



2 ways to run Python commands

Type	Description	Example
<p>Interactive shell</p> <p>>>></p> <p>Commands interpreted immediately</p>	<ul style="list-style-type: none"> • Enter commands at the prompt • Press <enter> to run • ... means that the command has not been fully entered 	
<p>Shell window</p> <p>commands saved in a file to be used again</p>	<ul style="list-style-type: none"> • File/new window to open shell window • Enter python commands in to the file • File/save and give the program a meaningful file name (MUST be type .py) • Run/run module (or F5) • Output is shown in the Python shell 	

How to...	Explanation
Save a file (must be type .py)	File > Save
Run a program	Run > Run module or F5
Display the last command entered in the shell	alt p
Repeat the last command entered in the shell	alt n
Indent and dedent blocks of code	Select and use Format > Indent /Dedent
Interrupt a program that is running	control z or control c

Python data types

Data type	Python name	Explanation	Example
integer	int	A whole number.	45
string	str	A sequence of characters.	"Hello"
Float (real)	float	A number with a fractional part.	16.76
Boolean (logical)	bool	Boolean values can only have one of two values: True or False.	True False

Precedence B E D M A S

Parentheses () control the order in which expressions are calculated. The precedence order is: parenthesis (round **B**rackets), **E**xponential, **D**ivision and **M**ultiplication, **A**ddition and **S**ubtraction . B E D M A S

A **variable** is a named location in a computer's memory where data is stored.

Rules for variable names

Must begin with a letter (upper or lower case) followed by zero or more other letters or numbers.

Cannot have spaces in the name and cannot use reserved Python command words.

variableName = "a string of characters"

variableName = 99

variableName = 87.54

variableName = True

Mathematical operator symbol	Operation	Example
/	divide	99 / 5
*	multiply	7 * 5
+	add	5 + 9
-	subtract	100 - 90
**	exponential	6 ** 2
//	integer division – how many times can 5 go into 27 exactly	27 // 5 (answer 5)
%	modulus (remainder after the division)	27 // 5 (answer 2)

IDLE colour coding	What does it show	Example
green	strings	"hello"
purple	functions	print()
black	variables and data	myName
orange	key commands	if
Dark red	comments	# This is a comment
red	Error messages	NameError: name 'jfkd' is not defined

Built-in functions		
Syntax	Description	Example
print()	Displays information on the screen.	print ("Hello world")
type()	Displays the type (int, bool, str or float) of a variable or value.	type (ans) <class 'float'>
int()	Converts a string or float value into an integer number.	ans=7.8 number = int (input ("Please enter number))
input("prompt")	Prompts for input. The data entered is assigned to a variable.	reply=input("Enter your name: ") Enter your name: Fred



Comparison operator	Operator	Example	Evaluates to
Equal to	==	"fred" == "sid"	False
Not equal to	!=	8 != 8	False
Greater than	>	10 > 2	True
Greater than or equal to	>=	5 >= 5	True
Less than	<	40 < 34	False
Less than or equal to	<=	2 <= 109	True

Selection (if)		
Syntax	Description	Example
if <expression>: <commands>	If <expression> is true then commands are executed.	if colour == "green": print("It is safe for you to cross.")
if <expression>: <commands1> else: <commands2>	If <expression> is true then <commands1> are executed otherwise <commands2> are executed	if colour == "green": print("It is safe for your to cross.") else: print("STOP! It is not safe to cross.")
if <expressionA>: <commands1> elif <expressionB>: <commands2> elif <expressionC>: <commands3> else: <commands4>	If <expressionA> is true then <commands1> are executed, else if <expressionB> is true then <commands2> are executed, etc else <commands4 are executed.	if answer == 1: print("You will make a new friend this week.") elif answer == 2: print("You will do well in your GCSEs.") elif answer == 3: print("You will find something you thought you'd lost.")

String commands	Explanation
stringa in stringb	Returns true if stringa is in stringb
stringa not in stringb	Returns true is stringa is not in stringb
len(stringa)	Returns the length of the string
stringa + stringb	Concatenates stringa and stringb
stringa * number	Concatenates stringa by "number" times
str.isalnum()	Returns true if all the characters are alphanumeric otherwise false
str.isdigit()	Returns true is all the characters are digits
str.islower()	Returns true if all lowercase
str.replace(old,new)	Returns a copy of the string with occurrences of old replaced by new
str.find(sub)	Returns the lowest index in the string of "sub"

String formatting method

The string formatting commands are given in curly brackets {} which act as placeholders which start at 0.

Example:

```
>>> print("The answers are {0} {1} {2}".format("xyz","abc",100))
The answers are xyz abc 100

>>> number = 1329.78651983
>>> print("The answer is {0:.5f}".format(number))
The answer is 1329.78652
>>> print("The answer is {0:.2f}".format(number))
The answer is 1329.79
```

String indexes

The position of each character in a string is called the index, which starts from 0

Items in a string can be referenced **[start index: end index]** The end index is **up to but not including** that index. Negative indexes refer to the end of the string

```
>>> mystr="HELLO"
>>> mystr[2]
'L'
>>> mystr[1:3]
'EL'
>>> mystr[-1]
'O'
>>> mystr[-3:-1]
'LL'
```

H	E	L	L	O
0	1	2	3	4
-5	-4	-3	-2	-1

Python errors	Description
TypeError	When an operation is attempted that is invalid for that type of data.
RuntimeError	An error that occurs when the program is running.
NameError	When a name is used that is not known about (often a misspelt variable name).
ZeroDivisionError	Dividing a number by zero.
KeyBoardInterrupt	When a program is interrupted from the keyboard by pressing control+c

Reserved Python command words

and	exec	not
assert	finally	or
break	for	pass
class	from	print
continue	global	raise
def	if	return
del	import	try
elif	in	while
else	is	with
except	lambda	yield

Lists are a mutable, data structure which can contain any type of data at indexed locations within the list.

The index locations start from zero and are referred to using integer numbers.

red	green	blue	white	orange
0	1	2	3	4

Items in a list can be referenced: **[start index: end index]**

The end index is the index **after** the one you want the range to finish on It is **up to but not including** the end index.

[1:3] refers to items at index 1 and 2 which is “green” and “blue”

Syntax	Description	Example
listName = [value,value,value]	Assigns a set of values to a list.	myList= ["apple", "oranges",8]
listName[index]	Identifies an element of a list by reference to its position in the list, where index is an integer value starting at 0.	myList[2]
print(listName)	Displays a list on the screen	print(listName)
listName[start_index:end_index]	Reference an item in a list	print(listName[2:6] will display list items at locations 2 to 5
del listName[index] del listName[start_index:end_index]	Delete an item or range of items in a list	del listName[3] del listName[5:9]

Useful List methods

list.append(x)	Appends the item x at the end of the list
list.sort()	Sorts the list (need to be a sortable data type)
list.reverse()	Reverses the elements in the list
list.count(x)	Counts the number of times x appears in a list
list.index(x)	Returns the index at which x appears
list.insert(index,x)	Inserts an item x at position before index

Range(start, stop, step)

Numbering starts at 0 and ends the item before “stop”

range(40)	Numbers from 0 to 39
range(5,15)	Numbers from 5 to 15
Range(10,100.5)	Numbers from 10 to 100 in steps of 5

Repetition		
Syntax	Description	Example
for variable in <expression>: <commands>	Executes <commands> for a fixed number of times, given by <expression>.	myList=["cat", "dog", "cow", "donkey", "rabbit", "canary"] for next in myList: print(next)
while <condition>: <commands>	Executes <commands> whilst <condition> is true. This is a pre-condition loop.	answer="N" counter=0 while answer != "Y": print("Are you hungry? You have been asked {0} times.".format(counter)) answer = input("Please respond Y or N:") counter = counter + 1 print("Please get something to eat!")

Symbol	Description
AND	Returns true if both conditions are true.
OR	Returns true if one of the conditions is true.
NOT	Reverses the outcome of the expression; true becomes false, false becomes true.

FUNCTION Terminology	
Function	A subprogram that returns a value.
Return value	The value returned by a subprogram
Parameter	The names that appear in a function definition when passing data to a function.
Argument	A piece of information/value that is required by a function to perform a task
To define a function	def functionName(parameter, parameter)
To call a function	functionName(argument, argument)

IDLE Debugger	
<ul style="list-style-type: none"> Open Python IDLE with an existing program file. Start the IDLE debugger using Debug/Debugger. The IDLE shell will display [DEBUG ON] to show you are in debug mode. To turn off this mode select Debug/Debugger. Start running your program. The Debug Control window will be displayed. 	