

## 5. INTERAZIONE

INPUT, comp cerca

```
<app-ricerca-risultati></app-ricerca-risultati>  
<app-visualizza [articolo]="articolo"></app-visualizza>
```



figlio



padre

```
articoloDao: ArticoloDaoService;  
articolo: Articolo;  
constructor(private dao: ArticoloDaoService) {  
    this.articoloDao = dao;  
}  
ngOnInit() {  
    this.articolo = this.articoloDao.getArticoloHome();  
}
```

## 5. INTERAZIONE

INPUT, comp visualizza

```
import { Input } from '@angular/core';
...
articoloDao: ArticoloDaoService;
@Input() articolo: Articolo; // @Input('art') indica art lato
    // figlio: [art]="articolo"
...
constructor(private dao: ArticoloDaoService) {
    this.articoloDao = dao;
    // this.articolo = null; // nessun effetto
}
ngOnInit() {
    // this.articolo = null; // annulla l'input passato dal padre
    this.articolo = this.articolo ?
        this.articolo : this.articoloDao.getArticoloHome();
}
```

L'Input setta il valore dopo il costruttore ma prima dell'init.

## 5. INTERAZIONE

OUTPUT, service articolo dao (aggiungere almeno un secondo articolo)

```
getArrayArticoliFiltrati(filtro: FiltroArticoloType): Array<Articolo> {  
  const arrayArticoliFiltrati: Array<Articolo> = [];  
  const filtroLower: string = filtro.testo.toLowerCase();  
  for (const articolo of this.arrayArticoli) {  
    if (articolo.titolo.toLowerCase().includes(filtroLower)) {  
      arrayArticoliFiltrati.push(articolo);  
    } else if (articolo.arraySottotitolo[1].toLowerCase().includes(filtroLower)) {  
      arrayArticoliFiltrati.push(articolo);  
    } else if (articolo.testo.toLowerCase().includes(filtroLower)) {  
      arrayArticoliFiltrati.push(articolo);  
    }  
  }  
  return arrayArticoliFiltrati;  
}
```

## 5. INTERAZIONE

OUTPUT, comp ricerca-risultati, visualizza articoli filtrati

```
<br /><br />
<input type="text" name="ricerca" placeholder="cerca tra gli articoli"
  [(ngModel)]="ricercaModel" (ngModelChange)="cerca()">
<ul>
  <li *ngFor="let articolo of arrayArticoliFiltrati">
    <h3 style="margin: 0;">{{articolo.titolo}}</h3>
    <h5 style="margin: 0;">{{articolo.arraySottotitolo[1]}}</h5>
  </li>
</ul>
```

## 5. INTERAZIONE

OUTPUT, comp ricerca-risultati, filtra articoli

```
_ricercaModel: string = 'git';
articoloDao: ArticoloDaoService;
arrayArticoliFiltrati: Array<Articolo>;
constructor(private dao: ArticoloDaoService) {
    this.articoloDao = dao;
}
ngOnInit() { }
cerca() {
    const filtro: FiltroArticoloType = { testo: this._ricercaModel };
    this.arrayArticoliFiltrati =
        this.articoloDao.getArrayArticoliFiltrati(filtro);
}
```

## 5. INTERAZIONE

OUTPUT, comp ricerca-risultati, seleziona articolo

```
<li *ngFor="let articolo of arrayArticoliFiltrati" (click)="selezionaArticolo(articolo)">
```

```
selezionaArticolo(articolo: Articolo) {  
    console.log(articolo.titolo);  
}
```

OUTPUT & EVENT EMITTER, comp ricerca-risultati

```
@Output() articoloSelezionato: EventEmitter<Articolo> = new EventEmitter<Articolo>();
```

```
selezionaArticolo(articolo: Articolo) {  
    // console.log(articolo.titolo);  
    this.articoloSelezionato.emit(articolo);  
}
```

## 5. INTERAZIONE

OUTPUT, comp cerca

```
<app-ricerca-risultati (articoloSelezionato)="visualizzaArticolo($event)">  
</app-ricerca-risultati>
```



figlio



padre

```
visualizzaArticolo(articolo: Articolo) {  
  this.articolo = articolo;  
}
```

## 5. INTERAZIONE

VIEW CHILD, comp visualizza

```
visualizzaArticolo(articolo: Articolo) {  
    this.articolo = articolo;  
}
```

comp cerca

```
@ViewChild(VisualizzaComponent, { static: true })  
    visualizza: VisualizzaComponent;  
visualizzaArticolo(articolo: Articolo) {  
    // this.articolo = articolo;  
    this.visualizza.visualizzaArticolo(articolo);  
}
```



## 5. INTERAZIONE

NG CONTENT, comp cerca

```
<app-visualizza [articolo]="articolo"><hr></app-visualizza>
```

comp visualizza

```
<ng-content></ng-content>
```