

## Intro to Angular

### Events

```
<div>Users list is inside module</div>

<div *ngFor="let user of users">
  {{ user.name }} is {{ user.age }} years old
  <span (click)="remove">Remove</span>
</div>
```

#### Step 1 - make sure it works

```
removeUser(id: string): void {
  console.log('removeUser', id)
}
```

#### Step 2 - actually implement

```
removeUser(id: string): void {
  this.users = this.users.filter(user => user.id !== id)
  console.log('removeUser', id)
}
```

Add a user - html

```
<input type="text" (change)="setNewUserName($event.target.value)" />
<button (click)="addUser()">Add user</button>
```

Step 1 - make sure it works

Add the code for our two new events

```
setNewUserName(userName: string): void {
  | console.log('setNewUserName', userName)
}

addUser(): void {
  | console.log('addUser')
}
```

Step 2: - set a local variable to the new username

```
export class UsersListComponent {
  | newUserName: string = ''
}
```

Step 3 - now assign this variable when we leave (blue) the user name input

```
setNewUserName(userName: string): void {
  | this.newUserName = userName
  | console.log('setNewUserName', userName)
}
```

Step 4 - add the new user when the add button is pressed

```
addUser(): void {
  | const uniqueId = Math.random().toString(16)
  | const newUser = {
  |   | id: uniqueId,
  |   | name: this.newUserName,
  |   | age: 30
  | }
  | this.users.push(newUser)
}
```

We learnt how event binding works... now let take a look at property binding

1. It could be nice if the user name input was cleared after we click add...

You would think that we can just set up `newUserName` variable to "" like this

```
this.users.push(newUser)
this.newUserName = ''
```

This good and we need to do this BUT the input is NOT BOUND to anything

2. How to bind the user name input to a variable...

```
[value]="newUserName"
```

(the whole line)

```
<input type="text" (change)="setNewUserName($event.target.value)" [value]="newUserName" />
```

\*\*\* when an element is BOUND to a variable - whenever that variable changes - the element will change.... Because it is bound to it

## Component Styles

In our user list we would like to style each row

```
<div *ngFor="let user of users" class="user-container">
  <span> {{ user.name }} is {{ user.age }} years old </
span>
  <span (click)="removeUser(user.id)">Remove</span>
</div>
```

The style can be

```
.user-container {  
  font-size: 18px;  
  border-bottom: 1px solid grey;  
  padding: 10px;  
  display: flex;  
  justify-content: space-between;  
}
```

Style the component from WITHIN the component

```
:host {  
  border: 1px solid blue;  
  display: block;  
}
```