

Welcome to the very first in a series of videos for problem solving and programming on Python. In this video we'll take a look at IDLE, the IDE that we'll be using to code Python, and go over everything you'll need to know to get up and running with IDLE and executing your very first Python program. Let's get into it. When you load IDLE you'll be presented with the shell, much like you see here. Notice that this is Python 3.6.1, as you can see in the top left corner, and that's what we'll be using throughout this course. IDLE is an integrated development environment or IDE. It provides you with the tools that you need to write, execute, and test a program. Everything you need for this course. You'll be using IDLE throughout this course for all of your Python code and all the examples we provide will be shown in IDLE as well. What you can see here is the shell that's part of IDLE. Getting started with Python you'll see that we have a prompt here, a flashing cursor in front of three greater than arrows. This tells us that the interpreter is ready to receive commands and provide us output based on those commands. The most simple thing you could do with the shell is to use it like a calculator. We could do five plus two and press ENTER and it will give us the answer, which of course is seven. We could do multiplication, for example five times two, which would give us ten. Or something a little bit more complicated, five to the power of two, which is 25. We could also put in some basic Python functions, for example, the print function will print something to the screen. We could say `print 'this is fun'` and you'll see that the shell has output 'this is fun'. We could also do simple variable assignments, which is something we'll get into in a later video, but we could save the output of five plus two to some variable that we've decided to name 'a'. 'a' is the result of five plus two, and then we could use the print function again to output what that five plus two evaluates to, which again of course is seven. If we wanted to exit the shell we could type `quit`. Notice this is a function so we need the two brackets like this, and press Enter. And then if we wanted to, we could exit Python. We'll say cancel at this point because you could also use `exit` to have the same result. Of course using the

shell is great for quickly testing functions or to do some quick calculations but if you want to be able to save your work and continue at a later stage you'll want to be able to create a new file. In order to do this we need to use the editor view. To get to the IDLE editor from the shell, click file, new file. This will open a new window containing the text editor. Next you want to enter your program into the editor. During this course it's expected that you'll start every program with this following comment.

This is used to identify the title of your file, your name, as in who authored this file, your student ID, your email ID, today's date, a brief description about what the program does, and this following academic misconduct line. Next you would enter your actual program in Python code. For instance, we could decide to print a message to the screen, much like we did in the shell before, so we could print welcome to problem solving and programming. Then we would save this, and as our comment at the top said we were going to call it welcome.py. Once we've saved this, we can go to run and run module and that will open in the shell where we can see that it outputted the string that we entered in, in the print function, welcome to problem solving and programming. While we're in the editor let's have a look at some of the features of IDLE. Firstly you'll note that comments in Python are started with a hash and they appear as red in the IDLE editor. Comments are ignored by Python and are not executed as code, they are merely for the reader's benefit. Also notice that the print function name has appeared in purple and the string is in green. This is called syntax highlighting and it helps us to identify parts of our program from other parts. There are a few style things that we should cover as well. So there should be a space between the hash that starts the comment and the comment itself. Each line of the comment, unless it's a continuation, should start with a capital letter, and each line should be terminated with a full-stop. We can also use comments to identify major parts of our program. If we want to explain some logic we can do that before the statement. For example we could put a comment here that says