Python Cheat Sheet

The `print()` Function

The `print()` function prints the specified message to the screen, or other output device. The message can be a string, or any other object that is converted into a string before being written to the screen.

```
print("Hello World!")
```

Printing a math solution

```
x = 9
y = 2
print("Sum: " + str(x + 9))
```

Data Types and Type Conversion

Integers

```
-2, -1, 0, 1, 2, 3, 4, 5
int()
```

Floats

```
-1.25, -1.0, -0.5, 0.0, 1.0
float()
```

Strings

```
"Hello", "This is a string."
str()
```

Comments

Inline Comment

```
# This is a comment.
```

Multiline Comment

```
# This is a
# multiline comment.
```

Code with Comment

```
a = 1    #initialization
```

Variables

Variables can be named anything as long as:

- It is only one word.
- Only uses letters, numbers, and the underscore character.
- It can't begin with a number.
- Starting with an underscore is considered "unuseful."

```
name = Alice
```

Input

Your programs can prompt the user for input. All input is stored as a string.

Prompting for a String

```
name = input("Who are you?")
print("Hello " + name)
```

Prompting for a Value

```
age = int(input("How old are
    you? "))
print(age)
```

Calculations with Variables

Math operators follow order of operations.

Exponent

```
**          2 ** 3 = 8
```

Modular Division

```
%           22 % 8 = 6
```

Division

```
/           22 / 8 = 2.75
```

Multiplication

```
*           3 * 3 = 9
```

Subtraction

```
-           5 - 2 = 3
```

Addition

```
+           2 + 2 = 4
```

Math Functions

```
import math
```

Square Root

```
math.sqrt()
```

Absolute Value

```
math.fabs()
```

Raising to a Power

```
math.pow(x, y)
```

Random Numbers

```
import random
```

Random Integer between x and y

```
random.randint(x, y)
```

Random Number from 0 to .999999999999

```
random.random()
```

Specifying a Seed for a sequence of Random Numbers

```
random.seed(x)
```

Pick a random element from a sequence

```
animal = random.choice(["cat",
"dog", "fish", "snake"])
```

Min and Max

min and max are functions in Python that can find the minimum or maximum of a list of numbers.

min()

```
min(4, 6, 2, 7, 1, 9)
```

max()

```
max(4, 6, 2, 7, 1, 9)
```

Conditional Tests

equals		greater than or equal to	
`==`	`x == 42`	`>=`	`x >= 42`
not equal		less than	
`!=`	`x != 42`	`<`	`x < 42`
great than		less than or equal to	
`>`	`x > 42`	`<=`	`x < 42`

Simple Plot

The first parameter ('Sample') is the title. The second and third are the width (400) and height (300) of the graph. The fourth and fifth label the x and y axes. The next parameter contains our x and y values. The last two are optional. The True in this example says that we want to indicate the points on our graph and the last parameter gives a legend for the graph.

```
import simpleplot

dataset1 = [(1, 4), (1, 5), (2,
    7), (4, 9)]
dataset2 = [(1, 2), (2, 7), (2,
    5), (7, 6)]

simpleplot.plot_lines('Sample',
    400, 300, 'x', 'y',
    [dataset1, dataset2], True,
    ['dataset1', 'dataset2'])
```

Boolean Operators

You can check multiple conditions at the same time.

and

```
n = int(input("a number: "))
if (n >= 0 and n <= 100):
    print("Grade is valid")
print("Done")
```

or

```
x = -5
y = 10
if (x < 0 or y < 0):
    print("x or y are negative")
```

not

```
x = 1
if (x > 0 not x == 10):
    print("Correct")
```

If Statements

Several kinds of if statements exist. Your choice of which to use depends on the number of conditions you need to test.

Simple if Statement

```
age = 19
if (age >= 18):
     print("You're old enough
    to vote!")
```

If-else Statement

```
age = 17
if (age >= 18):
    print("You're old enough to
    vote!")
else:
    print("You can't vote
    yet.")
```

Else-If Statement

```
age = 12
if age < 4:
    price = 0
elif (age < 18):
    price = 5
else:
    price = 10
```