

The following exercises are related to the use of Angular.

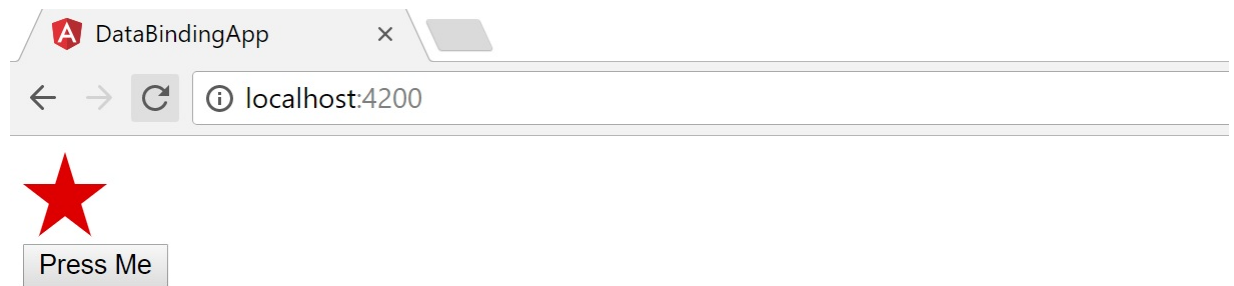
1. Create a git repository for your answers to this problem sheet. Push the repository to GitHub. Make a commit and push it to GitHub after each exercise.

Solution:

```
mkdir answers-TypeScript
-----
cd answers-TypeScript
-----
echo > README.md
-----
git init
Initialized empty Git repository in /Users/martin/answers-TypeScript/.git/
-----
git add .
-----
git commit -m "Empty first commit."
[master (root-commit) 6583cf6] Empty first commit.
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
-----

git remote add origin https://github.com/mkenirons/WHATEVER.git
You will need to add the path to your GitHub Repository here
-----
git push -u origin master
Add you file to GitHub (remote repository)
-----
```

2. Create a new Angular application entitled data-binding-app.
Use the following link: <https://cli.angular.io/>
(a) Update app.component.html so that it looks as follows:



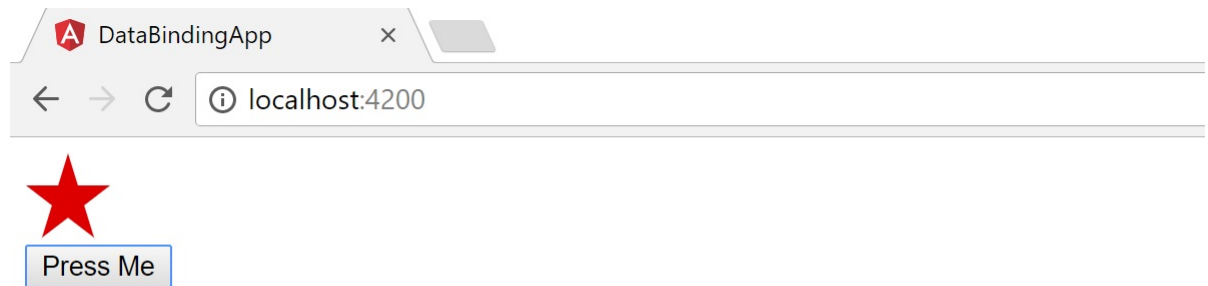
The Number of times the button has been pressed is: 0

NOTE: For the Star image use the following img tag:

```

```

- (b) Each time the user presses the button the “The Number of times the button has been pressed is: ” should be increased by 1. If the user pressed the button 3 times the view should look as follows:



The Number of times the button has been pressed is: 3

- (c) When the user double-clicks on the star, the view should show a h2 element (“Look at the star”) as follows:



Look at the star

Press Me

The Number of times the button has been pressed is: 0

- (d) If the user double-clicks on the star again, the h4 element (“Look at the star”) should not be shown.
- (e) Each time the user double-clicks on the star, if the h4 element (“Look at the star”) is shown it should be hidden, and if it is hidden it should be shown.

Solution:

```
npm install -g @angular/cli
ng new data-binding-app
cd data-binding-app
ng serve
```

```
//app.component.ts
```

```
export class AppComponent {
  title = 'app';
  numberOfTimes = 0;
  hidden = false;

  count(){
    this.numberOfTimes++;
  }

  show(){
    if(this.hidden == true)
```

```
{
  this.hidden = false;
}
else{
  this.hidden = true;
}
}
```

//app.component.html

```

<br>
<h2 [hidden]="hidden">Look at the star</h2>
<br>
<button (click)=count()>Press Me</button>

<br>
<h2>The Number of times the button has been pressed is: {{number}}</h2>
```

3. Add a new component to our application called listcomponent and get its template contents to display:

(a) Get our new component to display a list of the last four presidents of Ireland in an unordered list as follows:



- Higgins
- McAleese
- Robinson
- Hillery

i. The list can be stored in an array as follows:

```
presidents: string[] = ["Higgins", "McAleese", "Robinson", "Hillery"];
```

ii. If more presidents are added to the list, the view should not have to be updated.

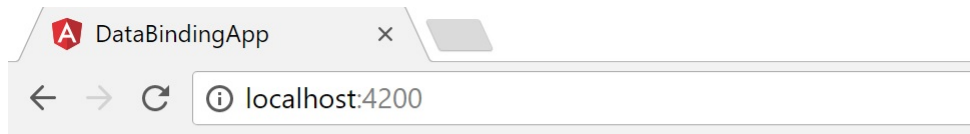
E.g.

```
presidents: string[] = ["Higgins", "McAleese", "Robinson", "Hillery", "O Dala"];
```

(b) Update the list of president to include a name and term attribute

```
presidents: any[] =  
  [{name: "Higgins", term: "2011-"},  
   {name: "McAleese", term: "1997-2011"},  
   {name: "Robinson", term: "1990-1997"},  
   {name: "Hillery", term: "1976-1990"},  
   {name: "O Dalaigh", term: "1974-1976"}];
```

(c) Update the view so that it now looks as follows:



- Higgins 2011-
- McAleese 1997-2011
- Robinson 1990-1997
- Hillery 1976-1990
- O Dalaigh 1974-1976

Solution:

```
<ul>  
  <li *ngFor= "let p of presidents">{{p.name}} {{p.term}}</li>  
</ul>
```