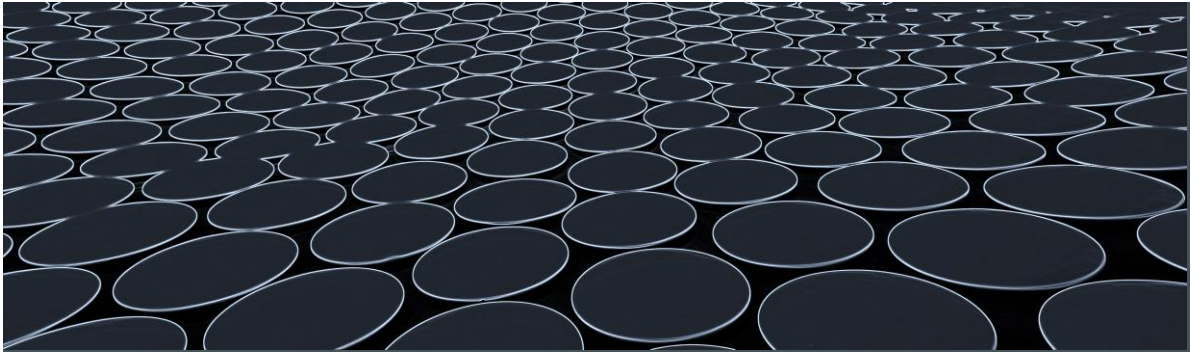




## #5 – COMUNICAZIONE TRA COMPONENTI

ANDREA RIZZIERI



1

## INDICE DEGLI ARGOMENTI

- Componenti annidati
- Comunicazioni in ingresso
- Comunicazioni in uscita

Andrea Rizzieri ©

2

## INDICE DEGLI ARGOMENTI

- Componenti annidati
- Comunicazioni in ingresso
- Comunicazioni in uscita

Andrea Rizzieri ©

3

## COMPONENTI ANNIDATI

- Annidare i componenti
  - Un componente si dice annidato (nested) quando il suo selettore è contenuto nel template HTML di un altro componente.
  - Il componente interno si dice elemento figlio (child element) mentre l'elemento contenitore è detto elemento padre o genitore (parent element)

**parent.component.html:**

...

**<child-component></child-component>**

**<child-component></child-component>**

**<child-component></child-component>**

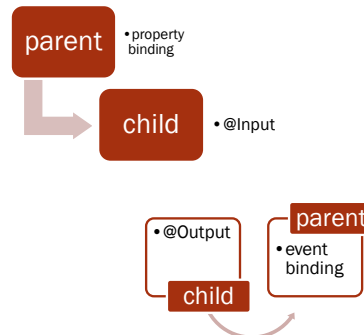
...

Andrea Rizzieri ©

4

## COMPONENTI ANNIDATI

- Annidare i componenti
  - Quando un componente è contenuto in un altro, per poter passare delle informazioni da padre al figlio e viceversa bisogna utilizzare particolari strategie



Andrea Rizzieri ©

5

## INDICE DEGLI ARGOMENTI

- Componenti annidati
- Comunicazioni in ingresso
- Comunicazioni in uscita

Andrea Rizzieri ©

6

## COMUNICAZIONI IN INGRESSO

- @Input()
  - All'interno di un componente è possibile ricevere dei dati dall'esterno utilizzando il decoratore @Input()
  - Questo decoratore mappa un valore che proviene dall'esterno (dal componente contenitore)
  - Nel componente contenitore possiamo inviare il valore all'interno del componente mediante property binding

```
export class ChildComponent {
```

```
  @Input() in = ...;
```

```
  ...
```

```
}
```

parent.component.html:

```
<child-component [in] = "valore">
```

```
</child>
```

Andrea Rizzieri ©

## INDICE DEGLI ARGOMENTI

- Componenti annidati
- Comunicazioni in ingresso
- **Comunicazioni in uscita**

Andrea Rizzieri ©

## COMUNICAZIONI IN USCITA

- `@Output()`
  - Il componente figlio, per poter inviare dei dati all'esterno dovrà usare il decoratore `@Output()`
  - Questo decoratore mappa un valore che dall'interno del componente verrà emesso verso l'esterno
  - Per emettere il valore è necessario utilizzare un emettitore di eventi, implementato mediante la classe `EventEmitter`
  - Nel componente contenitore possiamo restare in ascolto dell'evento che verrà generato dal figlio e catturare così il valore passato

```
export class ChildComponent {
  @Output() out = new EventEmitter();

  ...

  evento(){
    out.emit(valore);
  }
}

parent.component.html:
<child-component (out)="metodo($event)" >
</child>
```

Andrea Rizzieri ©