# The call function

How to call another process in python?

The subprocess module provides a function named **call**. This function allows you to call another program, wait for the command to complete and then return the return code. It accepts one or more arguments as well as the following keyword arguments (with their defaults): stdin=None, stdout=None, stderr=None, shell=False.

Let's look at a simple example:

```
import subprocess
subprocess.call("ls")
```

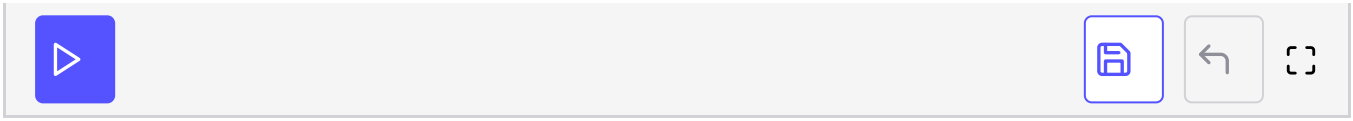
If you run this on a Windows machine, you should see Notepad open up. You will notice that IDLE waits for you to close Notepad and then it returns a code zero (0). This means that it completed successfully. If you receive anything except for a zero, then it usually means you have had some kind of error.

Normally when you call this function, you would want to assign the resulting return code to a variable so you can check to see if it was the result you expected. Let's do that:

```
import subprocess
code = subprocess.call("wget")
if code == 0:
    print("Success!")
else:
    print("Error!")
```

If you run this code, you will see that it prints out **Error!** to the screen. The **call** method also accepts arguments to be passed to the program that you're executing. Let's see how that works:

```
import subprocess
code = subprocess.call(["wget", "www.facebook.com"])
```



You will notice that in this example we are passing a list of arguments. The first item in the list is the program we want to call. Anything else in the list are arguments that we want to pass to that program. So in this example, we are executing a ping against Yahoo's website. You will notice that the return code was zero, so everything completed successfully.

You can also execute the program using the operating system's **shell**. This does add a level of abstraction to the process and it raises the possibility of security issues. Here is the Python documentation's official warning on the matter:

**Executing shell commands that incorporate unsanitized input from an untrusted source makes a program vulnerable to shell injection, a serious security flaw which can result in arbitrary command execution. For this reason, the use of shell=True is strongly discouraged in cases where the command string is constructed from external input.**

The usual recommendation is not to use it if an outside process or person can modify the call's arguments. If you're hard-coding something yourself, then it doesn't matter as much.