

Python Syntax Booklet

THE BASICS

Variables

```
my_string_variable = "hello world"
```

```
my_integer_variable = 7
```

```
my_float_variable = 2.75
```

```
my_boolean_variable = True
```

Printing to the screen

```
print("Hello world")
```

Maths

```
result = 1 + 1
```

```
print(result)    # prints 2
```

```
result = 2 - 1
```

```
print(result)    # prints 1
```

```
result = 2 * 2
```

```
print(result)    # prints 4
```

```
result = 15 / 3
```

```
print(result)    # prints 5
```

```
# Python uses BIDMAS:
```

```
result = 1 + 2 * 3
```

```
print(result)    # prints 7
```

```
result = (1 + 2) * 3
```

```
print(result)    # prints 9
```

IF Statements

```
my_float_variable = 2.75
```

```
if my_float_variable < 3.0:  
    print("The number was less than 3.0")  
  
elif my_float_variable > 5.0:  
    print("The number was greater than 5.0")  
  
else:  
    print("The number was between 3.0 and 5.0")
```

Comments

```
# Comments start with a # symbol  
  
# They are ignored by python  
  
# They help other people understand your code
```

User Input

```
your_name = input("Please Enter your name: ")
```

Converting Types

```
my_string_seven = "7" # This is a string  
  
# int() converts to an integer if it can  
my_integer_seven = int(my_string_seven)  
  
# also available is:  
#     str()    - to convert to string  
#     float()  - to convert to float
```

Loops

```
while True:      # Loop Forever
    print("hello world")
```

```
    # PRESS CTRL-C TO STOP PROGRAM!
```

```
for counter in range(10):    # repeat 10 times
    print( "iteration number {}".format(counter) )
```

```
while True:
    option = input("Enter q to quit: ")

    if option == 'q':
        break                # repeat-until
```

FORMATTING STRINGS

Multi-line strings

```
my_multi_line_string = """ if you want to make
a string go over multiple lines,
you can use 'triple-double-quotes' """
```

Printing the value of a variable inline with text

```
my_name = "Harambe"
```

```
print("Hello {}, how are you?".format(my_name))
```

```
# use {} as a placeholder, then pass the variable
# to the .format() function for each placeholder
```

More String Formatting

```
# String Formatting
```

```
username = 'm.watts'
```

```
#Printing a variable
```

```
print( "Welcome {}, how are you".format(username) )
```

```
average_sweat_ml = 12.772497532
```

```
# formatting floats to specific decimal places
```

```
print( "Your average sweat volume is {:.2f}".format(average_sweat_ml) )
```

```
# fixed widths
```

```
print( """
```

```
| username | sweat |
```

```
=====
```

```
|{:<10}|{:>7.5g}| """ .format(username, average_sweat_ml) )
```

```
# using parts of a string (prints "mw")
```

```
print( "Your initials are: {}".format(username[0], username[2]) )
```

```
# leading zeroes (prints "0017")
```

```
print( "you are customer number {:04d}".format( 17 ) )
```

Splitting Strings

```
my_string = "lion, tiger, giraffe"
```

```
animals = my_string.split(',')
```

```
print(animals)          # prints ['lion', 'tiger', 'giraffe']
```

```
print(animals[0])       # prints "lion"
```

SUBPROGRAMS (FUNCTIONS)

A simple subprogram

```
def display_greeting():  
    print("Hello, Welcome to the system")  
  
# Call the function  
display_greeting()
```

Parameters

```
def display_greeting():  
    print("Hello, Welcome to the system")  
  
# Call the function  
display_greeting()  
  
def display_personal_greeting(name):  
    print("Hello {}, Welcome to the system".format(name))  
  
# Call the function  
display_personal_greeting("Harambe")  
  
# or  
my_name = "Harambe"  
display_personal_greeting(my_name)
```

Returning Values

```
def calculate_vat(price_without_vat):  
    vat = 1.20 #20%  
  
    price_with_vat = price_without_vat * vat  
  
    return price_with_vat  
  
# Call the function  
price = 10.50  
final_price = calculate_vat(price)
```

WORKING WITH FILES

```
# Open a text file (in read mode)
file = open("my_text_file.txt", 'r')

# Read the contents from the
# file into a variable

contents = file.read()

# Break the contents into a
# list of lines

lines = contents.split('\n')

# Close the file when you're finished

file.close()
```

DATASTRUCTURES

Creating a List

```
# Create a new list

my_list = ['red', 'green', 'blue']

# Access the first element in the list

print( my_list[0] )      # prints "red"

# Access the 3rd element

print( my_list[2] )      # prints "blue"
```

Adding to a list

```
my_list = ['red', 'green', 'blue']

my_list.append('yellow')

# Access the new element
print( my_list[3] )      # prints "yellow"
```

Looping through the contents of a list

```
my_list = ['red', 'green', 'blue', 'yellow']
```

```
for colour in my_list:
    print(colour) # prints "red"
                  # prints "green"
                  # prints "blue"
                  # prints "yellow"
```

STANDARD ALGORITHMS

Linear Search (find first match)

```
customers = [ [1, 'Harambe'],
               [2, 'Kong'],
               [3, 'Joe'], ]
```

```
def find_customer_name(customer_id):
    for customer in customers:
        if customer[0] == customer_id:
            return customer[1]
```

```
cust = find_customer_name(2)
print(cust) # prints "Kong"
```

Linear Search (Find all matches)

```
          #    0    1          2
customers = [ [1, 'Harambe', 'yes'],
               [2, 'Kong', 'no'],
               [3, 'Joe', 'no' ] ]
```

```
def find_fictional_customers():
    results = [] # empty list
    for customer in customers:
        if customer[2] == 'no':
            results.append(customer)

    return results
```

```
fictional = find_fictional_customers()
print(fictional) # prints [[2, 'Kong', 'no'],
                  #        [3, 'Joe', 'no' ] ]
```

Bubble Sort

```
customers = [ [1, 'Harambe'],
               [2, 'Kong'],
               [3, 'Joe'], ]

def bubble_sort(the_list):
    while True:
        swapped = False
        for i in range( len( the_list )-1 ):
            # the_list[0][1] is 'Harambe'
            if the_list[i][1] > the_list[i+1][1]:
                temp = the_list[i]
                the_list[i] = the_list[i+1]
                the_list[i+1] = temp
                swapped = True
        if not swapped:
            break

bubble_sort(customers)
print(customers)          # prints [ [1, 'Harambe'],
                           #          [3, 'Joe']
                           #          [2, 'Kong'] ]
```

Find the biggest

```
customers = [ #    0    1          2
               [1, 'Harambe', 172],
               [2, 'Kong', 190],
               [3, 'Joe', 164], ]

def biggest(the_list):
    biggest = the_list[0]
    for item in the_list:
        if item[2] > biggest[2]:
            biggest = item
    return biggest

heaviest = biggest(customers)
print(heaviest)  # prints [2, 'Kong', 190]
```


PROGRAM MENU FRAMEWORK

```
def option_1():
    print("Option 1...")

def option_2():
    print("Option 2...")

def option_3():
    print("Option 3...")

while True:
    choice = input(""" Please choose an option:
1. Option 1
2. Option 2
3. Option 3
q. Quit
: """)
    if choice == '1':
        option_1()
    elif choice == '2':
        option_2()
    elif choice == '3':
        option_3()
    elif choice == 'q':
        break
    else:
        print("Choose a valid option")
```