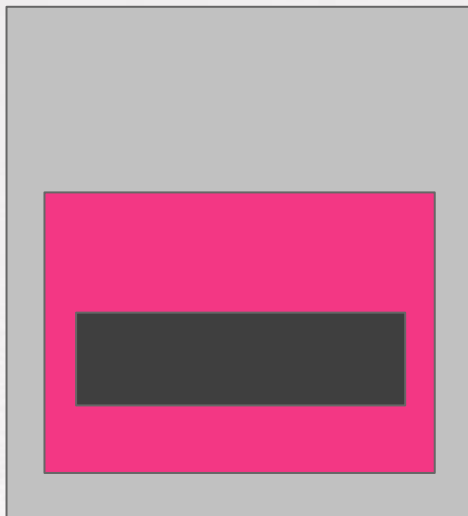
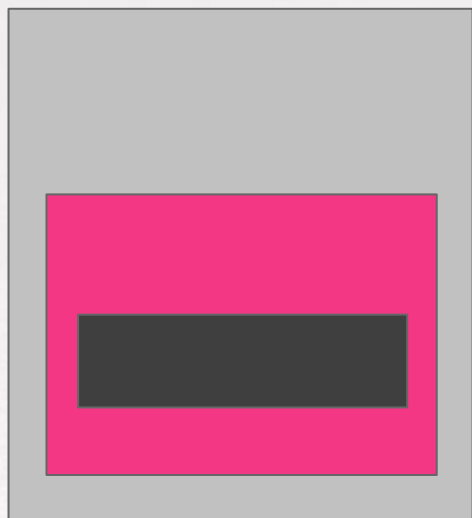


Comunicación Padre → Hijo (Data Binding)

Los componentes, al igual que los elementos html, tienen una jerarquía



Los componentes, al igual que los elementos html, tienen una jerarquía



Componente

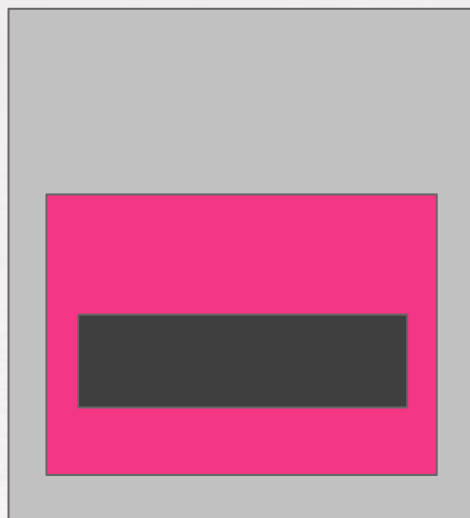
Padre de



Abuelo de



Los componentes, al igual que los elementos html, tienen una jerarquía

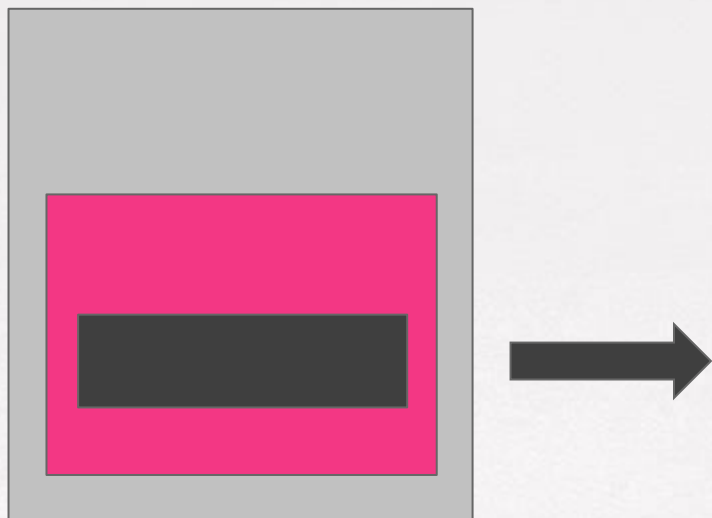


Componente

Padre de
Hijo de



Los componentes, al igual que los elementos html, tienen una jerarquía



Componente

Hijo de



Nieto de



Comunicación Padre → Hijo

- Preparación del componente hijo
- Comunicación

Preparación del componente hijo

- Componente hijo debe disponer de variables
- Para usar un “template” para transmitir la información, necesitamos indicarlo en nuestro componente hijo

Preparación del componente hijo

- Para indicar que una variable quiere usarse en el “template”
- Usamos el decorador `@Input()`
- Un **decorador** es una herramienta de Angular para indicar alguna configuración a tener en cuenta de forma dinámica (como al definir un componente con `@Component`)

Preparación del componente hijo

- Declaramos una variable con el decorador delante, en nuestro componente hijo, para indicar qué esperamos recibir

@Input() variable: tipo;

Preparación del componente hijo

- Declaramos una variable o método en nuestro componente hijo para poder acceder desde el padre

variable: tipo;

funcionDelHijo() {}

Preparación del componente hijo

Documentación completa:

@Input: <https://angular.io/api/core/Input>

Comunicación Padre → Hijo

- Preparación del componente hijo
- **Comunicación**

Formas de comunicación Padre → Hijo

- Usando el “template” con el propio HTML
- Usando TypeScript en el padre

En “template”

Vamos a editar la variable título de nuestro componente hijo, desde el “template” del componente padre:

```
<componente-hijo></componente-hijo>
```

En “template”

Usando el valor directamente:

```
<componente-hijo  
  titulo="Mi Titulo">  
</componente-hijo>
```

En “template”

Ahora en lugar de un valor, le indicaremos lo mismo pero usando una variable:

```
<componente-hijo  
  [titulo]="variableTitulo">  
</componente-hijo>
```


Mediante TypeScript

Lo usamos cuando:

- Queremos o necesitamos usar TypeScript para ello
- Necesitamos controlar el momento de enviar la información

Mediante TypeScript

Necesitamos una variable en el padre para poder acceder al hijo; para ello:

- Usamos el decorador `@ViewChild(selector)`
- Usamos el `selector` del componente hijo

`<componente-hijo></componente-hijo>`

Mediante TypeScript

Creamos la variable en el componente padre para poder acceder al hijo:

```
@ViewChild(componente-hijo)
```

```
variableComponenteHijo: HijoComponent;
```

Mediante TypeScript

Ya podemos editar las variables en el componente hijo, con una función por ejemplo:

```
enviarTituloAlHijo(){  
  this.componenteHijo.titulo = "Mi Titulo"  
}
```

Mediante TypeScript

También podemos ejecutar métodos del componente hijo:

```
enviarTituloAlHijo(){  
  this.componenteHijo.funcionDelHijo();  
}
```

¡Gracias!