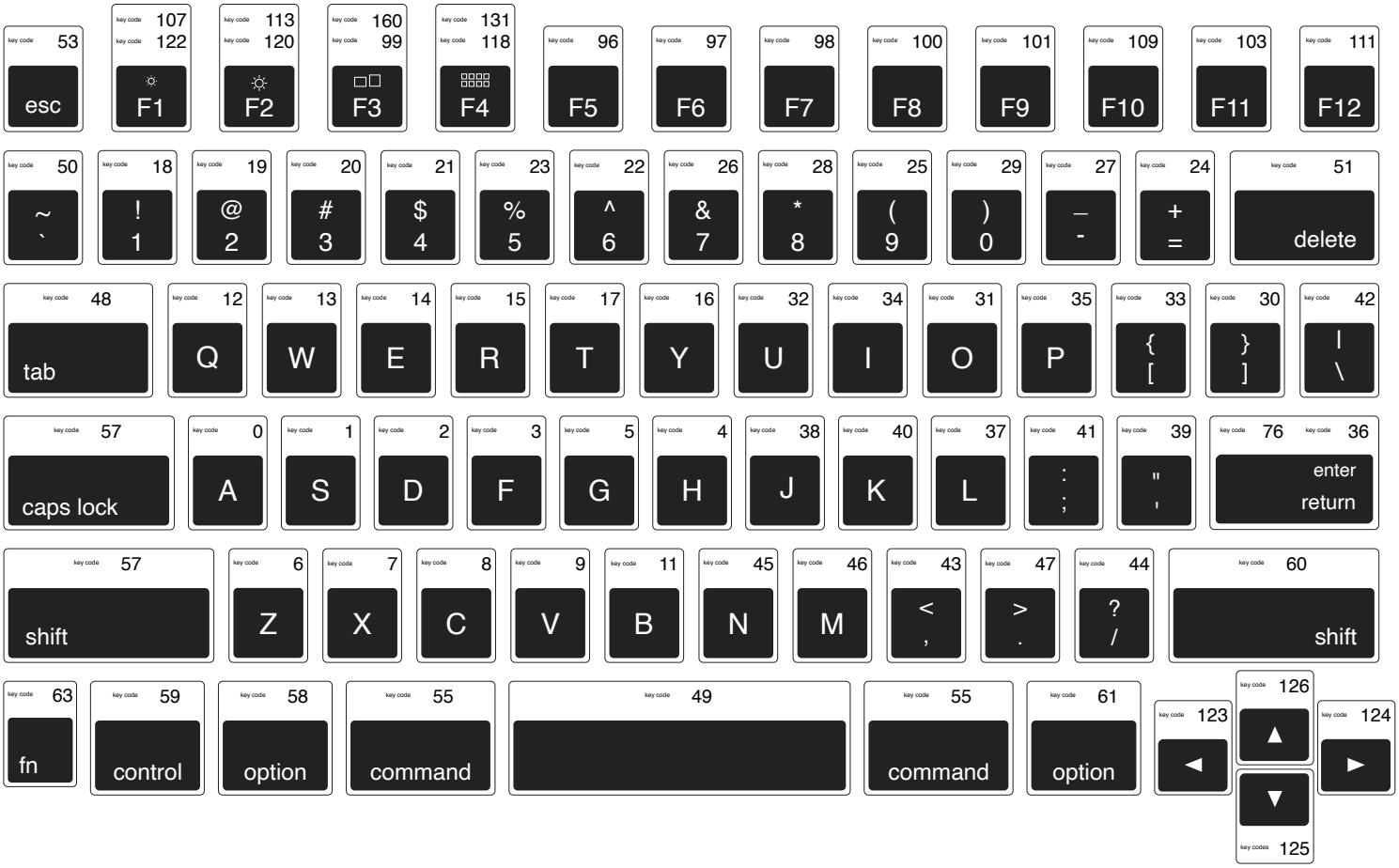


Complete list of AppleScript key codes

By Christopher Kielty, last updated May 10, 2018

Tweet Share Email RSS

All of the key codes. All of them. Ever. Maybe. I tested this out on a MacBook Air and also a MacBook Pro. If I missed something, please ✉ let me know.

Key code 49 in this example triggers the space bar.

```
tell application "System Events"
  key code 49
end tell
```

You may also want to check out this short introduction to ⌨ keyboard automation in macOS (with AppleScript).

The *play/pause function keys* don't trigger correctly using key codes, but there's a workaround for this, which I'll go over.

Keyboard brightness keys **don't work with key codes**. I don't know how to make them work. Yet... Yet.

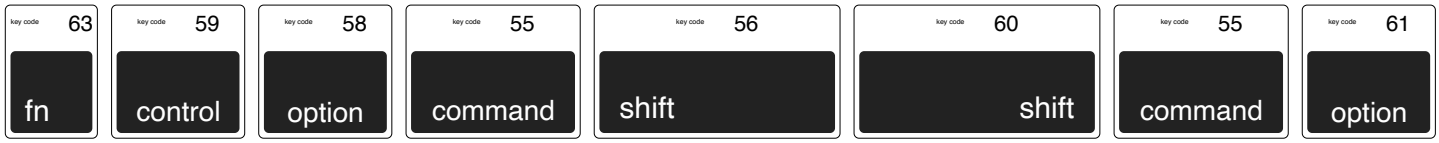
Oh, and also caps lock. It doesn't look like key code 57 enables caps lock. I am looking into this.

Return and enter

The enter key on most Macs is actually the return key (key code 36). The key code for enter is 76. The enter key is what you might see on a full size keyboard on the numpad side. On the more common, non-full-size Mac keyboards, enter can still be accomplished by hitting fn + enter. This is why the enter key says return *and* enter on it.

Most applications don't really care about the difference between return and enter. In pretty much every text editor both enter and return will result in the same new line. One example of where these two keys do different things is with iTunes. Interestingly, in iTunes, enter (76) renames the currently selected song while return (36) plays the song from the beginning. This is still true as of iTunes 12 running on OS X 10.10 Yosemite.

Modifier keys



Most of the modifiers have two different key codes. One for the left and one for the right. So instead of just triggering, say, option, you can trigger (right) option specifically. This applies to option, shift, and control. It appears that command has only one key code. I think it is worth noting that the keyboard on the MacBook Air I'm using right now has only one control key on the physical keyboard.

(Right) control : `key code 62`

(Right) option : `key code 61`

(Right) shift : `key code 60`

While this is all fine and dandy, I'm guessing you probably want to use these keys to modify other keys. They *are* modifier keys, right? Actually, key codes aren't even needed to accomplish this. For example, you could do command + A like this:

```
tell application "System Events" to keystroke a using command down
```

Use multiple modifier keys with `{..., ...}`. In this example, let's do a "Paste and Match Style" with option + shift + command + V like this:

```
tell application "System Events" to key code 9 using {option down,
shift down, command down}
```

Key code 9, in the example above, is the V key. But it would be just as easy to use keystroke, like this:

```
tell application "System Events" to keystroke "V" using {option
down, shift down, command down}
```

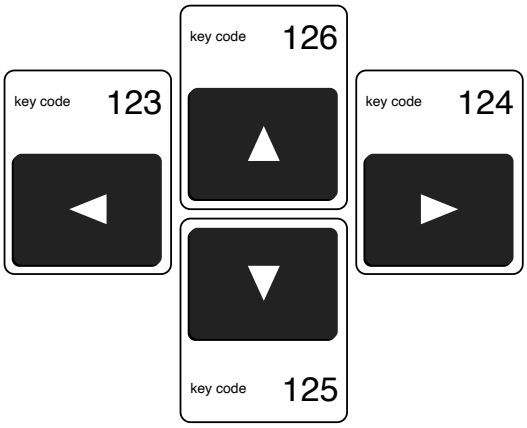
For numbers, letters, and symbols, using keystroke is probably better. However, there's still key codes if you want 'em.

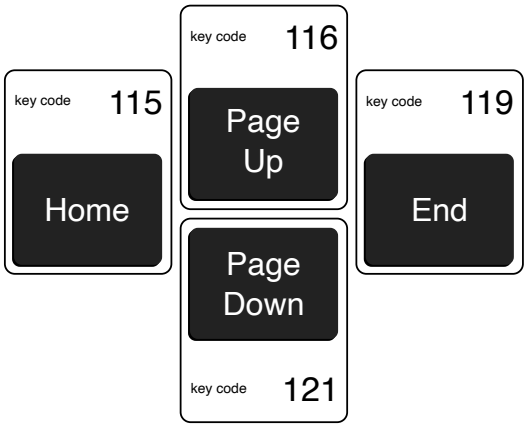
Remember, when using keystroke, place the characters in quotes. Also, don't forget, keystroke (*one word*), key code (*two words*). I accidentally type keycode sometimes.

```
keystroke "Hello world"
```

Like that

Arrow keys, page up, page down, home and end





Arrow keys are pretty great. And also the page up, page down, home, and end keys. Those are pretty great also. Scripting them is awesome. And this is where key codes are really necessary. You can't write *keystroke up arrow*. That won't work. You don't need to try that. I've saved you the trouble.

Some nifty arrow key examples. Try these in a text editor to move the cursor around:

Skip ahead one word: `tell application "System Events" to key code 124 using option down`

Go back one word: `tell application "System Events" to key code 123 using option down`

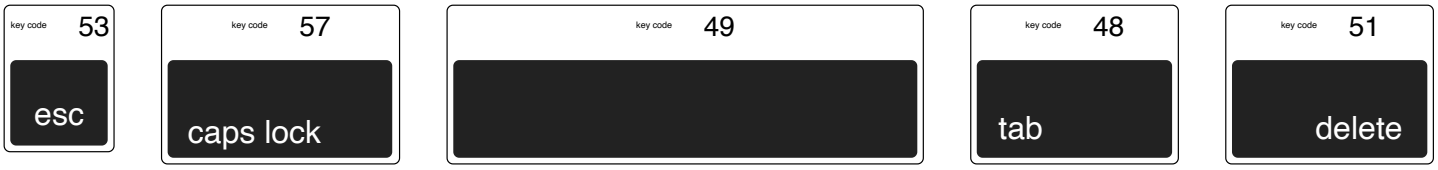
Skip to the end of the paragraph: `tell application "System Events" to key code 125 using option down`

Go back to the beginning of the paragraph: `tell application "System Events" to key code 126 using option down`

Skip to the end of the line: `tell application "System Events" to key code 124 using command down`

Go back to the beginning of the line: `tell application "System Events" to key code 123 using command down`

Esc, space bar, tab, delete, caps lock



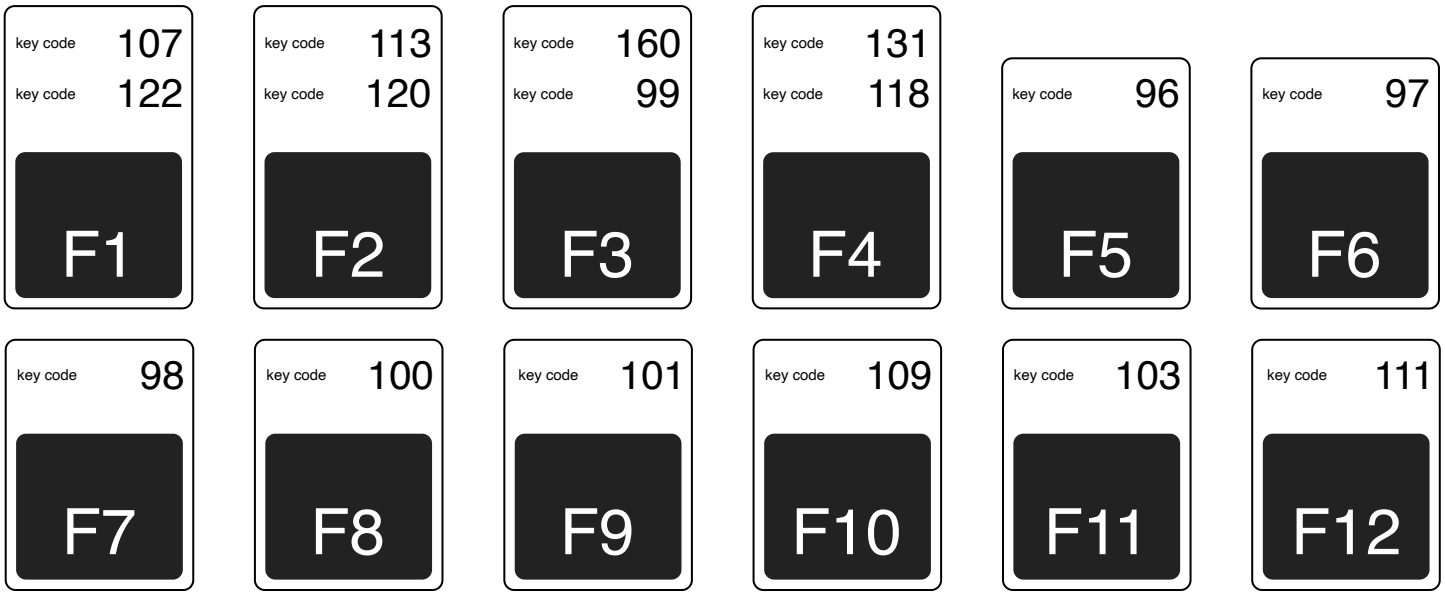
All of these work as expected, except for caps lock. It doesn't look like key code 57 does anything. Too bad. That'd be neat.

Keystroke works just fine for triggering the return, space, and tab keys.

```
keystroke return
keystroke space
keystroke tab
```

This is the exception to the rule about keystroke and quotes. Putting *"return"* in quotation marks would write out *return*. No quotes and the return key is triggered.

F keys and some of the function keys



Not all of the function keys can be reliably scripted using key codes. See further down for work arounds.

In the diagram above, some of the keys list two key codes. The top number corresponds to the function and the bottom number corresponds to the F key. For example, this would increase screen brightness by one increment, the same as pressing the screen brightness key.

```
tell application "System Events" to key code 113
```

What about F13-F20? Those are scriptable too!

```
F13 : key code 105      F17 : key code 64
F14 : key code 107      F18 : key code 79
F15 : key code 113      F19 : key code 80
F16 : key code 106      F20 : key code 90
```

Play, pause, fast forward, rewind and volume function keys

I dunno if there's a good way to do this using key codes. You can leapfrog this hurdle by just not using key codes.

Play: `tell application "iTunes" to play`

Pause: `tell application "iTunes" to pause`

Rewind (previous track): `tell application "iTunes" to previous track`

Fast forward (next track): `tell application "iTunes" to next track`

If you're using something other than iTunes, you can still try the above. Just substitute "iTunes" with the name of the app. Depending on how scriptable the app is, this may or may not work.

Mute, unmute, set and increment system volume like so:

Mute: `set volume with output muted`

Unmute: `set volume without output muted`

Set volume to 100%: `set volume output volume 100`

Set volume to 50%: `set volume output volume 50`

Set volume to 1%: `set volume 1`

Make your system volume slowly fade out with this nifty little script. Adjust the "delay 0.2" bit in the middle of the loop to speed up or slow down the fade.

```
set a to output volume of (get volume settings)

repeat while a is not 0
  set a to (a - 1)
  delay 0.2
  set volume output volume a
end repeat
```

Basically...

Set a to current volume

Repeat until volume is zero

Set volume decrement to current volume -1%

Delay 0.2 seconds between each decrement

Decrement volume by set amount.

Controlling keyboard brightness

I don't know how to control keyboard brightness with Applescript. Yet.

Letters, numbers and symbols

Instead of using *key codes*, why not just use *keystrokes*? This works great with all of the letter keys (upper and lower case). Also, the number keys (above the letter keys, not numpad) and all of the symbols you can make with them using `shift`. Also these keys: `~\`` `{[` `}]` `|\` `:;` `"'` `_-` `+=` `<,` `>.` and `?/`. And also other keys.

Keystroke works like this:

```
tell application "System Events" to keystroke "Abcde"
```

Keystrokes might be super cool, and awesome, and generally pretty great and what have you. But guess what! There's also key codes for these!...:

!1 : key code 18

@2 : key code 19

#3 : key code 20

$4 : key code 21

%5 : key code 23

^6 : key code 22

&7 : key code 26

*8 : key code 28

(9 : key code 25

)0 : key code 29

~` : key code 50

{[: key code 33

}] : key code 30

|\ : key code 42

:; : key code 41

"' : key code 39

_- : key code 27

+= : key code 24

<, : key code 43

>. : key code 47

?/ : key code 44

N : key code 45

M : key code 46

Q : key code 12

W : key code 13

E : key code 14

R : key code 15

T : key code 17

Y : key code 16

U : key code 32

I : key code 34

O : key code 31

P : key code 35

A : key code 0

S : key code 1

D : key code 2

F : key code 3

G : key code 5

H : key code 4

J : key code 38

K : key code 40

L : key code 37

Z : key code 6

X : key code 7

C : key code 8

V : key code 9

B : key code 11

Numpad key codes

And the numpad. There's even key codes for the numpad. Key codes for everyone!

Numpad 1 : key code 83	*Numpad* 0 : key code 82
Numpad 2 : key code 84	*Numpad* * : key code 67
Numpad 3 : key code 85	*Numpad* / : key code 75
Numpad 4 : key code 86	*Numpad* + : key code 69
Numpad 5 : key code 87	*Numpad* - : key code 78
Numpad 6 : key code 88	*Numpad* = : key code 81
Numpad 7 : key code 89	*Numpad* . : key code 65
Numpad 8 : key code 91	*Numpad* clear : key code 71
Numpad 9 : key code 92	

<-- Back to Front Page