** Event listeners **

Basic Syntax :-

With in the template access any event for a HTML item within parenthesis (.....).
↳ event

Eg:-

```
<input type="text" (keyup)="keyUpHandler()"/>
```

↳what to do
ts/js function

Now in ts file, within class we can make the handler function.

```
class GreeterComponent {
    keyUpHandler(){
        console.log(`key pressed`)
}
```

If an KeyboardEvent was passed:

i.e. `<input type="text" (keyup)="keyUpHandler($event)"/>`

then the function will look like:

→for ts

```
keyUpHandler(event: KeyboardEvent) {
    console.log(`key pressed: {event.key}`);
}
```

** Common Angular events **

- Mouse event:

(click) → on mouse click
(dblclick) → on mouse double click
(mouseenter) → when mouse enters an element
(mouseleave) → when mouse leaves an element
(mousemove) → when mouse moves in an element
(mousedown)/(mouseup) → mouse button pressed/released

- Keyboard Events:

(keydown) → key pressed down
(keyup) → key is released
(keypressed) → key is pressed

- Form & Input Events:

(input) → when value of <input>, <textarea> or <select> changes
(change) → when value of input/selection is commited
(focus)/(blur) → when an element gains or loses focus
(submit) → form submission.

- Touch Events (mobile):

(touchstart) → when the touch starts
(touchend) → when the touch ends
(touchmove) → when a finder moves on the screen.

- Window & Document Events:

for global events like resize or scroll, using @HostListener

eg:

```
@HostListner('window:scroll', ['$event'])

onScroll(event: Event){
    console.log('User scrolled!', event);
}
```