



```
left: 50%;  
transform: translate(-50%,  
width: 400px;  
padding: 40px;  
background: rgba(0, 0, 0,  
box-sizing: border-box;  
box-shadow: 0 15px 25px rgba(0, 0, 0,  
border-radius: 10px;  
  
h2{  
margin: 0 0 30px;  
padding: 0;  
color: #fff;  
text-align: center;  
  
.box h3{  
margin: 0 0 10px;  
padding: 0;  
color: #fff;  
text-align: center;  
}  
.box .inputP
```

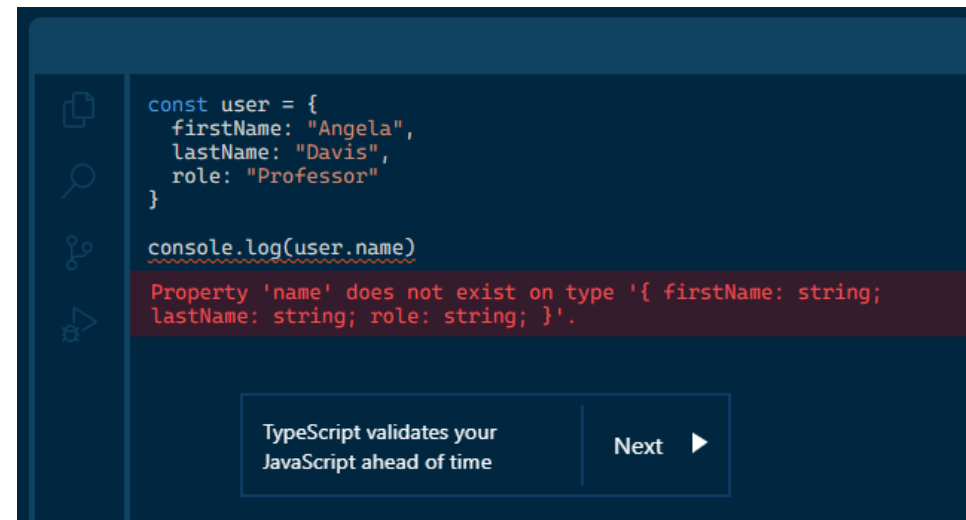


Angular

TypeScript

- Es un lenguaje de programación creado por Microsoft.
- Es un superset de JavaScript.
- Añade el tipado estático y los objetos mediante clases.

- ✓ Validación de JS en tiempo de codificación.
- ✓ Proporciona el autocompletado.
- ✓ Describe los objetos y funciones.
- ✓ Enriquece JSX.

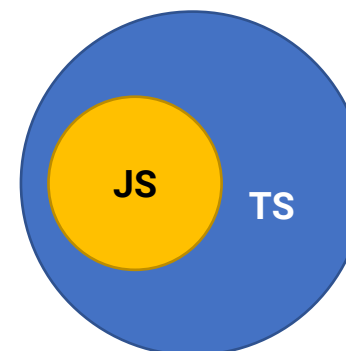


```
const user = {
  firstName: "Angela",
  lastName: "Davis",
  role: "Professor"
}

console.log(user.name)
```

Property 'name' does not exist on type '{ firstName: string; lastName: string; role: string; }'.

TypeScript validates your JavaScript ahead of time Next ▶



Angular

Es un Framework para crear aplicaciones de una sola página (SPA) de forma eficiente y sofisticada.

- ✓ Creado por Google.
- ✓ Desarrollado en TypeScript.
- ✓ Basado en componentes para crear aplicaciones web escalables.



Data Binding

Es la sincronización entre el modelo y la vista. Mantiene la página actualizada automáticamente según el estado de la aplicación.

✓ One Way Data Binding



✓ Two Way Data Binding



```
{{expression}}  
[target]="expression"
```

```
(target)="statement"
```

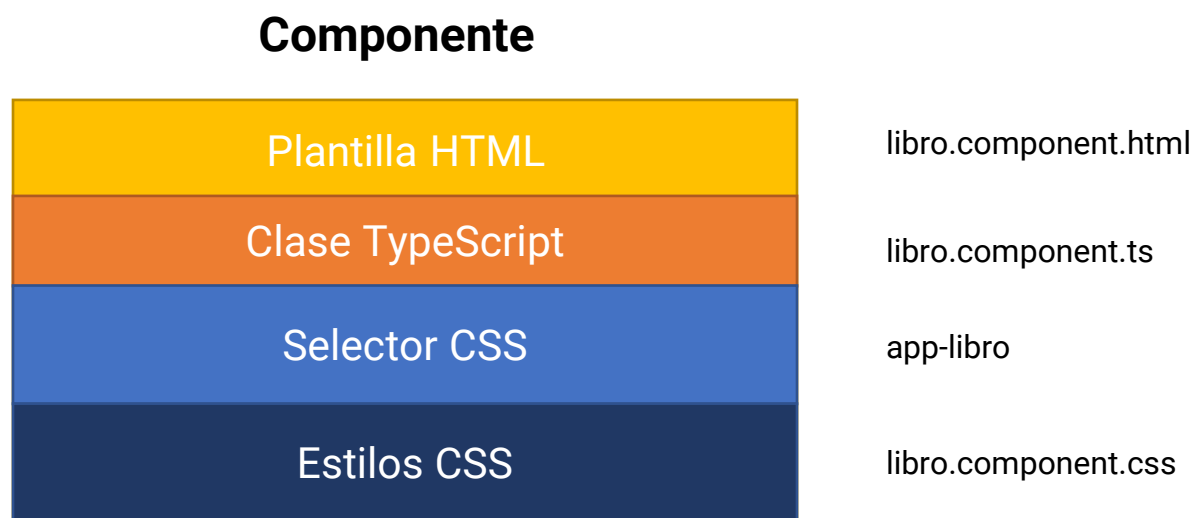
```
[(target)]="expression"
```

Data Binding

Type	Syntax	Category
Interpolation Property Attribute Class Style	<code>{{expression}}</code> <code>[target]="expression"</code> <code>bind-target="expression"</code>	One-way from data source to view target
Event	<code>(target)="statement"</code> <code>on-target="statement"</code>	One-way from view target to data source
Two-way	<code>[(target)]="expression"</code> <code>bindon-target="expression"</code>	Two-way

Componentes

Son el bloque de construcción principal de las aplicaciones Angular.



Plantilla

Es un fragmento de HTML que tiene la capacidad de aprovechar las características de Angular como:

- ✓ Interpolación
- ✓ Pipes
- ✓ Property binding
- ✓ Event binding
- ✓ Two-way binding

Directivas

Son clases que agregan comportamiento adicional a los elementos como administrar formularios, listas y estilos.

NgClass: agrega y elimina un conjunto de clases CSS.

NgStyle: agrega o elimina un conjunto de estilos HTML.

NgModel: agrega two-way data binding a un elemento de formulario HTML.

NgIf: crea o elimina condicionalmente subvistas de la plantilla.

NgFor: repite un nodo para cada elemento de una lista.

NgSwitch: un conjunto de directivas que cambian entre vistas alternativas.

Pipes

Son funciones útiles usadas en las expresiones de plantilla que recibe un valor y retorna una transformación de esta. Ej.: cadenas, montos monetarios, fechas, entre otros.

Los Pipes Angular más comunes son:

- **DatePipe**: Da formato a una fecha.
- **UpperCasePipe**: Transforma el texto a mayúsculas.
- **LowerCasePipe**: Transforma el texto a minúsculas.
- **CurrencyPipe**: Transforma un número a un valor monetario.
- **DecimalPipe**: Transforma un número usando un formato decimal.
- **PercentPipe**: Transforma un número usando un formato porcentual.

Formularios

Se usan para manejar las entradas del usuario. Por ejemplo, para iniciar sesión, actualizar un perfil, ingresar información personal, entre otras.

Angular proporciona dos formas de usar los formularios.

1. Basado en plantillas
2. Reactivo

Módulos

Son módulos de funciones usados para organizar el código. Por defecto, los módulos son cargados conjuntamente con la aplicación sean o no necesarios de forma inmediata. Sin embargo esto puede evitarse a través de Lazy Loading.

Entre los módulos Angular más utilizados tenemos:

- **BrowserModule**. Si la aplicación va a correr en un navegador.
- **CommonModule**. Si se va a utilizar directivas como `NgIf` y `NgFor`.
- **FormsModule**. Si se va a utilizar formularios basados en plantillas.
- **ReactiveFormsModule**. Para formularios reactivos.
- **RouterModule**. Si se va a utilizar `RouterLink`.
- **HttpClientModule**. Para la comunicación con un servidor.



**“Se amable, pues cada persona con
la que te cruzas está librando su
ardua batalla”**

Platón