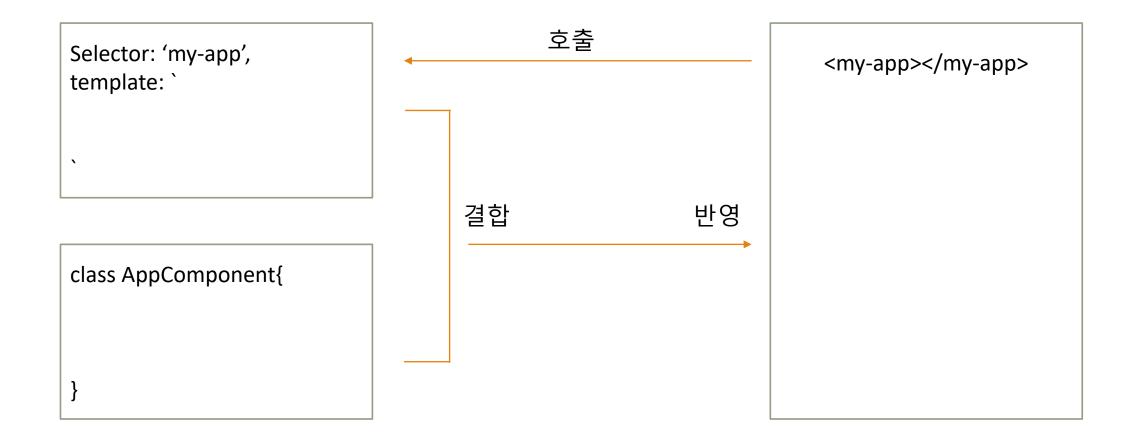
Angular2 Component

Component



Nested Component

```
Selector: 'my-app',
                                             1. 호출
                                                                        <my-app></my-app>
template: `
    <child></child>
                                                           반영
class AppComponent{
                                     2. 호출
Selector: 'child',
template: `
class ChildComponent{
```

Component 값 전달

- 1. Component 밖에서 들어오는 데이터는 프로퍼티로 바인딩.
- 2. 누가 전달하는지는 알 필요가 없으며, 어떤 형식의 데이터인지만 알면 된다.
- 3. 밖으로 데이터를 전달할 경우는 이벤트를 이용
- 4. 밖으로 전달할 경우도 어디로 전달되는지 알 필요 가 없다. 받는 쪽에서 이벤트를 통해 받음

Component 값 전달

```
2. 값 주입
              Selector: 'child',
                                                             <child
                                             변수 OR
              Template: `
                                                                  [name]='name'
                                             문자열 형태
                                                                  [age]='age'
                   <div>{{name}}</div>
                                                                  add='Seoul'>
                   <div>{{age}}</div>
                                                             </child>
                   <div>{{add}}</div>
3. 대입
                                                                                                   1. 전달
              Class ChildComponent{
                                                             class AppComponent{
                                                                  age: number = 20;
                   @Input() name;
                                                                  name: string = 'Hong';
                   @Input() age;
                   @Input() add;
```

Component 값 전달

```
Selector: 'child',
         Template: `
                                                                      <child
             <but
                                                                          (outObj)='getValue($event)'
                                                                      </child>
         (click)='sendObj()'>Click<button>
                                                                                              4. Event Catch
                                 1. Event 객체 생성
         Class ChildComponent{
                                                                                                 및 처리
             @Output() outObj: EventEmitter<any>
                                                                      class AppComponent{
                  = new EventEmitter<any>();
                                                                          getValue(value: Type) {
                                                     3. Event 전파
2. Event 발생
                                                                               변수 = value
             sendObj(): void {
                  outObj.emit( value );
```