<https://buckysroom.org/videos.php>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Python Programming Tutorials**

<http://www.youtube.com/course?list=ECEA1FEF17E1E5C0DA>

<http://thenewboston.org/list.php?cat=36>

|  |  |
| --- | --- |
| [1 - Installing Python](http://thenewboston.org/watch.php?cat=36&number=1) |  |
| [2 - Numbers and Math](http://thenewboston.org/watch.php?cat=36&number=2) |  |
| [3 - Variables](http://thenewboston.org/watch.php?cat=36&number=3) |  |
| [4 - Modules and Functions](http://thenewboston.org/watch.php?cat=36&number=4) |  |
| [5 - How to Save Your Programs](http://thenewboston.org/watch.php?cat=36&number=5) |  |
| [6 - Strings](http://thenewboston.org/watch.php?cat=36&number=6) |  |
| [7 - More on Strings](http://thenewboston.org/watch.php?cat=36&number=7) |  |
| [8 - Raw Input](http://thenewboston.org/watch.php?cat=36&number=8) |  |
| [9 - Sequences and Lists](http://thenewboston.org/watch.php?cat=36&number=9) |  |
| [10 - Slicing](http://thenewboston.org/watch.php?cat=36&number=10) |  |
| [11 - Editing Sequences](http://thenewboston.org/watch.php?cat=36&number=11) |  |
| [12 - More List Functions](http://thenewboston.org/watch.php?cat=36&number=12) |  |
| [13 - Slicing Lists](http://thenewboston.org/watch.php?cat=36&number=13) |  |
| [14 - Intro to Methods](http://thenewboston.org/watch.php?cat=36&number=14) |  |
| [15 - More Methods](http://thenewboston.org/watch.php?cat=36&number=15) |  |
| [16 - Sort and Tuples](http://thenewboston.org/watch.php?cat=36&number=16) |  |
| [17 - Strings n Stuff](http://thenewboston.org/watch.php?cat=36&number=17) |  |
| [18 - Cool String Methods](http://thenewboston.org/watch.php?cat=36&number=18) |  |
| [19 - Dictionary](http://thenewboston.org/watch.php?cat=36&number=19) |  |
| [20 - If Statement](http://thenewboston.org/watch.php?cat=36&number=20) |  |
| [21 - else and elif](http://thenewboston.org/watch.php?cat=36&number=21) |  |
| [22 - Nesting Statements](http://thenewboston.org/watch.php?cat=36&number=22) |  |
| [23 - Comparison Operators](http://thenewboston.org/watch.php?cat=36&number=23) |  |
| [24 - And and Or](http://thenewboston.org/watch.php?cat=36&number=24) |  |
| [25 - For and While Loops](http://thenewboston.org/watch.php?cat=36&number=25) |  |
| [26 - Infinite Loops and Break](http://thenewboston.org/watch.php?cat=36&number=26) |  |
| [27 - Building Functions](http://thenewboston.org/watch.php?cat=36&number=27) |  |
| [28 - Default Parameters](http://thenewboston.org/watch.php?cat=36&number=28) |  |
| [29 - Multiple Parameters](http://thenewboston.org/watch.php?cat=36&number=29) |  |
| [30 - Parameter Types](http://thenewboston.org/watch.php?cat=36&number=30) |  |
| [31 - Tuples as Parameters](http://thenewboston.org/watch.php?cat=36&number=31) |  |
| [32 - Object Oriented Program](http://thenewboston.org/watch.php?cat=36&number=32) |  |
| [33 - Classes and Self](http://thenewboston.org/watch.php?cat=36&number=33) |  |
| [34 - Subclasses Superclasses](http://thenewboston.org/watch.php?cat=36&number=34) |  |
| [35 - Overwrite Variable on Sub](http://thenewboston.org/watch.php?cat=36&number=35) |  |
| [36 - Multiple Parent Classes](http://thenewboston.org/watch.php?cat=36&number=36) |  |
| [37 - Constructors](http://thenewboston.org/watch.php?cat=36&number=37) |  |
| [38 - Import Modules](http://thenewboston.org/watch.php?cat=36&number=38) |  |
| [39 - reload Modules](http://thenewboston.org/watch.php?cat=36&number=39) |  |
| [40 - Getting Module Info](http://thenewboston.org/watch.php?cat=36&number=40) |  |
| [41 - Working with Files](http://thenewboston.org/watch.php?cat=36&number=41) |  |
| [42 - Reading and Writing](http://thenewboston.org/watch.php?cat=36&number=42) |  |
| [43 - Writing Lines](http://thenewboston.org/watch.php?cat=36&number=43) |  |

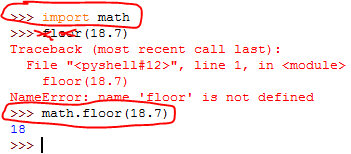
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## Modules and Functions (4/43)

We can import **modules** with additional functions

import math

to use for example a "floor" function print: math.floor(18.7)

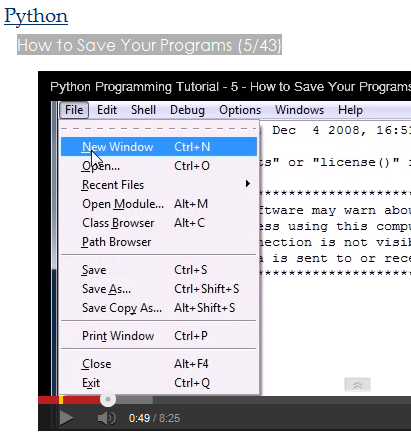


We can assign a function to a variable



--============================================================================

## How to Save Your Programs (5/43)



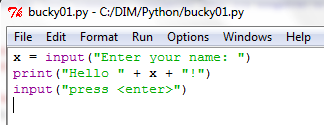
Save with the extension **.py**

**Run the program using F5 from the shell.**

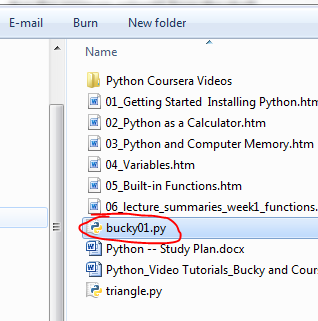
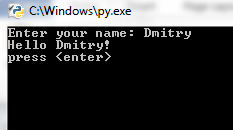
x = input("Enter your name: ")

print("Hello " + x + "!")

input("press <enter>")



**or double click it in the Windows explorer**

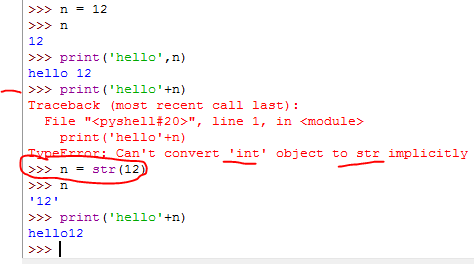


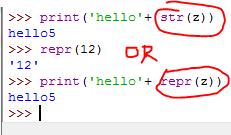
--============================================================================

## Strings (6/43)

--============================================================================

## More on Strings (7/43)





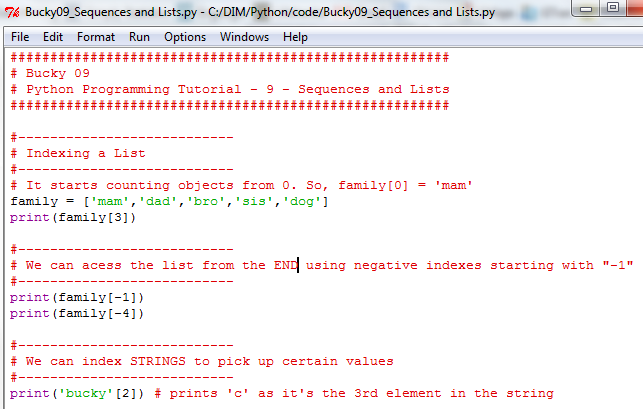
--============================================================================

## Python Programming Tutorial - 8 - Raw Input

--============================================================================

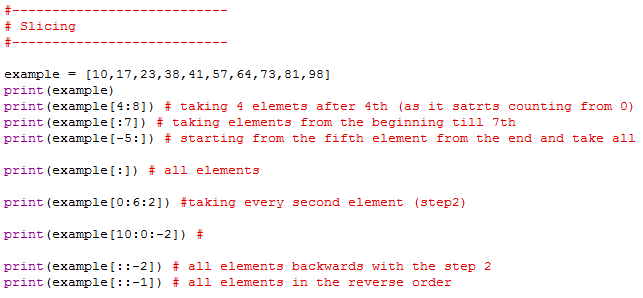
## Python Programming Tutorial - 9 - Sequences and Lists

Something that lets you store data



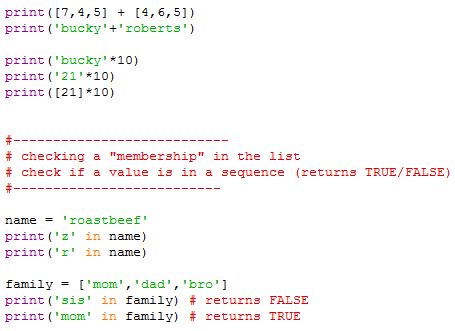
--============================================================================

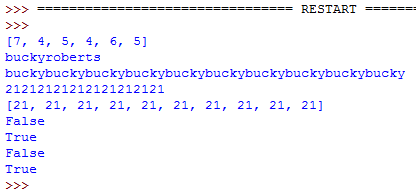
## Python Programming Tutorial - 10 - Slicing



--============================================================================

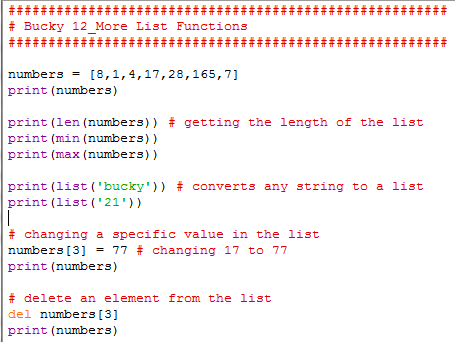
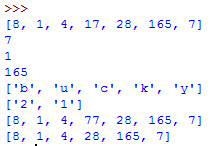
## Python Programming Tutorial - 11 - Editing Sequences





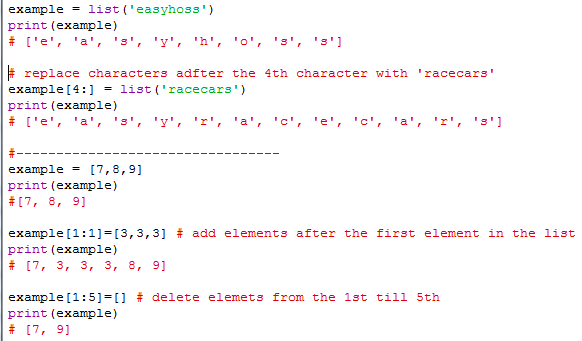
--============================================================================

## Python Programming Tutorial - 12 - More List Functions



--============================================================================

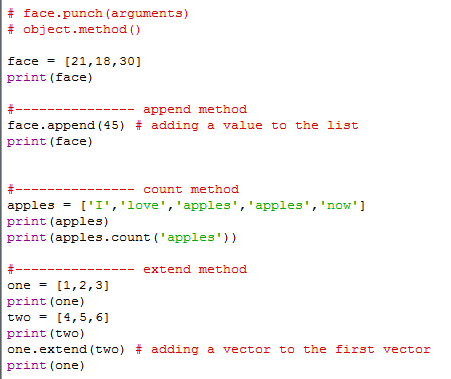
## Python Programming Tutorial - 13 - Slicing Lists



--============================================================================

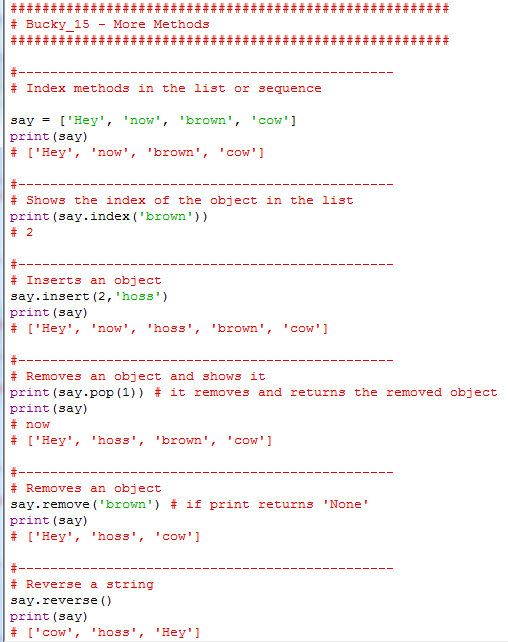
## Python Programming Tutorial - 14 - Intro to Methods

Something you do with the object



--============================================================================

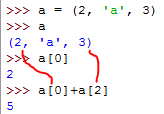
## Python Programming Tutorial - 15 - More Methods

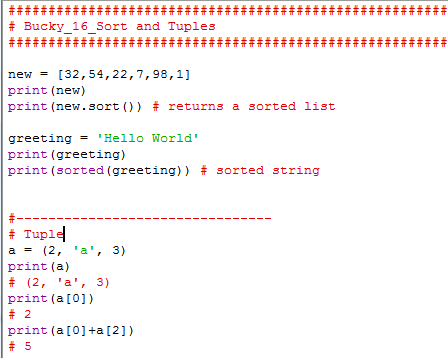


--============================================================================

## Python Programming Tutorial - 16 - Sort and Tuples

Tuples a list that can handle different value types: e.g. strings and numbers

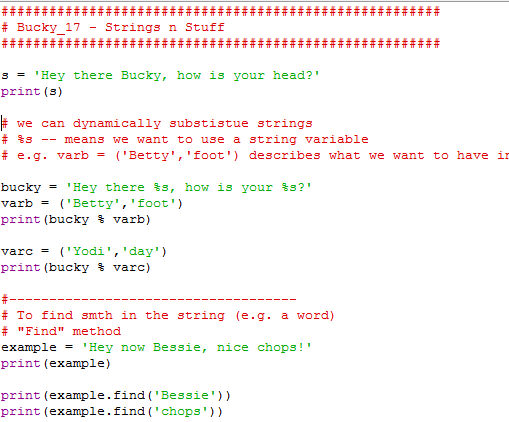


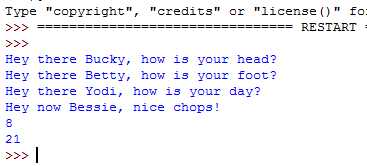


--============================================================================

## Python Programming Tutorial - 17 - Strings n Stuff

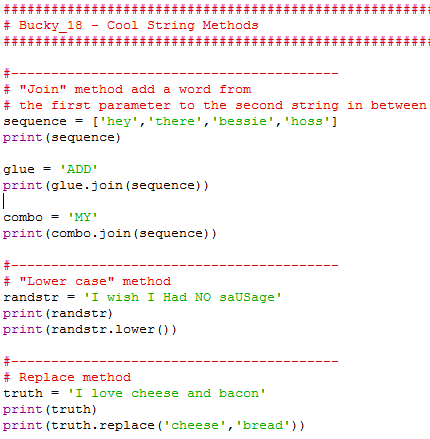
How to forma a string

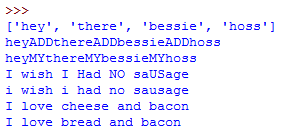




--============================================================================

## Python Programming Tutorial - 18 - Cool String Methods





--============================================================================

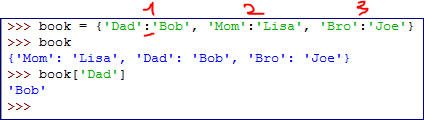
# Python Programming Tutorial - 19 - Dictionary

A dictionary has "Keys" and "Values".

For example 'Dad' is a "key" in this dictionary.

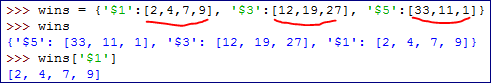
In order to get a value associated with this "key" we use the syntax:

dictionaryName['key']



We can have more elements associated with one key.

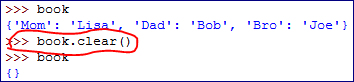
Those values must be in the form of a list.



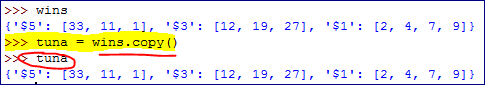
**Methods:**

1. Clear a dictionary

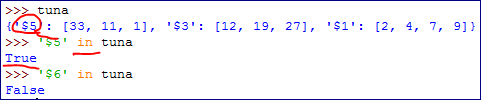
<dictionary name>.clear() -- removes everything from the dictionay



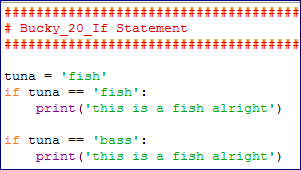
2. Copy a dictionary to another dictionary



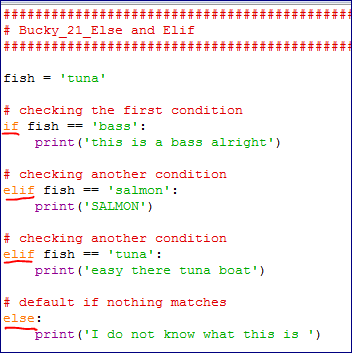
3. Check if a dictionary has a "key". If it does have the "key" it returns "TRUE" otherwise it returns "FALSE"



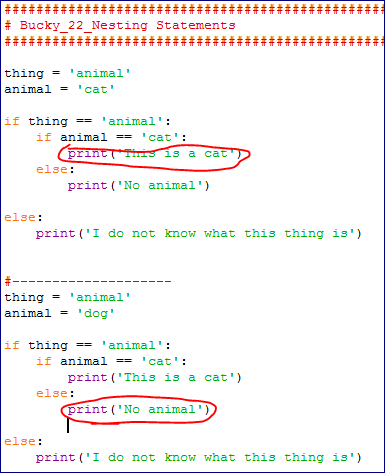
# Python Programming Tutorial - 20 - If Statement



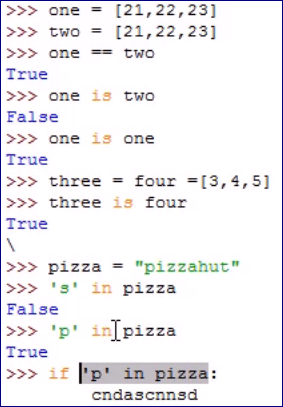
# Python Programming Tutorial - 21 - else and elif



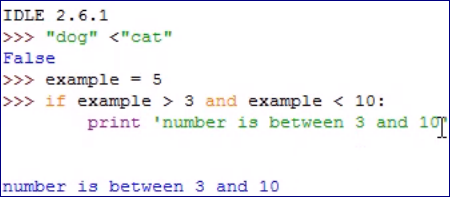
# Python Programming Tutorial - 22 - Nesting Statements

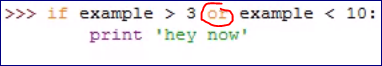


# Python Programming Tutorial - 23 - Comparison Operators

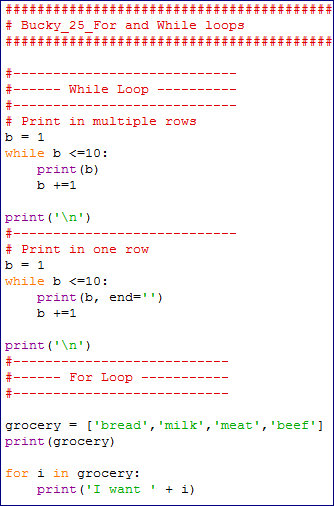
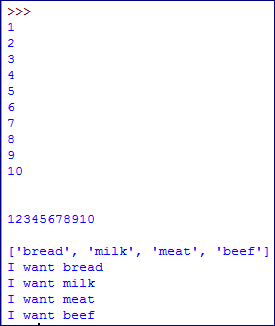


# Python Programming Tutorial - 24 - And and Or

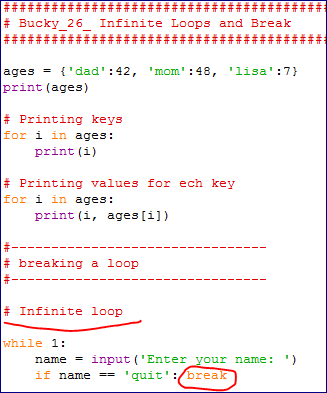
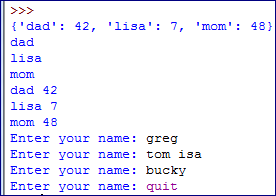




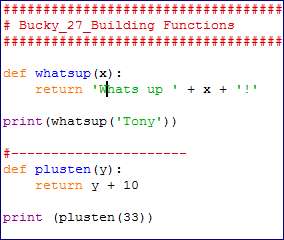
# Python Programming Tutorial - 25 - For and While Loops



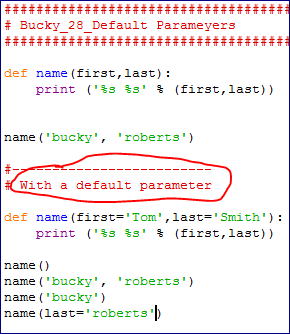
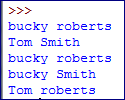
# Python Programming Tutorial - 26 - Infinite Loops and Break



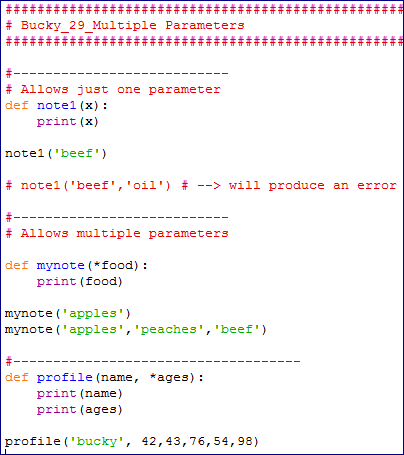
# Python Programming Tutorial - 27 - Building Functions

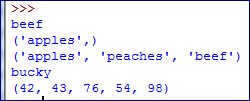


# Python Programming Tutorial - 28 - Default Parameters

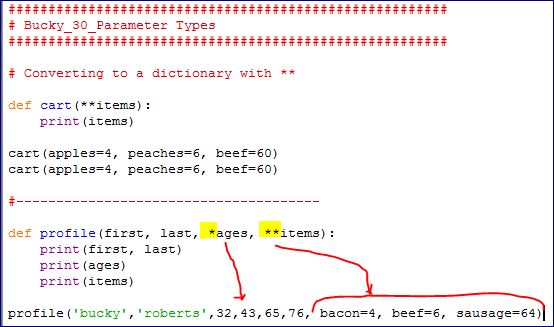


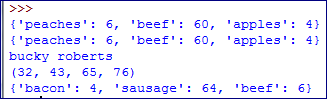
# Python Programming Tutorial - 29 - Multiple Parameters



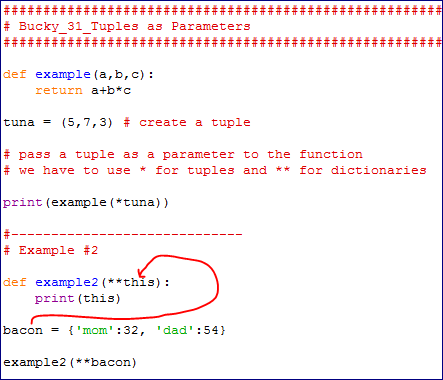


# Python Programming Tutorial - 30 - Parameter Types

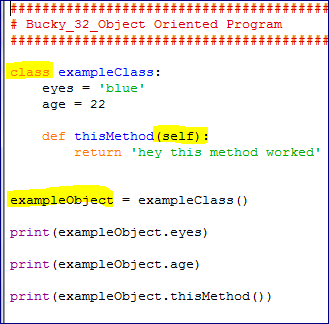
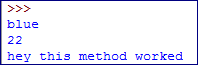




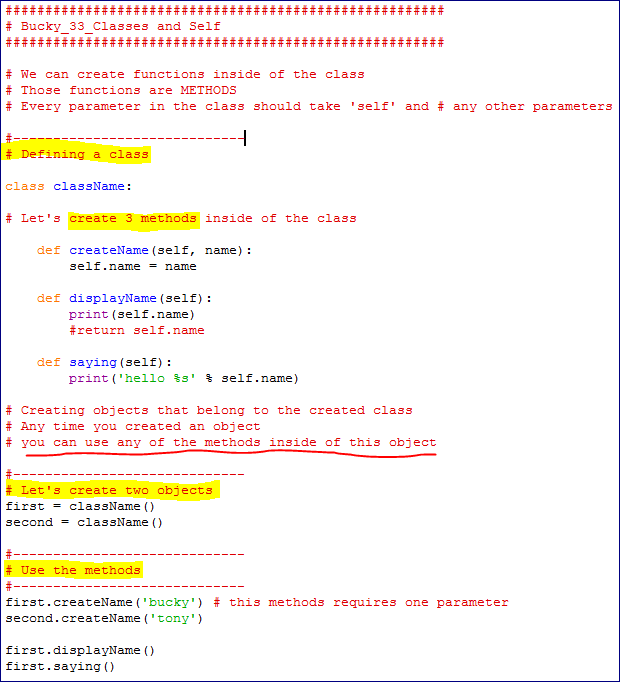
# Python Programming Tutorial - 31 - Tuples as Parameters



# Python Programming Tutorial - 32 - Object Oriented Program

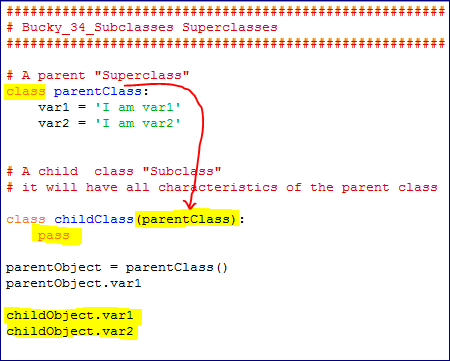


# Python Programming Tutorial - 33 - Classes and Self

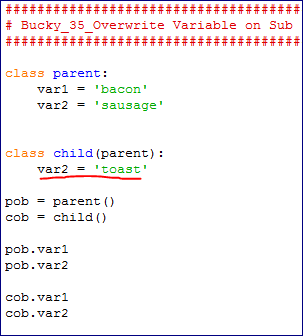
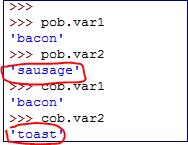




# Python Programming Tutorial - 34 - Subclasses Superclasses

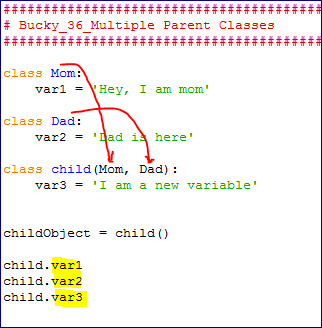
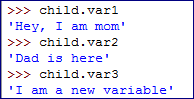


# Python Programming Tutorial - 35 - Overwrite Variable on Sub

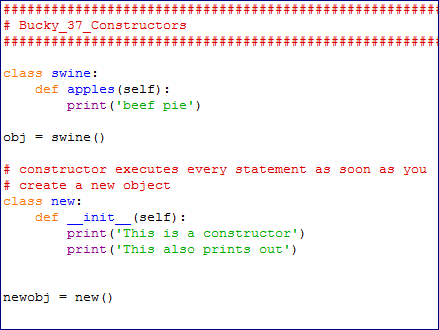
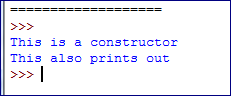


# Python Programming Tutorial - 36 - Multiple Parent Classes

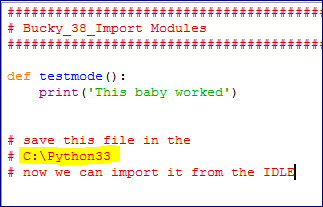
We can access all variables in the super classes



# Python Programming Tutorial - 37 - Constructors



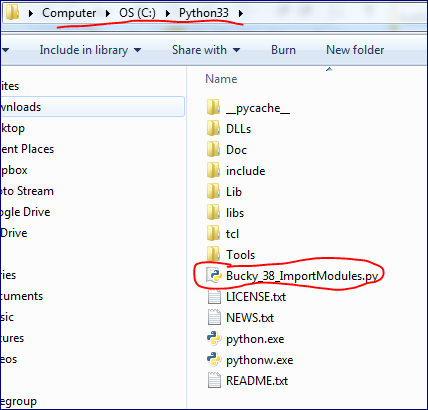
# Python Programming Tutorial - 38 - Import Modules





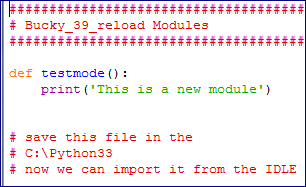
If you made changes in the module you HAVE to reload the IDLE.

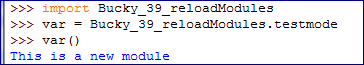
Just resaving the file on the hard drive and importing it will not change the module.



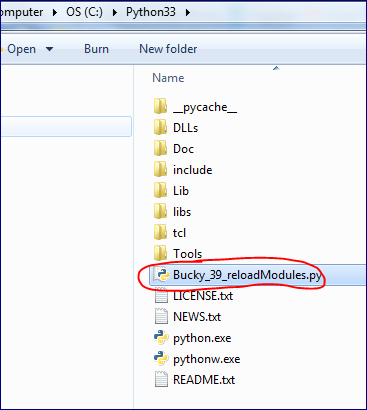
# Python Programming Tutorial - 39 - reload Modules

How to reload modules once your source file has been edited





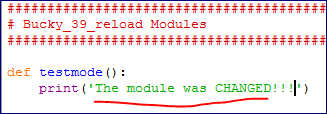
This is where the module is saved.

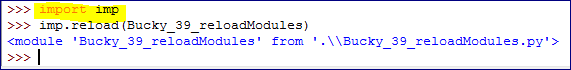


We want to change the module.

Save it

And reload without exiting and opening the IDLE again



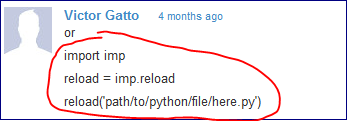




import﻿ imp

reload = imp.reload

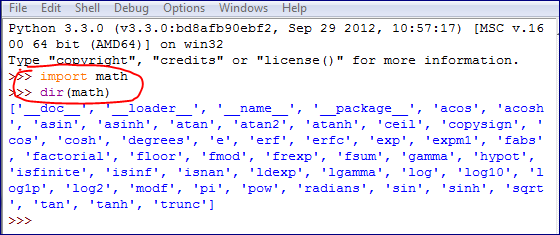
reload('path/to/python/file/he­re.py')

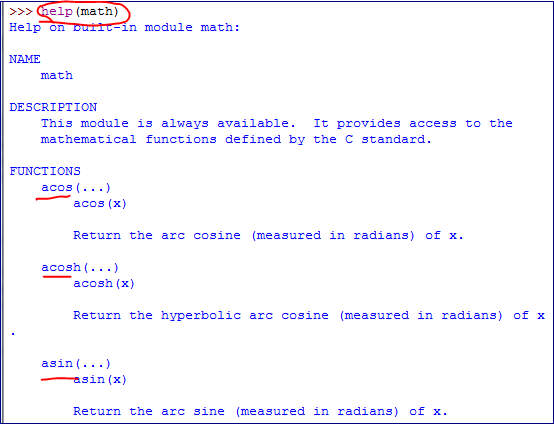


# Python Programming Tutorial - 40 - Getting Module Info

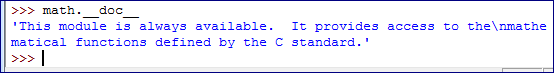
How to get help with built in modules.

**dir()** tells you what the model contains





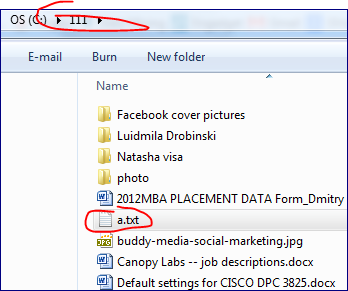
A quick summary about the module

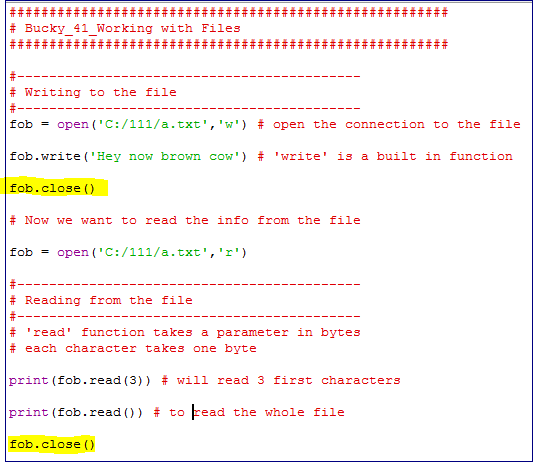


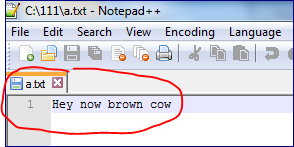
# Python Programming Tutorial - 41 - Working with Files

We need to create a file object in order to work with a file.

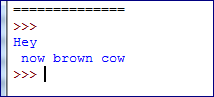
Let's create an empty .txt file



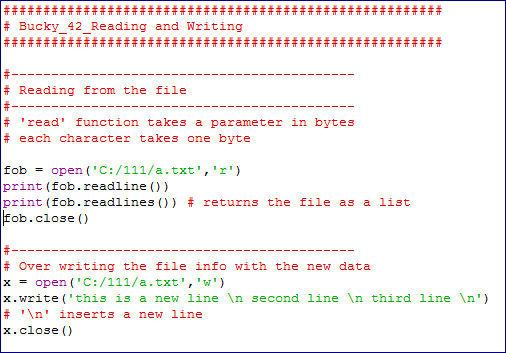


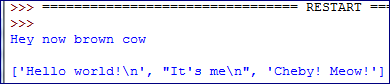


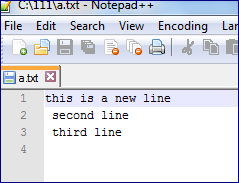
Reading from the file



# Python Programming Tutorial - 42 - Reading and Writing







# Python Programming Tutorial - 43 - Writing Lines

