

C

GETTING HELP



Everyone gets stuck at some point when they're learning to program. So one of the most important skills to learn as a programmer is how to get unstuck efficiently. This appendix outlines several ways to help you get going again when programming gets confusing.

First Steps

When you're stuck, your first step should be to assess your situation. Before you ask for help from anyone else, answer the following three questions clearly:

- What are you trying to do?
- What have you tried so far?
- What results have you been getting?

Make your answers as specific as possible. For the first question, explicit statements like “I’m trying to install the latest version of Python on my Windows 10 laptop” are detailed enough for others in the Python community to help you. Statements like “I’m trying to install Python” don’t provide enough information for others to offer much help.

Your answer to the second question should provide enough detail so you won’t be advised to repeat what you’ve already tried: “I went to <https://python.org/downloads/> and clicked the Download button for my

system. Then I ran the installer,” is more helpful than “I went to the Python website and downloaded something.”

For the third question, it’s helpful to know the exact error messages you received so you can search online for a solution or provide them when asking for help.

Sometimes just answering these three questions before you ask for help from others allows you to see something you’re missing and get you unstuck without having to go any further. Programmers even have a name for this: it’s called *rubber duck debugging*. The idea is that if you clearly explain your situation to a rubber duck (or any inanimate object), and ask it a specific question, you’ll often be able to answer your own question. Some programming shops even keep a real rubber duck around to encourage people to “talk to the duck.”

Try It Again

Just going back to the start and trying again can be enough to solve many problems. Say you’re trying to write a `for` loop based on an example in this book. You might have only missed something simple, like a colon at the end of the `for` line. Going through the steps again might help you avoid repeating the same mistake.

Take a Break

If you’ve been working on the same problem for a while, taking a break is one of the best tactics you can try. When we work on the same task for long periods of time, our brains start to zero in on only one solution. We lose sight of the assumptions we’ve made, and taking a break helps us get a fresh perspective on the problem. It doesn’t need to be a long break, just something that gets you out of your current mind-set. If you’ve been sitting for a long time, do something physical: take a short walk or go outside for a bit; perhaps drink a glass of water or eat a light and healthy snack.

If you're getting frustrated, it might be worth putting your work away for the day. A good night's sleep almost always makes a problem more approachable.

Refer to This Book's Resources

The online resources for this book, available at <https://nostarch.com/pythoncrashcourse2e/>, include a number of helpful sections about setting up your system and working through each chapter. If you haven't done so already, take a look at these resources and see if there's anything that helps your situation.

Searching Online

Chances are that someone else has had the same problem you're having and has written about it online. Good searching skills and specific inquiries will help you find existing resources to solve the issue you're facing. For example, if you're struggling to install the latest version of Python on Windows 10, searching for *install python windows 10* and limiting the results to resources from the last year might direct you to a clear answer.

Searching the exact error message can be extremely helpful too. For example, say you get the following error when you try to start a Python terminal session:

```
> python
```

```
'python' is not recognized as an internal or external command,  
operable program or batch file
```

```
>
```

Searching for the full phrase “python is not recognized as an internal or external command” will probably yield some good advice.

When you start searching for programming-related topics, a few sites will appear repeatedly. I'll describe some of these sites briefly, so you'll know how helpful they're likely to be.

Stack Overflow

Stack Overflow (<https://stackoverflow.com/>) is one of the most popular question-and-answer sites for programmers, and will often appear in the first page of results on Python-related searches. Members post questions when they're stuck, and other members try to give helpful responses. Users can vote for the responses they find most helpful, so the best answers are usually the first ones you'll find.

Many basic Python questions have very clear answers on Stack Overflow, because the community has refined them over time. Users are encouraged to post updates too, so responses tend to stay relatively current. At the time of this writing, over one million Python-related questions have been answered on Stack Overflow.

The Official Python Documentation

The official Python documentation (<https://docs.python.org/>) is a bit more hit or miss for beginners, because its purpose is more to document the language than to provide explanations. The examples in the official documentation should work, but you might not understand everything shown. Still, it's a good resource to check when it comes up in your searches and will become more useful to you as you continue building your understanding of Python.

Official Library Documentation

If you're using a specific library, such as Pygame, Matplotlib, Django, and so on, links to the official documentation for that project will often appear in searches—for example, <https://docs.djangoproject.com/> is very helpful. If you're planning to work with any of these libraries, it's a good idea to become familiar with their official documentation.

r/learnpython

Reddit is made up of a number of subforums called *subreddits*. The *r/learnpython* subreddit (<https://reddit.com/r/learnpython/>) is fairly active and supportive. Here you can read others' questions and post your own.

Blog Posts

Many programmers maintain blogs and share posts about the parts of the language they're working with. You should skim the first few comments on a blog post to see what reactions other people have had before taking any advice. If no comments appear, take the post with a grain of salt. It's possible no one else has verified the advice.

Internet Relay Chat

Many programmers interact in real time through Internet Relay Chat (IRC). If you're stuck on a problem and searching online isn't providing answers, asking in an IRC channel might be a good option. Most people who hang out in these channels are polite and helpful, especially if you can be specific about what you're trying to do, what you've already tried, and what results you're getting.

Making an IRC Account

To create an account on IRC, go to <https://webchat.freenode.net/>. Choose a nickname, fill out the CAPTCHA box, and click **Connect**. You'll see a message welcoming you to the freenode IRC server. In the box at the bottom of the window, enter the following command:

```
/msg nickserv register password email
```

Enter your own password and email address in place of *password* and *email*. Choose a password that you don't use for any other account. You'll receive an email with instructions to verify your account. The email will provide you with a command like this:

```
/msg nickserv verify register nickname verification_code
```

Paste this line into the IRC site with *nickname* as the name you chose earlier and a value for *verification_code*. Now you're ready to join a channel.

If you have trouble logging into your account at some point, you can issue the following command:

```
/msg nickserv identify nickname password
```

Replace *nickname* and *password* with your own nickname and password. This will authenticate you on the network, and you'll be able to access channels that require an authenticated nickname.

Channels to Join

To join the main Python channel, enter `/join #python` in the input box. You'll see a confirmation that you joined the channel and some general information about the channel.

The channel `##learnpython` (with two hashtags) is usually quite active as well. This channel is associated with <https://reddit.com/r/learnpython/>, so you'll see messages about posts on `r/learnpython` too. You might want to join the `#django` channel if you're working on web applications.

After you've joined a channel, you can read the conversations other people are having and ask your own questions as well.

IRC Culture

To get effective help, you should know a few details about IRC culture. Focusing on the three questions at the beginning of this appendix will definitely help guide you to a successful solution. People will be happy to help you if you can explain precisely what you're trying to do, what you've already tried, and the exact results you're getting. If you need to share code or output, IRC members use external sites made for this purpose, such as <https://bpaste.net/+python>. (This is where `#python` sends you to share code and output.) This keeps the channels from being flooded with code and also makes it much easier to read the code that people share.

Being patient will always make people more likely to help you. Ask your question concisely, and then wait for someone to respond. Often,

people are in the middle of many conversations, but usually someone will address you in a reasonable amount of time. If few people are in the channel, it might take a while to get a response.

Slack

Slack is like a modern-day reinvention of IRC. It's often used for internal company communications, but there are also many public groups you can join. If you want to check out Python Slack groups, start with <https://pysslackers.com/>. Click the **Slack** link at the top of the page, and enter your email address to get an invitation.

Once you're in the Python Developers workspace, you'll see a list of channels. Click **Channels**, and then choose the topics that interest you. You might want to start with the *#learning_python* and *#django* channels.

Discord

Discord is another online chat environment with a Python community where you can ask for help and follow Python-related discussions.

To check it out, head to <https://pythondiscord.com/> and click the **Chat Now** link. You should see a screen with an automatically generated invitation; click **Accept Invite**. If you already have a Discord account, you can log in with your existing account. If you don't have an account, enter a username and follow the prompts to complete your Discord registration.

If this is your first time visiting the Python Discord, you'll need to accept the rules for the community before participating fully. Once you've done that, you can join any of the channels that interest you. If you're looking for help, be sure to post in one of the Python Help channels.