Python Basics

**Whitespace matters, your code will not run correctly unless you use proper indentation - 4 spaces.*

```
# this is an individual comment

“”” This is how you make
a comment block with multiples lines.
“””
```

Variables and Strings: variables and strings are used to store values. A string is a series of characters, surrounded by ‘ ‘, or “ “

```
# assign variables
>>> my_string = “Hello, world!”
>>> my_int = 10
>>> my_float = 10.99
>>> my_boolean = True
```

User Input: your programs can prompt the user for input. All input is stored as a string.

```
>>> name = input(“Enter your name? ”)

# prompting for numerical input
>>> age = input(“Enter your age: ” )
>>> age = int(age)
>>> tax = input(“How much is the tax? “)
>>> tax = float(tax)
```

Lists: a list stores series of items in a particular order. You access items using an index, or within a loop.

```
# make list
>>> animals = [ “cat”, “dog”, “horse”]

# get the first item in a list -- notice the index
>>> first_animal = animals[0]
>>> first_animal
cat

# get the last item in a list
>>> last_animal = animals[-1]
>>> last_animal
horse

# add item to list
>>> animals.append(‘goat’)
>>> animals
[ “cat”, “dog”, “horse”, “goat”]

# slicing a list
>>> first_two_animals = animals[:2]
>>> first_two_animals
[“cat”, “dog”]
```

If statements: if statements are used to test for particular conditions respond appropriately.

```
# conditional tests
Equals                  x == 42
Not equal               x !=  42
Greater than            x  >  42
   or equal to          x >= 42
Less than               x  <  42
   or equal to          x <= 42
```

```
# simple if test
if age >= 21:
   print(“You can drink!”)

# conditional test with a list
>>> ‘dog’ in animals
TRUE

# if-else
if pass == True:
   print(“You have access”)
else:
```

```
    print("You have been denied
access")

# if-elif-else
if color == "Green":
    print("GO")
elif color == "Red"
    print("STOP")
else:
    print("SLOW")
```

Dictionaries: dictionaries store connections between pieces of information. Each item in a dictionary is a key-value pair.

```
# assign a simple dictionary
>>> my_dict = { 'Species': 'cat', 'Age':
7, 'Name': 'Marilyn'}

# access a value
>>> my_dict['Species']
cat

# adding a new key-value pairs
>>> my_dict['Age'] = 8

# looping through all key-value pairs
for key, value in my_dict.items():
    print(str(key) + " : " + str(value))

# looping through all the values
for value in my_dict.values():
    print(str(value))
```

While loops: a while loop repeats a block of code as long as a certain condition is true.

```
# a simple while loop
current_value = 1
while current_value <= 5:
    print(current_value)
    current_value += 1

# let user decide when to quit
activate = True
while activate != False:
    activate = input("Continue? ")
    print(activate)
```

Functions: functions are named for blocks of code, designed to do a specific job. The information passed to a function is called an argument, and information received by a function is call a parameter.

```
# a simple function
def write_message():
    msg = input("Write your message: ")
    print(msg)

# call a function
>>>write_message()

# passing an argument
def welcome_user(username):
    print("Hello " + username)

>>> welcome_user("Ashley")

# default values for parameters
def make_pizza(topping='mushroom'):
    print(" Ordering a " + topping + "
pizza!")

>>>make_pizza()
Ordering a mushroom pizza!
>>>make_pizza('pepperoni')
Ordering a pepperoni pizza!

# returning a value
def add_tip(bill,tip):
    return bill + tip

>>> bill_sum = add_tip(20.00, 10)
>>> print(bill_sum)
30.00
```
