



#ANGULAR

# Angular Quick Tip: Binding Specific Keys to the Keyup and Keydown Events



When binding to either the `keyup` or `keydown` events in your Angular 2+ templates, you can specify key names. This makes it very easy to trigger the event only when specific keys are pressed.

First, let's see an example without using a key name. Let's say we want the event to be triggered only when the user presses the `enter` key:

```
<input (keydown)="onKeydown($event)">
```

```
onKeydown(event) {  
  if (event.key === "Enter") {  
    console.log(event);  
  }  
}
```

**Buildkite**

**Try Buildkite!** Fast, secure, and scalable CI/CD for all your software projects.

ETHICAL AD BY CODEFUND



## more angular

Building Maps in Angular using Leaflet, Part 4: The Shape Service

Building Maps in Angular using Leaflet, Part 3: The Popup Service

Building Maps in Angular using Leaflet, Part 2: The Marker Service



Fast-track your **Angular** skills to build any project or get hired at a top company.

Show me how!

[This affiliate banner](#) helps support the site 🙌

Building Maps in Angular  
Using Leaflet, Part 1:  
Generating Maps

Using Angular CLI  
Schematics

[all angular posts](#)

Now the same example, but made simpler with the addition of the `enter` key name to the event:

```
<input (keydown.enter)="onKeydown($event)">
```

```
onKeydown(event) {  
  console.log(event);  
}
```

## Key Combinations

You can also combine keys together to trigger the event only when the key combination is triggered. In the following example, the event will trigger only if the `control` and `1` keys are pressed at the same time:

```
<input (keyup.control.1)="onKeydown($event)">
```

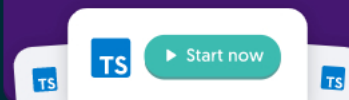
## Some Examples



follow us  
@alligatorio

ultimatecourses.com

The best courses to  
learn TypeScript



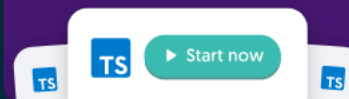
🕒 About this affiliate banner



stay in the loop  
with our web dev  
newsletter

ultimatecourses.com

The best courses to  
learn TypeScript



🕒 About this affiliate banner

This feature works for special and modifier keys like `enter`, `escape` (`esc`), `shift`, `alt`, `tab`, `backspace` and `command` (`meta`), but it also works for `letters`, `numbers`, `arrows` and `f` keys (`f1` though `f12`).

Here are quite a few more examples, to give you an idea of what's possible:

```
<input (keydown.enter)="...">
<input (keydown.a)="...">
<input (keydown.esc)="...">
<input (keydown.shift.esc)="...">
<input (keydown.control)="...">
<input (keydown.alt)="...">
<input (keydown.meta)="...">
<input (keydown.9)="...">
<input (keydown.tab)="...">
<input (keydown.backspace)="...">
<input (keydown.arrowup)="...">
<input (keydown.shift.arrowdown)="...">
<input (keydown.shift.control.z)="...">
<input (keydown.f4)="...">
```

Published: May 1, 2017




## Ultimate Angular

Level up your Angular game  
with courses from Todd  
Motto

[About this affiliate link](#)



stay in the loop  
with our web dev  
newsletter:


 [me@croco.io](mailto:me@croco.io)

Subscribe!

[About this affiliate banner](#)




stay in the loop  
with our web dev  
newsletter:

 [me@croco.io](mailto:me@croco.io)

Subscribe!

Catch up with the latest in front-end  
web dev with our monthly  
newsletter:

 me@croco.io

Subscribe

Write for Us

About

Contact

Credits

Privacy

[Support us on Patreon](#) • [Advertise](#)

Note that we link to certain 3rd party products via affiliate or sponsored links. You  
can [learn more about it here](#).

Code snippets licensed under [MIT](#), unless otherwise noted.

Content & Graphics © 2019 Alligator.io LLC



