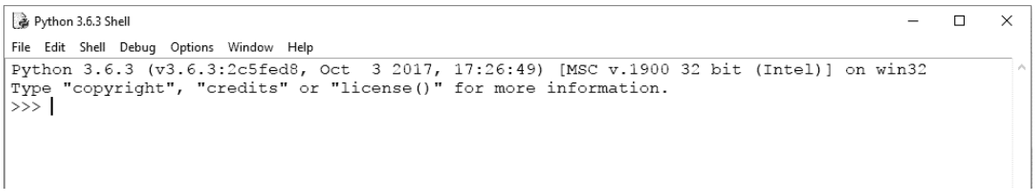
The Basics

Explanation

This is the **shell** window and is the first screen you see when you launch Python.

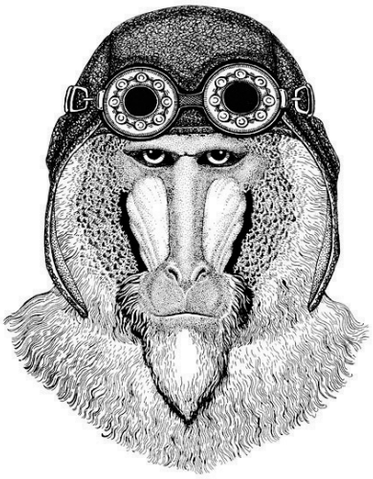


It is possible to write Python code straight into the shell, but as soon as you hit [Return] at the end of a line, it will run that line of code. This may be suitable for using Python as a quick calculator; for instance, you can type in **3*5** at the prompt and Python will show the answer **15** on the next line; however, this style of inputting is not useful for more complex programs.

It is much better to start a new window, create all the code in the new window, save your code and run it.

To create a new window in which to write your code, select **File** and **New**. Once you enter your code in this new window you can save it and run it all in one go. This will then run the code in the shell window.

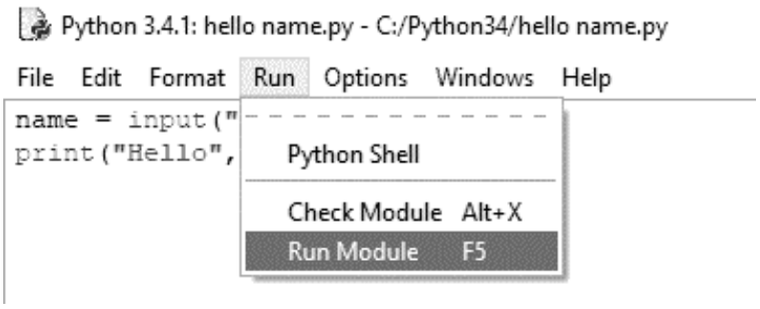
Alternatively, Python programs can be written using any text editor and must be saved with the file name extension .py in order to work. These programs can then be run from the command prompt by typing in the full directory root and file name.



Running Your Program

Every time you run the code your program will need to be saved afresh in case there have been any changes to it.

In this version of Python, you can run the program by selecting the **Run** menu and selecting **Run Module**. Alternatively, you can press the **[F5]** key. If this is the first time the program is saved, Python will prompt you to name and save the file before it will allow the program to run.



Important Things to Note When Writing Your Programs

Python is case sensitive so it is important that you use the correct case, otherwise your code will not work.

Text values need to appear in speech marks (") but numbers do not.

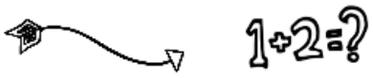
When naming **variables** (i.e. values that you want to store data in) you cannot use any dedicated words such as print, input, etc. otherwise your code will not work.

When saving your files **do not save them with any dedicated words** that Python already uses, such as print, input, etc. If you do this it will not run and you will need to rename the file before it works.

To edit a program you have saved and closed, right-click on the file and select **Edit with IDLE**. If you just double-click on the file it will only try to run it and you will not be able to edit it.

Example Code

num1 = 93
Set the value of a **variable**, if there is not a variable already created, it will create one. A variable is a container for a value (in this case the variable will be called "num1" and store the value 93). The value stored in the variable can change while the program is running. The variable can be called whatever you want (except Python dedicated words such as print, save, etc.) and it must start with a letter rather than a number or symbol and have no spaces.



answer = num1 + num2
Adds together num1 and num2 and stores the answer in a variable called answer.

answer = num1 - num2
Subtracts num2 from num1 and stores the answer in a variable called answer.

answer = num1 * num2
Multiplies num1 by num2 and stores the answer in a variable called answer.

answer = num1 / num2
Divides num1 by num2 and stores the answer in a variable called answer.

answer = num1 // num2
A whole number division (i.e. 9//4 = 2) and stores the answer in a variable called answer.

print ("This is a message")
Displays the message in the brackets. As the value we want displayed is a text value it has the speech marks, which will not be displayed in the output. If you wanted to display a numerical value or the contents of a variable, the speech marks are not needed.

print ("First line\nSecond line")
"\n" is used as a line break.

print ("The answer is", answer)
Displays the text "The answer is" and the value of the variable answer.

textValue = input("Enter a text value: ")
Displays the question "Enter a text value: " and stores the value the user enters in a variable called textValue. The space after the colon allows a space to be added before the user enters their answer, otherwise they appear squashed unattractively together.

numValue = int(input("Enter a number: "))
Displays the question "Enter a number: " and stores the value as an integer (a whole number) in a variable called numValue Integers can be used in calculations but variables stored as text cannot.

Challenges

001

Ask for the user's first name and display the output message **`Hello [First Name]`**.

002

Ask for the user's first name and then ask for their surname and display the output message **`Hello [First Name] [Surname]`**.

003

Write code that will display the joke "What do you call a bear with no teeth?" and on the next line display the answer "A gummy bear!" Try to create it using only one line of code.

004

Ask the user to enter two numbers. Add them together and display the answer as **`The total is [answer]`**.

005

Ask the user to enter three numbers. Add together the first two numbers and then multiply this total by the third. Display the answer as **`The answer is [answer]`**.

006

Ask how many slices of pizza the user started with and ask how many slices they have eaten. Work out how many slices they have left and display the answer in a user-friendly format.

007

Ask the user for their name and their age. Add 1 to their age and display the output **`[Name] next birthday you will be [new age]`**.

008

Ask for the total price of the bill, then ask how many diners there are. Divide the total bill by the number of diners and show how much each person must pay.

009

Write a program that will ask for a number of days and then will show how many hours, minutes and seconds are in that number of days.

010

There are 2,204 pounds in a kilogram. Ask the user to enter a weight in kilograms and convert it to pounds.

011

Task the user to enter a number over 100 and then enter a number under 10 and tell them how many times the smaller number goes into the larger number in a user-friendly format.

Answers

001

```
firstname = input("Please enter your first name: ")
print ("Hello",firstname)
```

002

```
firstname = input("Please enter your first name: ")
surname = input("Please enter your surname: ")
print ("Hello",firstname, surname)
```

003

```
print("What do you call a bear with no teeth?\nA gummy bear!")
```

004

```
num1 = int(input("Please enter your first number: "))
num2 = int(input("Please enter your second number: "))
answer = num1 + num2
print("The answer is", answer)
```

005

```
num1 = int(input("Please enter your first number: "))
num2 = int(input("Please enter your second number: "))
num3 = int(input("Please enter your third number: "))
answer = (num1 + num2)* num3
print("The answer is", answer)
```

006

```
startNum = int(input("Enter the number of slices of pizza you started with: "))
endNum = int(input("How many slices have you eaten? "))
slicesLeft = startNum - endNum
print("You have", slicesLeft, "slices remaining")
```

007

```
name = input("What is your name? ")
age = int(input("How old are you? "))
newAge = age + 1
print(name, "next birthday you will be", newAge)
```

008

```
bill = int(input("What is the total cost of the bill? "))
people = int(input("How many people are there? "))
each = bill/people
print("Each person should pay £", each)
```

009

```
days = int(input("Enter the number of days: "))
hours = days*24
minutes = hours*60
seconds = minutes*60
print("In", days,"days there are...")
print(hours, "hours")
print(minutes, "minutes")
print(seconds, "seconds")
```

010

```
kilo = int(input("Enter the number of kilos: "))
pound = kilo * 2.204
print("That is", pound,"pounds")
```

011

```
larger = int(input("Enter a number over 100: "))
smaller = int(input("Enter a number under 10: "))
answer = larger//smaller
print(smaller,"goes into", larger, answer,"times")
```

If Statements

Explanation

`If statements` allow your program to make a decision and change the route that is taken through the program.

Below is how the if statement for this flow chart would look in Python.

```
if num1 > 10:
    print("This is over 10")
elif num1 == 10:
    print("This is equal to 10")
else:
    print("This is under 10")
```

Indenting Lines of Code

Indenting is very important in Python as it shows the lines that are dependent on others, as shown in the example on the previous page. In order to indent text you can use your [**Tab**] key or you can press your [**space key**] five times. The [**backspace**] key will remove indents.

The first line of the if statement tests a condition, and if that condition is met (i.e. the first condition is true) then the lines of code directly below it are run. If it is not met (i.e. the first condition is false) it will test the second condition, if there is one, and so on. Below are examples of the different comparison and logical operators you can use in the condition line of your if statement.

Comparison Operators

Operator	Description
==	Equal to
!=	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

Logical Operators

Operator	Description
and	Both conditions must be met
or	Either condition must be met

Example Code

Please note: In the examples shown, num is a variable entered by the user that has been stored as an integer.

```
if num > 10:
 print("This is over 10")
else:
 print("This is not over 10")
```

If num1 is over 10, it will display the message "This is over 10", otherwise it will display the message "This is under 10".

```
if num > 10:
 print("This is over 10")
elif num == 10:
 print("This is equal to 10")
else:
 print("This is under 10")
```

If num1 is over 10, it will display the message "This is over 10", otherwise it will check the next condition. If num1 is equal to 10, it will display the message "This is equal to 10". Otherwise, if neither of the first two conditions have been met, it will display the message "This is under 10".

```
if num >= 10:
 if num <= 20:
  print("This is between 10 and 20")
 else:
  print("This is over 20")
else:
 print("This is under 10")
```

If num1 is 10 or more then it will test another if statement to see if num1 is less than or equal to 20. If it is, it will display the message "This is between 10 and 20". If num1 is not less than or equal to 20 then it will display the message "This is over 20". If num1 is not over 10, it will display the message "This is under 10".

```
text = str.lower(text)
```

Changes the text to lower case. As Python is case sensitive, this changes the data input by the user into lower case so it is easier to check.

```
num = int(input("Enter a number between 10 and 20: "))
if num >= 10 and num <= 20:
 print("Thank you")
else:
 print("Out of range")
```

This uses **and** to test multiple conditions in the if statement. Both the conditions must be met to produce the output "Thank you".

```
num = int(input("Enter an EVEN number between 1 and 5: "))
if num == 2 or num == 4:
 print("Thank you")
else:
 print("Incorrect")
```

This uses **or** to test the conditions in the if statement. Just one condition must be met to display the output "Thank you".