitter เพื่อสร้างอีเวนต์สำหรับใช้ส่งข้อมูลออกไป และจะทำงานเมื่อมีการเรียกฟังก์ชัน onClick2Parent() ดังต่อไปนี้

```
TS child.component.ts •
lab3-angular-app > src > app > components > child > TS child.component.ts > ...
1  import { Component, OnInit,
2     Input, Output, EventEmitter } from '@angular/core';
3
4  @Component({
5    selector: 'app-child',
6    templateUrl: './child.component.html',
7    styleUrls: ['./child.component.css']
8  })
9  export class ChildComponent implements OnInit {
10
11    @Input() parentData: number;
12
13    @Output() messageEvent = new EventEmitter<number>();
14
15    childData: number;
16
17    constructor() { }
18
19    ngOnInit(): void {
20      this.childData = 0;
21    }
22
23    onClick2Parent(){
24      this.childData += 1;
25      this.messageEvent.emit(this.childData)
26    }
27  }
```

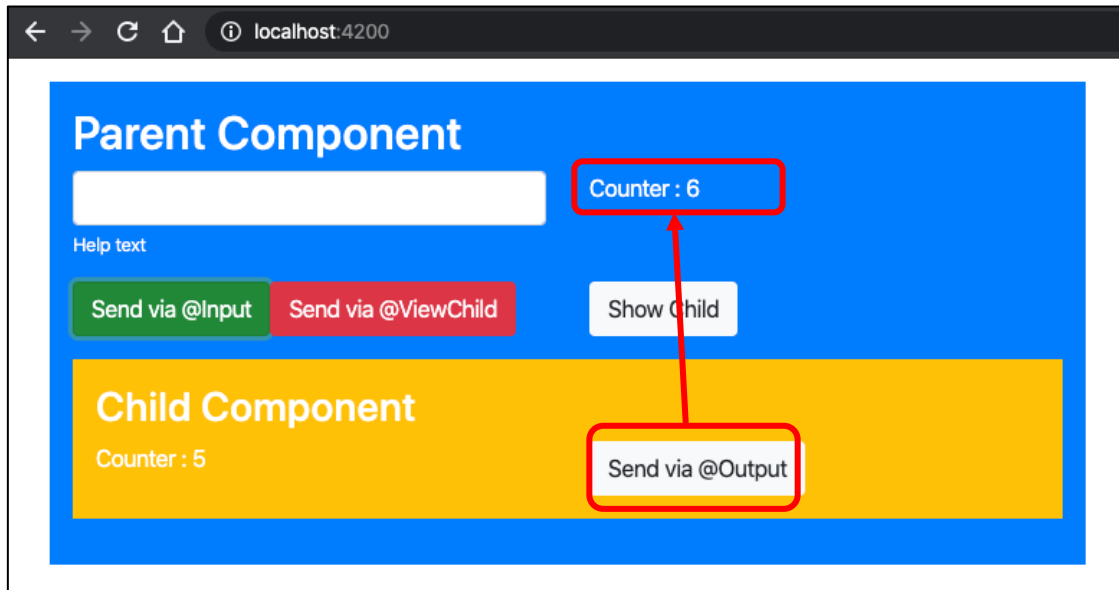
2. แก้ไขไฟล์ parent.component.html โดยเมื่ออีเวนต์ messageEvent เกิดขึ้นจะเรียกใช้เมธอด receiveData (\$event) ดังต่อไปนี้

```
22  <div class="row">
23    <div class="col-md-12">
24      <app-child [parentData]="parentData" (messageEvent)="receiveData($event)"></app-child>
25    </div>
26  </div>
```

3. แก้ไขไฟล์ parent.component.ts เพื่อสร้างเมธอด receiveData (\$event) โดยข้อมูลที่ส่งมาจาก child component จะอยู่ใน \$event ดังต่อไปนี้

```
9  export class ParentComponent implements OnInit {
10
11    parentData: number;
12    childData: number;
13
14    constructor() { }
15
16    ngOnInit(): void {
17      this.parentData = 0;
18      this.childData = 0;
19    }
20
21    onClick2Child(){
22      this.parentData += 1;
23    }
24
25    receiveData($event) {
26      this.childData = $event;
27    }
28
29  }
```

- เปิดเบราว์เซอร์ จากนั้นกรอก url คือ “http://localhost:4200” จะปรากฏที่หน้าเว็บเพจ โดยคลิกที่ “Send via @Output” จะแสดงผลลัพธ์ของการนับเลขที่ parent component ดังรูปต่อไปนี้



แบบฝึกปฏิบัติการที่ 3.3 Child to Parent via @ViewChild

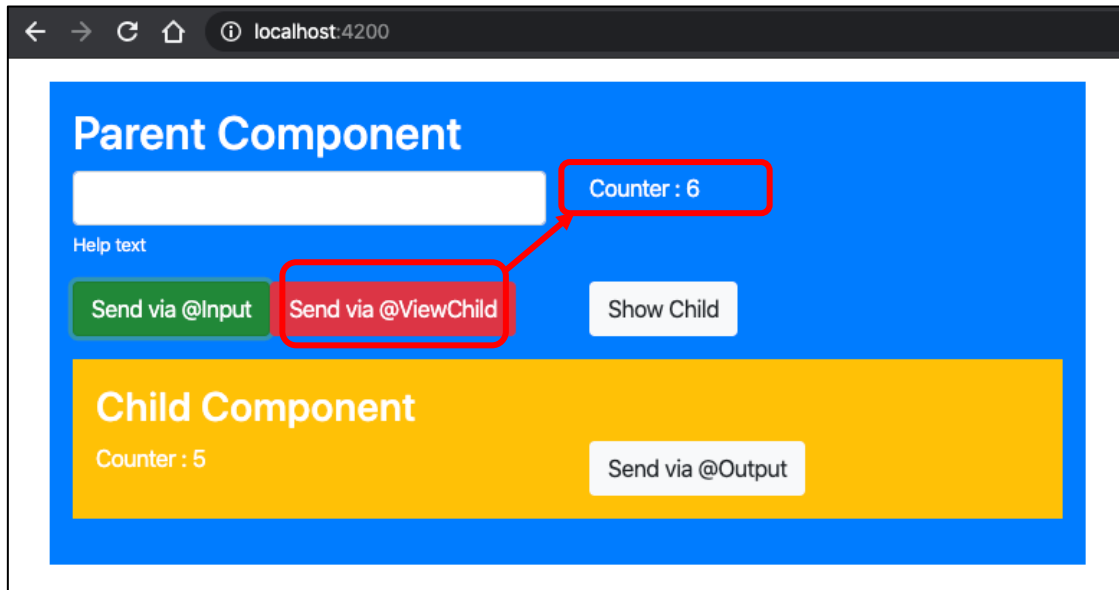
- แก้ไขไฟล์ parent.component.ts โดยจะต้องกำหนด @ViewChild(ChildComponent) เพื่อเป็นการที่จะเข้าถึงเมธอดต่าง ๆ หรือพเพอร์ตีต่าง ๆ ของ child component และจะทำงานเมื่อมีการเรียกฟังก์ชัน onClickViewChild() ดังต่อไปนี้

```
TS parent.component.ts ×
lab3-angular-app > src > app > components > parent > TS parent.component.ts > ParentComponent >
1  import { Component, OnInit, ViewChild } from '@angular/core';
2  import { ChildComponent } from '../child/child.component'
3
4  @Component({
5    selector: 'app-parent',
6    templateUrl: './parent.component.html',
7    styleUrls: ['./parent.component.css']
8  })
9
10 export class ParentComponent implements OnInit {
11
12   @ViewChild(ChildComponent)
13   childComponent: ChildComponent;
14
15   parentData: number;
16   childData: number;
```

- ที่ไฟล์ parent.component.ts สร้างฟังก์ชัน onClickViewChild() เพื่อเรียกใช้งานฟังก์ชัน onClick2Parent() ที่อยู่ที่ child component ดังต่อไปนี้

```
onClickViewChild(){
  this.childComponent.onClick2Parent();
}
```

- เปิดเบราว์เซอร์ จากนั้นกรอก url คือ “http://localhost:4200” จะปรากฏที่หน้าเว็บเพจ โดยคลิกที่ “Send via @ViewChild” จะแสดงผลพัทธ์ของการนับเลขที่ parent component ดังรูปต่อไปนี้



แบบฝึกปฏิบัติที่ 3.4 Lifecycle Hook

- แก้ไขไฟล์ parent.component.ts และ child.component.ts เพื่อเรียกใช้งาน Lifecycle Hook ในแต่ละเมธอด โดยส่วนแรกทำการ import ดังต่อไปนี้

```
TS parent.component.ts x
lab3-angular-app > src > app > components > parent > TS parent.component.ts > ParentComponent
1 import { Component, ViewChild, OnChanges, OnInit,
2   DoCheck, AfterContentInit, AfterContentChecked,
3   AfterViewInit, AfterViewChecked, OnDestroy } from '@angular/core';
4
5 import { ChildComponent } from '../child/child.component'
6
7 @Component({
8   selector: 'app-parent',
9   templateUrl: './parent.component.html',
10  styleUrls: ['./parent.component.css']
11 })
12
13 export class ParentComponent implements OnChanges,
14   OnInit,
15   DoCheck,
16   AfterContentInit,
17   AfterContentChecked,
18   AfterViewInit,
19   AfterViewChecked,
20   OnDestroy {
  TS child.component.ts x
lab3-angular-app > src > app > components > child > TS child.component.ts > ChildComponent
1 import { Component, Input, Output, EventEmitter, OnInit,
2   DoCheck, AfterContentInit, AfterContentChecked,
3   AfterViewInit, AfterViewChecked, OnDestroy, OnChanges } from '@angular/core';
4
5 @Component({
6   selector: 'app-child',
7   templateUrl: './child.component.html',
8   styleUrls: ['./child.component.css']
9 })
10 export class ChildComponent implements
11   OnChanges,
12   OnInit,
13   DoCheck,
14   AfterContentInit,
15   AfterContentChecked,
16   AfterViewInit,
17   AfterViewChecked,
18   OnDestroy {
```

- เรียกใช้งาน Lifecycle Hook ในแต่ละเมธอดทั้งไฟล์ parent.component.ts และ child.component.ts ดังต่อไปนี้

```
TS parent.component.ts •
lab3-angular-app > src > app > components > parent > TS parent.component.ts
28 constructor() { console.log('Constructor Work!'); }
29 ngOnInit(): void {
30   this.parentData = 0;
31   this.childData = 0;
32   console.log('ngOnInit Work!');
33 }
34
35 ngOnChanges() {
36   console.log('ngOnChanges Work!');
37 }
38
39 ngDoCheck() {
40   console.log('ngDoCheck Work!');
41 }
42 ngAfterContentInit() {
43   console.log('ngAfterContentInit Work!');
44 }
45 ngAfterContentChecked() {
46   console.log('ngAfterContentChecked Work!');
47 }
48
49 ngAfterViewInit() {
50   console.log('ngAfterViewInit Work!');
51 }
52 ngAfterViewChecked() {
53   console.log('ngAfterViewChecked Work!');
54 }
55 ngOnDestroy() {
56   console.log('ngOnDestroy Work!');
57 }

TS child.component.ts ×
lab3-angular-app > src > app > components > child > TS child.component.ts
25
26 constructor() { console.log('Constructor Work!'); }
27
28 ngOnChanges() {
29   console.log('ngOnChanges Work!');
30 }
31
32 ngOnInit(): void {
33   this.childData = 0;
34   console.log('ngOnInit Work!');
35 }
36
37 ngDoCheck() {
38   console.log('ngDoCheck Work!');
39 }
40 ngAfterContentInit() {
41   console.log('ngAfterContentInit Work!');
42 }
43 ngAfterContentChecked() {
44   console.log('ngAfterContentChecked Work!');
45 }
46 ngAfterViewInit() {
47   console.log('ngAfterViewInit Work!');
48 }
49 ngAfterViewChecked() {
50   console.log('ngAfterViewChecked Work!');
51 }
52 ngOnDestroy() {
53   console.log('ngOnDestroy Work!');
54 }
```

- สังเกตผลลัพธ์โดยดูลำดับการทำงานของแต่ละฟังก์ชันที่ Console ของเว็บเบราว์เซอร์
- แก้ไขไฟล์ parent.component.html เพื่อทดสอบการทำงานของ Lifecycle Hook ดังต่อไปนี้

```
<div class="col-md-6">
  <div class="form-group">
    <input type="text" [(ngModel)]="textMessage" class="form-control">
    <small id="helpId" class="form-text text-white">text status : {{status}}</small>
  </div>
</div>
```

- แก้ไขไฟล์ parent.component.ts เพื่อแก้ไขฟังก์ชัน ngDoCheck() และ ngAfterContentChecked() ดังต่อไปนี้

```
43 ngDoCheck() {
44   console.log(this.textMessage);
45   console.log('ngDoCheck Work!');
46 }
47 ngAfterContentInit() {
48   console.log('ngAfterContentInit Work!');
49 }
50 ngAfterContentChecked() {
51   if(this.textMessage == "B5111299"){
52     this.status = "Your ID is OK";
53   }else{
54     this.status = "Error";
55   }
56   console.log('ngAfterContentChecked Work!');
57 }
```

- สังเกตผลลัพธ์โดยดูลำดับการทำงานของแต่ละฟังก์ชันที่ Console ของเว็บเบราว์เซอร์

7. แก้ไขไฟล์ parent.component.html เพื่อทดสอบการทำงานของ ngOnDestroy() ดังต่อไปนี้

```
19     <div class="col-md-6">
20         <button type="button" [(click)]=update() class="btn btn-light">Show Child</button>
21     </div>
22 </div>
23 <div class="row">
24     <div class="col-md-12">
25         <app-child *ngIf='showChild' [parentData]="parentData"
26             (messageEvent)="receiveData($event)"></app-child>
27     </div>
28 </div>
29 </div>
30
```

8. แก้ไขไฟล์ parent.component.ts กำหนดพรีอเพอร์เตอร์ที่ public showChild = true และทำการสร้างฟังก์ชัน update() ดังต่อไปนี้

```
81
82     update(){
83         this.showChild = !this.showChild
84     }
85
```

9. สังเกตผลลัพธ์โดยดูลำดับการทำงานของแต่ละฟังก์ชันที่ Console ของเว็บเบราว์เซอร์ เมื่อคลิกปุ่ม Show Child