

Jessica Yung
<https://www.jessicayung.com/>

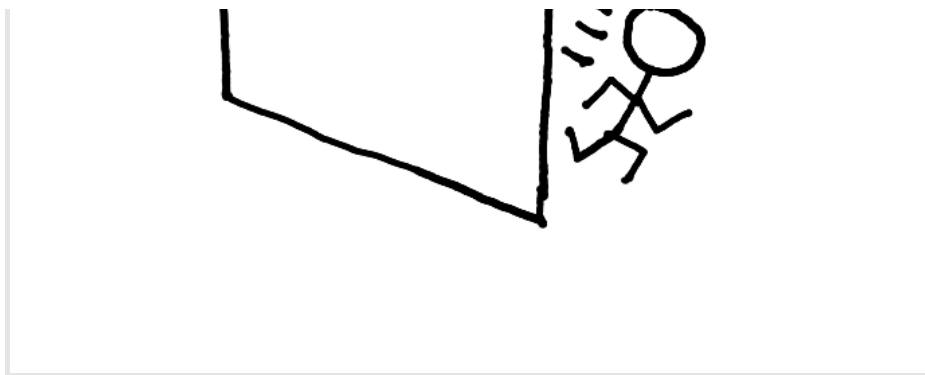
HOME ([HTTPS://WWW.JESSICAYUNG.COM/](https://www.jessicayung.com/))

ABOUT ✕ ([HTTP://HK.LINKEDIN.COM/IN/JESSICAYUNGC](http://hk.linkedin.com/in/jessicayungyc))

PROJECTS ✕ ([HTTPS://WWW.JESSICAYUNG.COM/PROJECTS/](https://www.jessicayung.com/projects/))

BLOG ([HTTPS://WWW.JESSICAYUNG.COM/BLOG/](https://www.jessicayung.com/blog/))

CONTACT ([HTTP://JESSICAYUNG.COM#CONTACT](http://jessicayung.com/#contact))



When running machine learning experiments, you might want to run multiple scripts simultaneously, hide printouts for a script or just do things in a terminal

[LSTMs for Time Series in PyTorch](https://www.jessicayung.com/lstm-for-time-series-in-pytorch/)
(<https://www.jessicayung.com/lstm-for-time-series-in-pytorch/>)

[How to run scripts in the background](https://www.jessicayung.com/how-to-run-scripts-in-the-background/)
(<https://www.jessicayung.com/how-to-run-scripts-in-the-background/>)

[Python Lists vs Dictionaries: The space-time tradeoff](#)

window while running a Jupyter Notebook in the background.

In this post, we will go through how to run scripts in the background, bring them back to the foreground, and check if the scripts are still running.

Running scripts in the background

Suppose you've already started running your script, `python script.py`. Then:

1. Press `Ctrl+Z` to pause the script.

1. You might see

```
1 ^Z
2 [1]+ Stopped python script.py
```

2. Type `bg` to run the script in the background. You should see

```
1 [1]+ python script.py &
```

1. OR type `fg` to run the script in the foreground. You should see

```
1 [1]+ python script.py &
```

and the script continuing to run.

You can also run the script in the background directly by typing

([https://www.jessicayung.com/pyth-
lists-vs-dictionaries-the-
space-time-tradeoff/](https://www.jessicayung.com/pyth-
lists-vs-dictionaries-the-
space-time-tradeoff/))

[MSE as Maximum](#)

[Likelihood](#)

([https://www.jessicayung.com/mse-
as-maximum-likelihood/](https://www.jessicayung.com/mse-
as-maximum-likelihood/))

[Numpy Views vs Copies:](#)

[Avoiding Costly Mistakes](#)

([https://www.jessicayung.com/nun-
views-vs-copies-avoiding-
costly-mistakes/](https://www.jessicayung.com/nun-
views-vs-copies-avoiding-
costly-mistakes/))

[What makes Numpy Arrays](#)

[Fast: Memory and Strides](#)

([https://www.jessicayung.com/nun-
arrays-memory-and-
strides/](https://www.jessicayung.com/nun-
arrays-memory-and-
strides/))

[Using generators in Python](#)

[to train machine learning](#)

[models](#)

([https://www.jessicayung.com/usin-
generators-in-python-to-
train-machine-learning-
models/](https://www.jessicayung.com/usin-
generators-in-python-to-
train-machine-learning-
models/))

Recent Comments

[Jessica Yung](#)

(<http://jessicayung.com>) on [Explaining Tensorflow Code for a Convolutional Neural Network](#)

([https://www.jessicayung.com/expl-
tensorflow-code-for-a-
convolutional-neural-
network/#comment-134](https://www.jessicayung.com/expl-
tensorflow-code-for-a-
convolutional-neural-
network/#comment-134))

[Jessica Yung](#)

(<http://jessicayung.com>) on [Self-Driving Car Engineer](#)

```
1 python script.py &
```

in the console. The `&` symbol instructs the process to run in the background. E.g. I often run `jupyter notebook &`.

Inspecting processes

Sometimes you may want to check if a process is still running, how long a process has been running or whether it is hanging. (Hanging here means the program is stuck or is not responding to inputs.)

1. Type `ps -x` to list all processes (that are executables).
 1. If you're on your home computer as opposed to a remote server, there may be many processes running, and you may have to run `ps -x | grep python` or `ps -x | grep script.py` instead to find your script.
 1. This finds all processes with the word `python` in them.
 1. `|` pipes the output of the first command (`ps -x`) to the input of the second command (`grep [word to search] [files to search]`).
 2. `grep python files_to_search` finds instances of the string `python` in `files_to_search`.
 2. Find the id of your process.

```
1 [pid] [tty] [time script's been running for] [scr
```

Nanodegree
(<https://www.jessicayung.com/portfolio/self-driving-car-engineer-nanodegree/#comment-133>)

KL Tah
(<https://plus.google.com/113572011800000000000/post/self-driving-car-engineer-nanodegree-term-1-review/#comment-132>)

Jessica Yung
(<http://jessicayung.com>) on Self-Driving Car Engineer Nanodegree Term 1 Review
(<https://www.jessicayung.com/self-driving-car-engineer-nanodegree-term-1-review/#comment-131>)

Jessica Yung
(<http://jessicayung.com>) on Self-Driving Car Engineer Nanodegree Term 1 Review
(<https://www.jessicayung.com/self-driving-car-engineer-nanodegree-term-1-review/#comment-130>)

Archives

October 2018
(<https://www.jessicayung.com/2018/10/>)

September 2018
(<https://www.jessicayung.com/2018/09/>)

August 2018
(<https://www.jessicayung.com/2018/08/>)

```
2 ipt name]
3 2939 ttys003 0:01.60 python script.py
  2949 ttys003 0:00.00 grep python
```

1. `pid` stands for process ID.
2. `tty` stands for teletype terminals, which were the terminals people used when people first started to use computers (<https://unix.stackexchange.com/questions/5421/what-is-the-concept-behind-tty-in-linux>).

1. If you are on Linux, you can run `pstack $ID`, which should print out the ongoing output of your process. If the process is not hanging, you should see a lot of continuous output which will suggest which part of the program is running. If it is hanging, there likely won't be many (if any) continuing printouts.
2. If you want to stop the process, you can type `kill $ID`.
 1. If you want to stop the process while it is in the foreground, type `Ctrl+C`.
 2. You can use this to e.g. get rid of an experiment that isn't responding.

I hope this has been helpful! You can try running a few Python scripts using the same terminal or debugging your Python scripts using this method.

References:

- `tty`: Teletype terminals (<https://unix.stackexchange.com/questions/5421/what-is-the-concept-behind-tty-in-linux>)

[June 2018](https://www.jessicayung.com/2018/06/)
(<https://www.jessicayung.com/2018/06/>)

[March 2018](https://www.jessicayung.com/2018/03/)
(<https://www.jessicayung.com/2018/03/>)

[January 2018](https://www.jessicayung.com/2018/01/)
(<https://www.jessicayung.com/2018/01/>)

[December 2017](https://www.jessicayung.com/2017/12/)
(<https://www.jessicayung.com/2017/12/>)

[October 2017](https://www.jessicayung.com/2017/10/)
(<https://www.jessicayung.com/2017/10/>)

[September 2017](https://www.jessicayung.com/2017/09/)
(<https://www.jessicayung.com/2017/09/>)

[June 2017](https://www.jessicayung.com/2017/06/)
(<https://www.jessicayung.com/2017/06/>)

[May 2017](https://www.jessicayung.com/2017/05/)
(<https://www.jessicayung.com/2017/05/>)

[April 2017](https://www.jessicayung.com/2017/04/)
(<https://www.jessicayung.com/2017/04/>)

[March 2017](https://www.jessicayung.com/2017/03/)
(<https://www.jessicayung.com/2017/03/>)

[February 2017](https://www.jessicayung.com/2017/02/)
(<https://www.jessicayung.com/2017/02/>)

[January 2017](https://www.jessicayung.com/2017/01/)
(<https://www.jessicayung.com/2017/01/>)

[December 2016](https://www.jessicayung.com/2016/12/)
(<https://www.jessicayung.com/2016/12/>)

[November 2016](https://www.jessicayung.com/2016/11/)
(<https://www.jessicayung.com/2016/11/>)

[October 2016](https://www.jessicayung.com/2016/10/)
(<https://www.jessicayung.com/2016/10/>)

[September 2016](https://www.jessicayung.com/2016/09/)
(<https://www.jessicayung.com/2016/09/>)