Hack the Ram
Python Cheat Sheet

Function	Example	Output	Description
print(*string*)	print("Hello World") print("Hello", "World") print("Hello " + "World")	Hello World	This function allows the programmer to output data to the console for the user to see.
input(*string*)	name = input("What is your name?") age = input("What is your age?")	What is your name? What is your age?	This function allows the programmer to take input from the user and store into a variable.
var = *datatype*	var = "Hello World" var = 42 var = True	*"Hello World"* *42* *True*	Using a variable name that starts with a non-number/special-character name, the programmer can store data to be used later in the program.
str(*datatype*)	str(42) str(True) str(42.42)	"42" "True" "42.42"	Allows the programmer to change one datatype to another datatype. -----------------------------
int(*string or float*)	int(42.42) int("42")	42 42	- Integers are whole numbers - Floats are integers but have a decimal place
float(*integer or string*)	float(42)	42.0	- Strings are text that has no numerical value unless converted

"Python is an experiment in how much freedom programmers need."
–Guido van Rossum

Numerical Operator	Meaning	Example
+	Addition	40 + 2 = 42
-	Subtraction	44 - 2 = 42
/	Division	42 / 2 = 21
*	Multiplication	21 * 2 = 42
**	Exponent	2 ** 2 = 4
//	Floor Division	43 // 2 = 21
%	Remainder	43 % 2 = 1

Logical Operator	Meaning	Example
==	Equal To	1 == 1? **True**
!=	Not Equal To	1 != 1? **False**
>	Greater Than	1 > 4? **False**
<	Less Than	1 < 3? **True**
>=	Greater or Equal To	1 >= 1? **True** 1 >= 2? **False**
<=	Less Than or Equal To	1 <= 1? **True** 1 <= 2? **True**
not	not	not True? **False**
and	and	True and False? **False**
or	or	True or False? **True**

Decision Statements: if/elif/else:

```
age = int(input("What is your age?"))
requirement = 16

if age >= requirement:
        print("You are allowed to drive.")
elif age < 0:
    print("You cannot be", age, "years old.")
else:
    print("You are not allowed to drive.")
```

IF STATEMENT SYNTAX NOTES:
- You MUST have a colon after the logic statement.
- Any code that you want to execute because of the `if` statement, MUST be indented 1 tab.
- `elif` can be used if you want another if statement to be checked if the ABOVE is False.
- Else can be used if none of the `if` or `elif` statements are True.

```
Example:
What is your age? 14
You are not allowed to drive.
----------------------------
What is your age? 16
You are allowed to drive.
----------------------------
What is your age? -4
You cannot be -4 years old.
```

My code isn't working :-(

Start here...

Do you get an error when you run the code?

yes → **What type of error do you get?**

Error types

Attribute Error
You are calling a method on the wrong type of object

SyntaxError
You've forgotten the quotes around a string

You have forgotten to put a colon at the end of a def/if/for line

You have different number of open and close brackets in a statement

TypeError
You're trying to use an operator on the wrong type of objects

An object which you expect to have a value is actually None

You've used non-integer numbers in a list slice

You've called a method/function with the wrong number or type of arguments

Indentation Error
You've used a mixture of tabs and spaces

You haven't indented all lines in a block equally

NameError
You've misspelt a variable, function or method name

You've forgotten to import a module

You've forgotten to define a variable

Your code uses a variable outside the scope where it's defined

Your code calls a function before it's defined

You're trying to print a single word and have forgotten the quotes

IOError
You're trying to open a file that doesn't exist

KeyError
You're trying to look up a key that doesn't exist in a dict

http://pythonforbiologists.com

---

no → **Does the code use loops or if statements?**

if
Two numbers which should be equal are not
You are comparing a number with a string representation of a number (e.g. if 3 == "3")

A complex condition is not giving the expected result
The order of precedence in the condition is ambiguous - add some parentheses

loops
A list which should have a value for every iteration only has a single value
You have defined the list inside the loop: move it outside

A loop which uses the range function misses out the last value
The range function is exclusive at the finish: increase it by one.

I am trying to loop over a collection of strings, but am getting individual characters
You are iterating over a string by mistake

I am trying to write multiple lines to a file but only getting a single one
You have opened the file inside the loop: move it outside

neither / also check...

A variable that should contain a value does not
You are storing the return value of a function which changes the variable itself (e.g. sort)

A number which should be a fraction is coming out as zero in Python 2
You are dividing integers rather than floats. Convert the numbers to floats or from __future__ import division

I'm trying to print a value but getting a weird-looking string
You are printing an object (e.g. a FileObject) when you want the result of calling a method on the object

A regular expression is not matching when I expect it to
You have forgotten to use raw strings or escape backslash characters

I am reading a file but getting no input
You have already read the contents of the file earlier in the code, so the cursor is at the end.