

Learn to code — free 3,000-hour curriculum

FEBRUARY 3, 2023 / #LINUX

How to Manage Linux Processes



Arunachalam B

We all follow certain processes to achieve our goals. Similarly, every system has its own processes to accomplish tasks.

Every program or command that executes in a Linux system is called a process.

In this tutorial, let's explore processes and how we can manage them in Linux.

What is a Linux Process?

A process is theoretically called a program in execution. It's basically a task that a system is currently working on.

Every action you take on the system will result in a new process. For example, opening a browser initiates a process.

In simple words, a process is an instance of a program. The user action is transformed into a command and a new process will be

Learn to code — free 3,000-hour curriculum

of the hierarchy implies, a process initiated from a command/program is called the parent process and the produced process of a parent process is called the child process.

Types of Linux Processes

Processes are classified into 2 types in Linux Distributions:

1. Foreground Processes
2. Background Processes

Foreground processes

A process that requires the user to start it using a Terminal command or Program is called a foreground process. This means that foreground processes require an input trigger from a user. So every foreground process is manually triggered.

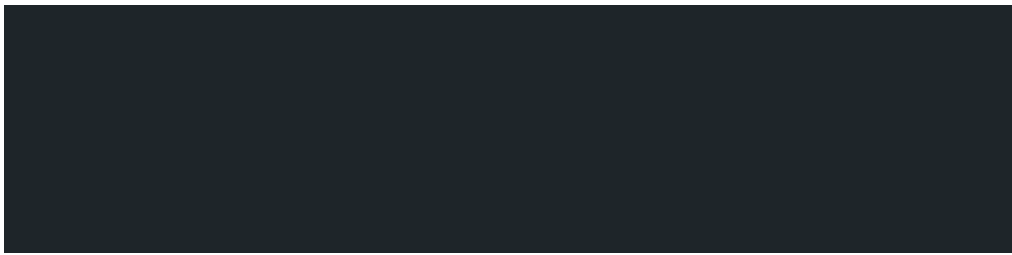
Whenever a process is running in the foreground, the other processes should wait until the current process completes.

The best example to demonstrate this is via the `sleep` command. The `sleep` command does not allow the user to interact with the terminal until a given number of seconds has passed.

```
sleep 10
```

Terminal command to sleep for 10 sec in foreground

Learn to code — free 3,000-hour curriculum



`sleep` terminal command running on foreground and blocks user input

We should wait for 10 seconds to access the terminal to run another command.

Background Processes

A process that runs independently on user input is called a background process. Unlike the foreground processes, we can run multiple processes at the same time in a background process.

To run a process in the background, place an ampersand (&) at the end of the command that you use to start the process.

Here's a quick example to demonstrate that:

Let's execute the `sleep` command in a background process. It'll run in the background and gives the terminal back to us to run other commands.

Learn to code — free 3,000-hour curriculum

```
/home/gogosoon
gogosoon@gogosoon-Inspiron-5515:~$
[1]+  Done                  sleep 10
gogosoon@gogosoon-Inspiron-5515:~$
```

Sample Terminal command for a background process

```
sleep 10 &
```

Terminal command to sleep for 10 sec in background.

Now we can see that the above command runs in the background. It created a process with the PID (19003). So we can run another command simultaneously (`pwd` command).

How to Change a Foreground Process to a Background Process

If we start a process in the foreground and would like to place it in the background, we can do it using the `bg` command. Let's see how to change the foreground process to the background.

If a process is running, press the key `CTRL+Z` . This command will suspend the current process.

Learn to code — free 3,000-hour curriculum

```
The following signatures were invalid: EXPKEYSIG 23E7166788B63E1E Yarn Packaging <yarn@dan.cx>
Hit:6 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:7 https://repo.pritunl.com/stable/apt focal InRelease
Hit:8 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:9 http://ppa.launchpad.net/zorinos/apps/ubuntu focal InRelease
Hit:10 http://security.ubuntu.com/ubuntu focal-security InRelease
8% [Waiting for headers] [Connected to packagecloud.io (54.183.32.181)] [Connected to packages.zorinos.com (138.68.36.163)]^Z
[1]+  Stopped                  sudo apt update
gogosoan@gogosoan-Inspiron-5515:~$
```

Foreground process output

Then run the `bg` command. It takes a process id as an argument and places the process into the background. If the argument is empty it will place the currently suspended process in the background.

```
bg <process_id>
```

Command to move the process to background

```
bg
```

Command to move the last process to background

```
gogosoan@gogosoan-Inspiron-5515:~$ bg
[1]+ sudo apt update &
Hit:11 http://ppa.launchpad.net/zorinos/drivers/ubuntu focal InRelease
Hit:12 http://ppa.launchpad.net/zorinos/patches/ubuntu focal InRelease
Hit:14 https://packages.zorinos.com/stable focal InRelease
Hit:15 https://packages.zorinos.com/patches focal InRelease
Hit:16 https://packages.zorinos.com/apps focal InRelease
Hit:17 https://packages.zorinos.com/drivers focal InRelease
Hit:18 http://ppa.launchpad.net/zorinos/stable/ubuntu focal InRelease
Hit:13 https://packagecloud.io/slacktechnologies/slack/debian jessie InRelease
Fetched 17.1 kB in 1min 41s (169 B/s)
Reading package lists... 6%
```

Foreground process to background process output

Learn to code — free 3,000-hour curriculum

How to List Linux Processes

Before we go over how to do this, you should know why you might need to know a list of processes. Here are a few reasons:

1. To know which process consumes more time.
2. To know which process takes more memory and CPU usage.
3. To know the triggered command for a running process.

To see the processes that are running currently, we can use `ps` (Process Status) command:

`ps`

Terminal command to list the running processes

```
gogosoon@gogosoon-Inspiron-5515:~$ ps
  PID TTY          TIME CMD
 17654 pts/4        00:00:00 bash
 18194 pts/4        00:00:00 ps
gogosoon@gogosoon-Inspiron-5515:~$
```

`ps` command showing the list of running processes

Learn to code — free 3,000-hour curriculum

UP NEXT: [How to Manage Linux Processes](#)

W

Terminal command to list the processes of all logged in users

```
gogosoon@gogosoon-Inspiron-9515:~$ w
20:25:35 up 10:04, 1 user, load average: 2.98, 1.78, 1.12
USER      TTY      FROM          LOGIN@   IDLE   XCPU  PCPU  WHAT
gogosoon  :0                10:21    ?xdm?  42:11  0.01s /usr/libexec/gdm-x-session --run-script env GNOME_SHELL_SESSION_MODE=zorin /usr/bin/gnome-session --session=zorin
gogosoon@gogosoon-Inspiron-9515:~$
```

w command displaying the list of processes of all users

How to List the Processes in Tree View

When a program/command runs, it initiates a main process called the parent process. The parent process may depend on some other command/program which will create a child process.

Here's an example screenshot.

Learn to code — free 3,000-hour curriculum

```
2*[Isolated Web Co—25*[{Isolated Web Co}]]
15*[Isolated Web Co—24*[{Isolated Web Co}]
Isolated Web Co—24*[{Isolated Web Co}]
Privileged Cont—24*[{Privileged Cont}]
RDD Process—2*[{RDD Process}]
Socket Process—4*[{Socket Process}]
Utility Process—4*[{Utility Process}]
3*[Web Content—17*[{Web Content}]]
WebExtensions—28*[{WebExtensions}]
205*[{firefox}]
```

Child processes of the parent (firefox) process

In the above screenshot, Firefox is the parent process and the other processes are its child processes.

Let's explore how to list the process in a tree-like structure.

`ps tree` is a Linux command to list the currently running process of all users in a tree-like structure. It is used as a more visual alternative to the `ps` command.

`ps tree`

Terminal command to list the processes in a tree like structure

Learn to code — free 3,000-hour curriculum



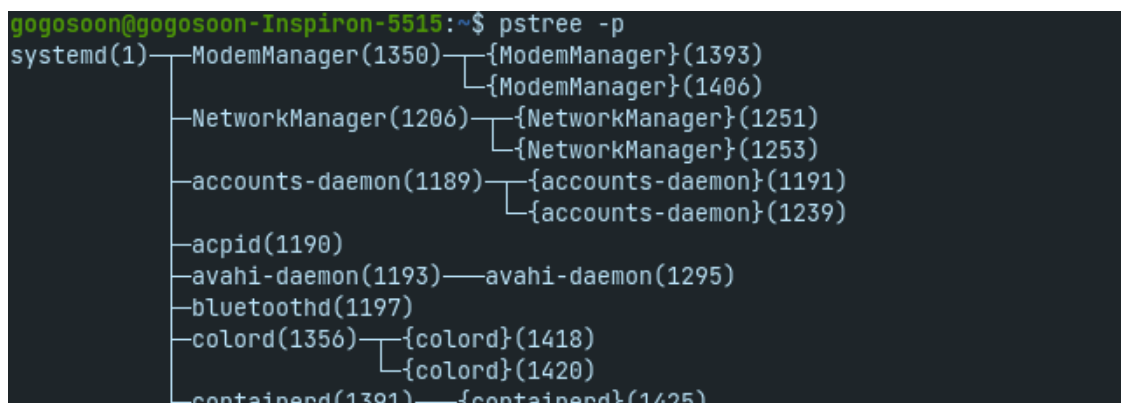
pstree command listing processes in tree view

As we can see, the running processes are in tree form. This can be useful to visualize the processes.

Adding the `-p` flag with the command will display each branch with its process id.

```
pstree -p
```

Terminal command to display the child processes of a parent process in tree like structure

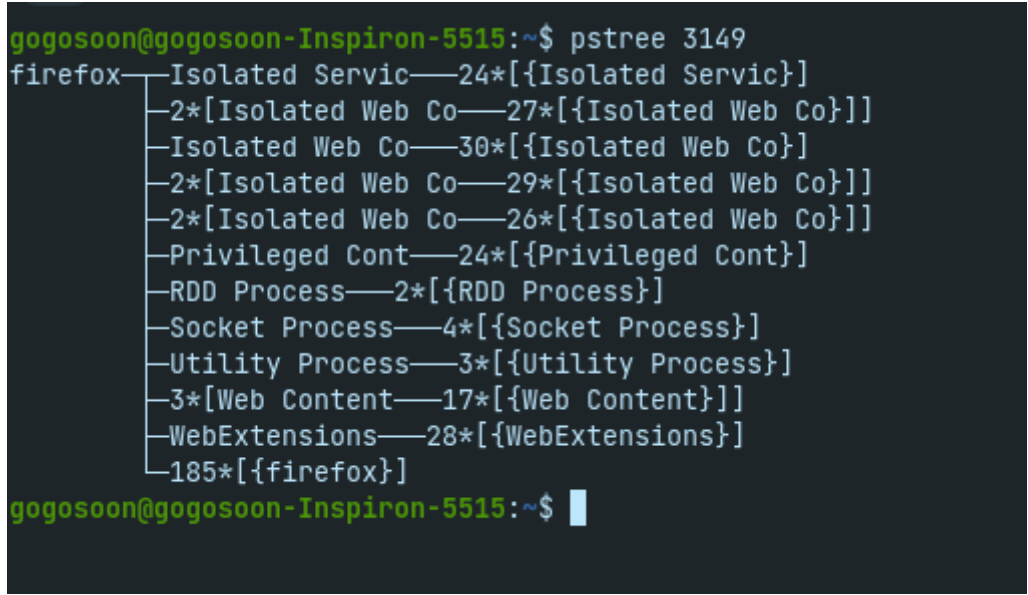


Terminal command displaying the list of processes in tree view with PID

Learn to code — free 3,000-hour curriculum

```
pstree 3149
```

Terminal command to list the processes of 3149 process id



```
gogosoon@gogosoon-Inspiron-5515:~$ pstree 3149
firefox—Isolated Servic—24*[{Isolated Servic}]
      |—2*[Isolated Web Co—27*[{Isolated Web Co}]]
      |—Isolated Web Co—30*[{Isolated Web Co}]
      |—2*[Isolated Web Co—29*[{Isolated Web Co}]]
      |—2*[Isolated Web Co—26*[{Isolated Web Co}]]
      |—Privileged Cont—24*[{Privileged Cont}]
      |—RDD Process—2*[{RDD Process}]
      |—Socket Process—4*[{Socket Process}]
      |—Utility Process—3*[{Utility Process}]
      |—3*[Web Content—17*[{Web Content}]]
      |—WebExtensions—28*[{WebExtensions}]
      |—185*[{firefox}]
gogosoon@gogosoon-Inspiron-5515:~$
```

Listing processes in tree view for a particular process

Earlier, I mentioned that `pstree` command lists the processes from all the users. Passing the username along with the `pstree` command lists only the processes run by the user.

```
pstree root
```


Learn to code — free 3,000-hour curriculum

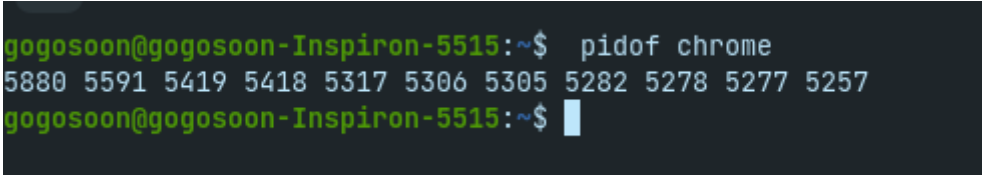
We know that every program (including the browser) runs as a process. So then you just have to find the process id and kill it.

Let's see how to find the process id of a command/program you need.

In my system, Chrome is running, Now we can get the PIDs of Chrome by running the following command:

```
pidof chrome
```

Terminal Command to find ID of a process



```
gogosoon@gogosoon-Inspiron-5515:~$ pidof chrome
5880 5591 5419 5418 5317 5306 5305 5282 5278 5277 5257
gogosoon@gogosoon-Inspiron-5515:~$
```

Terminal command to find process id of chrome

How to Kill a Process

There is a command called `kill` in Linux that is used to kill any process by passing the PID (Process id) or Process Name.

Here's the syntax of the `kill` command:

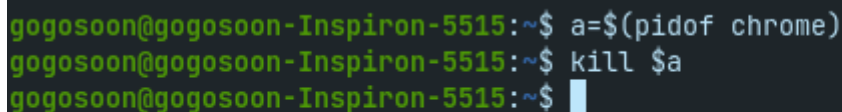
Learn to code — free 3,000-hour curriculum

Syntax for kill command

Let's store the PID of Chrome and kill it using the kill command:

```
a=$(pidof chrome)
kill $a
```

Command to kill a process

A terminal window with a dark background and green text. It shows three lines of commands being entered at a prompt: 'a=\$(pidof chrome)', 'kill \$a', and a blank line with a cursor.

```
gogosoon@gogosoon-Inspiron-5515:~$ a=$(pidof chrome)
gogosoon@gogosoon-Inspiron-5515:~$ kill $a
gogosoon@gogosoon-Inspiron-5515:~$
```

Terminal command to kill a process

The above command will kill the Chrome web browser.

How to List All Processes

We can see all the Linux processes using the `top` command. It shows real-time updates of each process for all users.

```
top
```

Terminal command to list all the processes in real-time

Learn to code — free 3,000-hour curriculum

MiB Swap: 3906.0 total, 3906.0 free, 0.0 used. 10457.6 avail Mem

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3202	gogosoon	20	0	4149796	575916	263380	S	23.5	3.7	15:16.58	firefox
2569	gogosoon	20	0	5788540	337860	130664	S	12.9	2.2	2:09.01	gnome-shell
2369	gogosoon	20	0	1449520	109428	61200	S	7.9	0.7	2:20.34	Xorg
3633	gogosoon	20	0	2954876	590364	157124	S	5.3	3.8	5:37.87	Isolated Web Co
6623	gogosoon	20	0	2641820	241252	128304	S	3.3	1.5	0:53.62	Isolated Web Co
248	root	-51	0	0	0	0	S	1.7	0.0	0:03.96	irq/37-DELL0A78
317	root	-2	0	0	0	0	S	1.0	0.0	0:22.53	gfx
1204	root	20	0	334416	14716	12608	S	0.7	0.1	0:02.28	touchegg
2283	gogosoon	9	-11	2604696	19144	15168	S	0.7	0.1	1:34.07	pulseaudio
3551	gogosoon	20	0	3089040	571820	130472	S	0.7	3.6	1:11.82	Isolated Web Co
3706	gogosoon	20	0	2675852	274352	105324	S	0.7	1.7	0:18.86	Isolated Web Co

Terminal command displaying all the process in real-time

Let's understand the heading to understand the underlying data.

- PID represents a Unique process ID.
- USER represents the Username of the owner of the task.
- PR represents the Priority of the process. Lower the number, higher the priority.
- NI represents a Nice Value of task. A Negative nice value implies higher priority, and a positive Nice value means lower priority.
- VIRT represents the total virtual memory used by the task.
- RES represents RAM Usage of a process in kilobytes.
- SHR represents Shared Memory Size (Kb) used by a process.
- S represents the Status of the process:
 - D: Uninterruptible sleep
 - R: Running
 - S: Sleeping
 - T: Traced (stopped)
 - Z: Zombie
- CPU represents the CPU usage.

Learn to code — free 3,000-hour curriculum

- **COMMAND** represents the command that used to start the process.

To display specific user processes we should use the flag `-u` :

```
top -u <username>
```

Terminal command syntax to list specific user's processes

To look at the processes run by the user `gogosoon` , run the following command:

```
top -u gogosoon
```

Terminal command to list processes started by user `gogosoon`

Learn to code — free 3,000-hour curriculum

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME	COMMAND
4105	gogosoon	20	0	4113576	558608	274676	S	31.9	3.6	2:49.75	firefox
4629	gogosoon	20	0	3088252	511120	116620	S	29.2	3.3	3:29.31	Isolated Web Co
6422	gogosoon	20	0	3145220	570032	254476	S	21.3	3.6	0:13.60	Isolated Web Co
2662	gogosoon	20	0	5738000	325588	127904	S	15.0	2.1	0:51.17	gnome-shell
2469	gogosoon	20	0	1457608	117900	61248	S	6.6	0.8	0:41.23	Xorg
4429	gogosoon	20	0	2684708	304844	114136	S	5.3	1.9	0:23.39	WebExtensions
2378	gogosoon	9	-11	2866840	19112	15132	S	4.3	0.1	0:24.63	pulseaudio
5882	gogosoon	20	0	2731056	234068	112764	S	3.3	1.5	0:09.45	Isolated Web Co
7013	gogosoon	20	0	821396	57172	43012	S	3.0	0.4	0:00.67	gnome-terminal-
4501	gogosoon	20	0	3148016	518604	130136	S	2.7	3.3	0:32.21	Isolated Web Co
5834	gogosoon	20	0	2586284	193796	111080	S	1.7	1.2	0:07.57	Isolated Web Co
3954	gogosoon	20	0	1129.9g	162092	108944	S	1.0	1.0	0:08.83	Pritunl
5790	gogosoon	20	0	2497428	159028	103988	S	1.0	1.0	0:05.34	Isolated Web Co
3882	gogosoon	20	0	1122.0g	141920	113476	S	0.7	0.9	0:01.57	Pritunl
5989	gogosoon	20	0	2445116	115812	92572	S	0.7	0.7	0:01.31	Isolated Web Co
2706	gogosoon	20	0	270072	26364	17040	S	0.3	0.2	0:00.97	ibus-extension-
5985	gogosoon	20	0	2502116	132768	100416	S	0.3	0.8	0:02.40	Isolated Web Co
6064	gogosoon	20	0	2506028	127760	92592	S	0.3	0.8	0:00.99	Isolated Web Co
6154	gogosoon	20	0	2662404	221752	106540	S	0.3	1.4	0:05.86	Isolated Web Co
6192	gogosoon	20	0	2434408	110740	90736	S	0.3	0.7	0:00.53	Isolated Web Co
6297	gogosoon	20	0	2434132	111776	91708	S	0.3	0.7	0:00.52	Isolated Web Co
7262	gogosoon	20	0	12124	3936	3144	R	0.3	0.0	0:00.25	top

Terminal output of all process started by user gogosoon

You might be confused about seeing the command line output 😊.
It'll be a bit hard to debug the processes in real time.

Here comes the handy GUI tool to handle the processes in Linux.
But we have to install this manually. This will work more like task manager in Windows.

```
sudo apt install gnome-system-monitor
```

Terminal command to install system monitoring app

After installing, just type the name of the software in terminal:

```
gnome-system-monitor
```

Command to open the List of process in GUI

Learn to code — free 3,000-hour curriculum

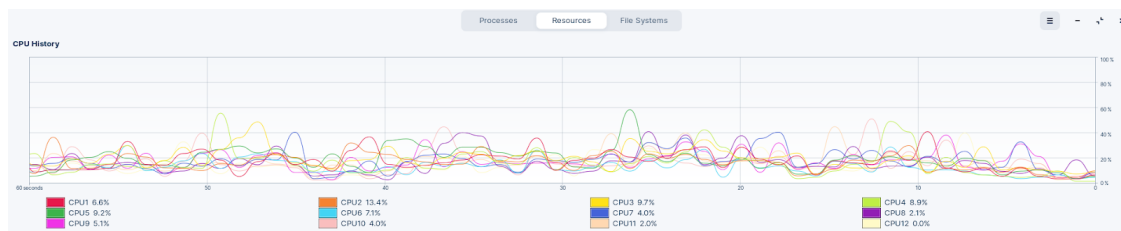
Processes									
Process Name	User	% CPU	ID	Memory	Disk read total	Disk write total	Disk read	Disk write	Priority
evolution-alarm-notify	gogosoon	0	2839	12.4 MiB	676.0 KiB	N/A	N/A	N/A	Normal
evolution-calendar-factory	gogosoon	0	2753	8.9 MiB	652.0 KiB	N/A	N/A	N/A	Normal
evolution-source-registry	gogosoon	0	2740	8.7 MiB	13.6 MiB	N/A	N/A	N/A	Normal
firefox	gogosoon	0	4105	270.1 MiB	793.6 MiB	188.5 MiB	N/A	14.7 KiB/s	Normal
gdm-x-session	gogosoon	0	2464	620.0 KiB	104.0 KiB	N/A	N/A	N/A	Normal
gjs	gogosoon	0	2789	5.0 MiB	N/A	N/A	N/A	N/A	Normal
nic	gogosoon	0	3059	10.0 MiB	356.0 KiB	N/A	N/A	N/A	Normal

Gnome-System-Monitor

When we right-click on any process it will show the actions like kill, stop, end, and so on.

The Resources tab shows the following utilities:

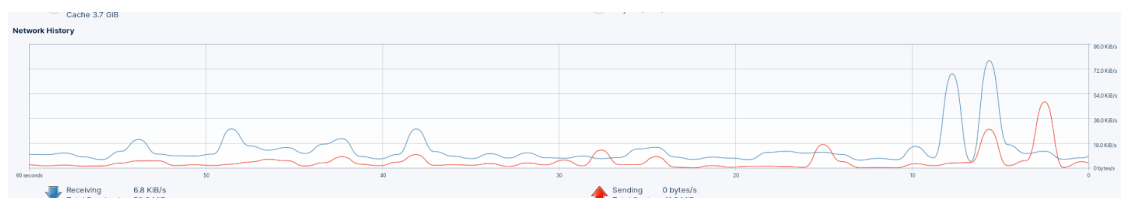
1. CPU History
2. Memory and Swap History
3. Network History



CPU History graph



Memory and Swap History Graph



Learn to code — free 3,000-hour curriculum

These graphs will be useful to determine the load in your system.

Conclusion

In this article, you have learned the basics of processes in Linux. I hope you now understand how they work a bit better. I recommend you all try these commands in your system.

To learn more about Linux, subscribe to my email newsletter at my [site](#) and follow me on social media.



Arunachalam B

Project Manager by Profession. Software Developer by Passion. Curious to explore technologies. Live in Chennai

If this article was helpful, [tweet it](#).

Learn to code for free. freeCodeCamp's open source curriculum has helped more than 40,000 people get jobs as developers.

[Get started](#)

ADVERTISEMENT

Learn to code — free 3,000-hour curriculum

freeCodeCamp is a donor-supported tax-exempt 501(c)(3) charity organization (United States Federal Tax Identification Number: 82-0779546)

Our mission: to help people learn to code for free. We accomplish this by creating thousands of videos, articles, and interactive coding lessons - all freely available to the public. We also have thousands of freeCodeCamp study groups around the world.

Donations to freeCodeCamp go toward our education initiatives, and help pay for servers, services, and staff.

You can [make a tax-deductible donation here](#).

Trending Guides

Python list .append()

C++ Operators

Python .zip() Function

Excel Formulas

What is a QA Engineer?

Break in Python

Python Print Same Line

Text Align in CSS

Rename Branches in Git

How to Minify CSS

What Does Coding Mean?

Python Split String

What is Data Analysis?

Python List insert()

How to Comment Out CSS

Merge Sort Algorithm

Double vs Float in C++

What is an SVG File?

[Forum](#)

[Donate](#)

[Learn to code — free 3,000-hour curriculum](#)

[JavaScript .setTimeout\(\)](#)

[How to Lock Cells in Excel](#)

[Excel Keyboard Shortcuts](#)

[Python Delete Key from Dict](#)

[Excel Absolute Reference](#)

[Beginner Tech Jobs Examples](#)

[Insert Checkbox in Excel](#)

[Crow’s Foot Notation](#)

Our Charity

- [About](#)
- [Alumni Network](#)
- [Open Source](#)
- [Shop](#)
- [Support](#)
- [Sponsors](#)
- [Academic Honesty](#)
- [Code of Conduct](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [Copyright Policy](#)