

Pro Angular

Chapter 12: Using Events and Forms

INSTRUCTOR: MD. HABIBUL HAQ
domain.habib@gmail.com

10 Nov 2022

Q1. Which one use to events binding?

- A. Round brackets
- B. Angle brackets
- C. Curly braces
- D. Banana in a box, [(event)]

Answer: A

Q2. Which character is used to define template reference variables?

- A. ?
- B. #
- C. \$
- D. &

Answer: B

Q3. Which one correctly uses template reference variable?

- A. `<input id="t" />{{t.value}}`
- B. `<input id="t" />{{#t.value}}`
- C. `<input ame="t" />{{t.value}}`
- D. `<input #t />{{t.value}}`

Answer: D

Q4. You want to pass a text input element's value to method when clicked on a button. Which code does this correctly?

- A. `<input type="text" id="t">`
`<button (click)="show(t)">Show</button>`
- B. `<input type="text" name="t">`
`<button (click)="show(t)">Show</button>`
- C. `<input type="text" [(ngModel)]= "t">`
`<button (click)="show(t)">Show</button>`
- D. `<input type="text" #t>`
`<button (click)="show(t)">Show</button>`

Answer: D

Q5. Which template properly sets two way binding to a model element?

- A. `<input [(input)]= "newProduct.name" [(value)]= "newProduct.name"/>`
- B. `<input [ngModel]= "newProduct.name" />`
- C. `<input (ngModel)= "newProduct.name" />`
- D. `<input [(ngModel)]= "newProduct.name" />`

Answer: D

Q6. Which one is not a built-in angular validation attribute?

- A. required
- B. minlength
- C. maxlength
- D. number

Answer: D

[required minlength maxlength pattern]

Q7. Which angular form validation class is assigned if element's contents have not been changed by the user?

- A. ng-untouched
- B. ng-touched
- C. ng-pristine
- D. ng-dirty

Answer: C

Q8. Which angular form validation class is assigned if element's contents have been changed by the user but the user then returns to the previous contents?

- A. ng-untouched
- B. ng-touched
- C. ng-pristine
- D. ng-dirty

Answer: D

Q9. Which template is appropriate to work field level validation?

- A. `<input name="name" [(ngModel)]="newProduct.name" required />`
`<div class="text-danger" *ngIf="name.dirty && name.invalid && name.errors.required">You must enter a product name </div>`
- B. `<input name="name" [(ngModel)]="newProduct.name" id="name" required />`
`<div class="text-danger" *ngIf="name.dirty && name.invalid && name.errors.required">You must enter a product name </div>`
- C. `<input name="name" [(ngModel)]="newProduct.name" #name="ngModel" required />`
`<div class="text-danger" *ngIf="name.dirty && name.invalid && name.errors.required">You must enter a product name </div>`
- D. `<input name="name" [(ngModel)]="newProduct.name" #name required />`
`<div class="text-danger" *ngIf="name.dirty && name.invalid && name.errors.required">You must enter a product name </div>`

Answer: C