1. Why are most of the assignment operators restricted in template expression but allowed in template statement?

* Template expression we need to get value from component to view so we don’t want to create any change the value. But in templet statement we need to change the component so without using this we are unable to do this.

1. What is the use of safe navigation operator in a template expression? Can we live without it?

* It’s the way that we can navigate through property safely without null pointer and undefined values. **YES**, we can live without it by using nested loop if conditions.

1. What is the expression context for template expression? And for template statement?

* With the instance of the same component. But it can have templet reference variable.
* With the instance of the same templet. But it can have templet reference variable and $event.

1. Why is it important to make sure our template expressions execute quickly?

* Change detection that every time there is a change it needs to make Asynchronous call it needs to go and check it.

1. All the assignment operators are disabled in template expression. Does that guarantees template expression is free of side effects?

* No, we can make a method for getting but inside the method we can perform change to the value that is change the value before returning.

1. Angular always binds to DOM property not the HTML attribute. (YES/NO)

* No, sometimes when there is no property to bind angular uses the HTML attributes. Like for example colspan this doesn’t have corresponding property in DOM and another are like the ARIA (Accessible Rich Internet Applications), SVG (Scalable Vector Graphics).

1. What step should you take in order to make sure your angular application is free from script injection attack?

* Nothing. By itself the temp expression is safe.