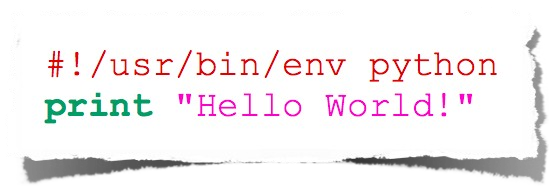
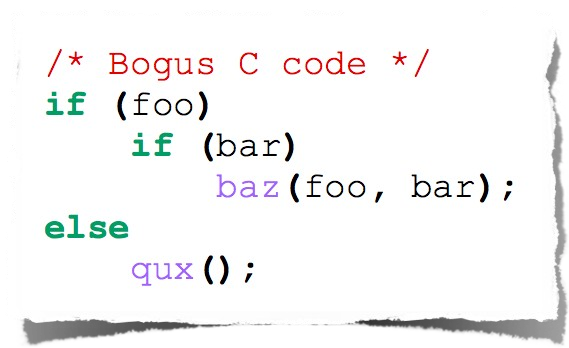
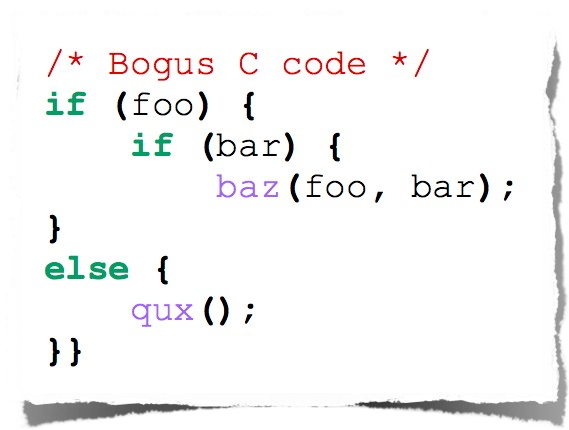
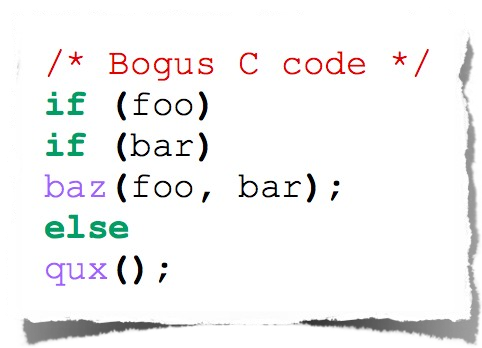
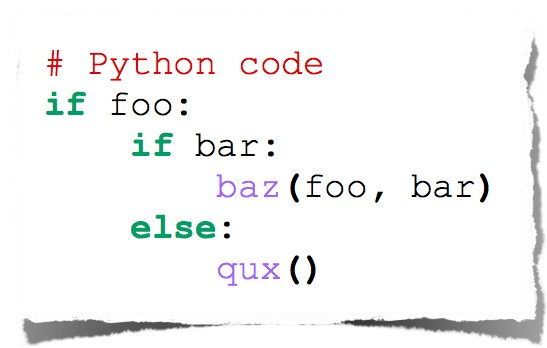
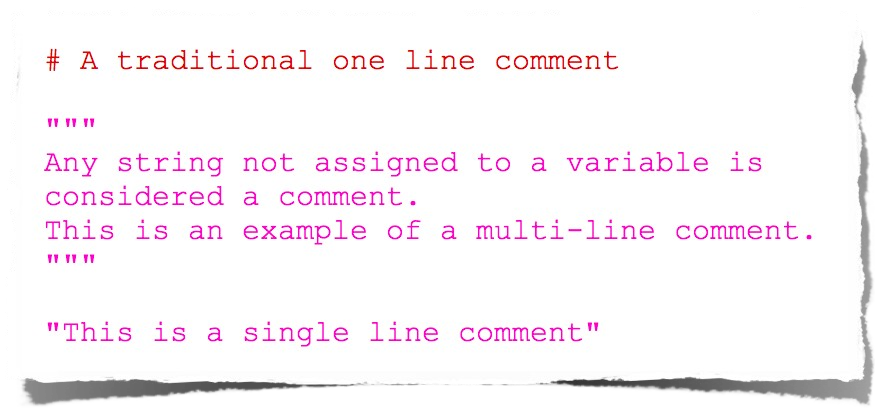
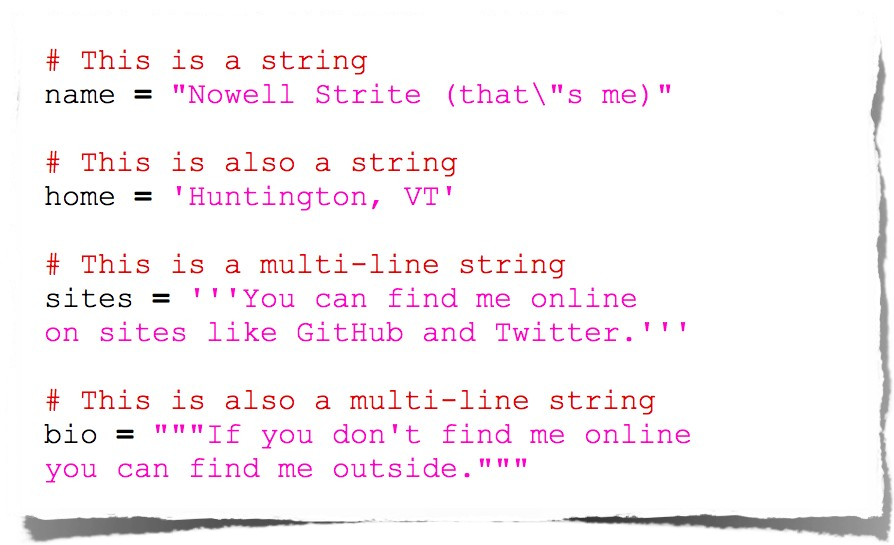
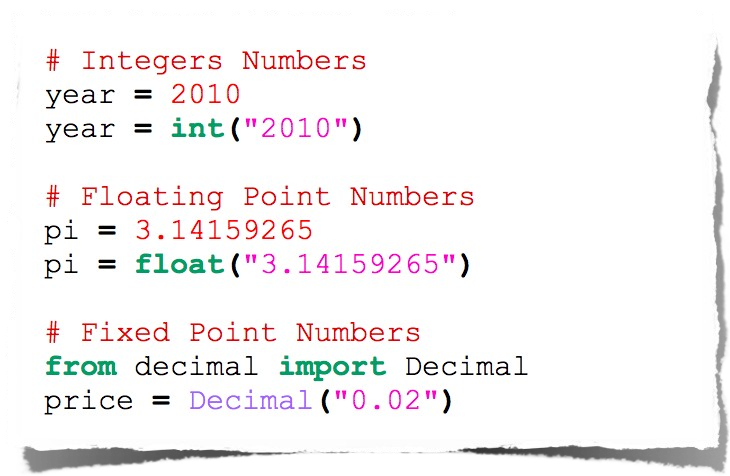
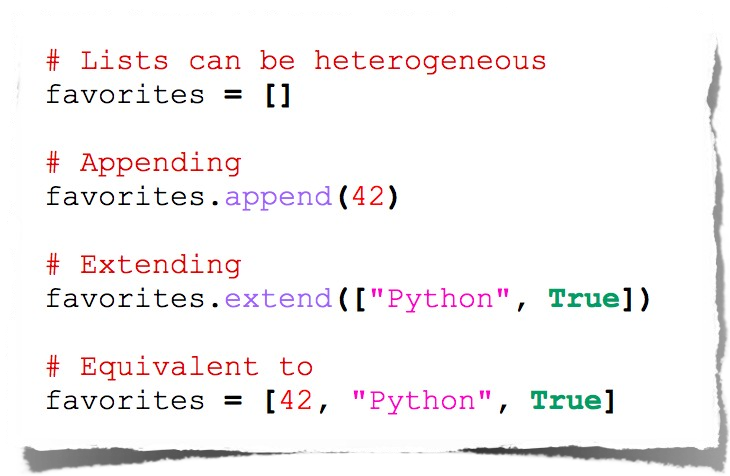
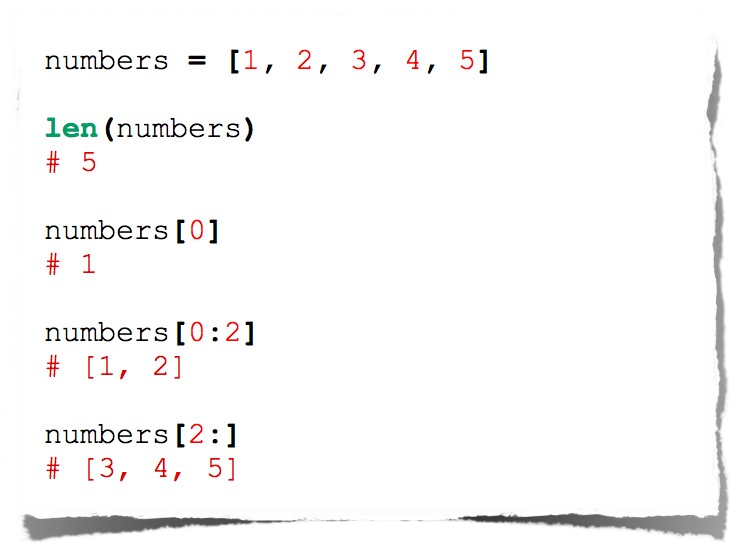
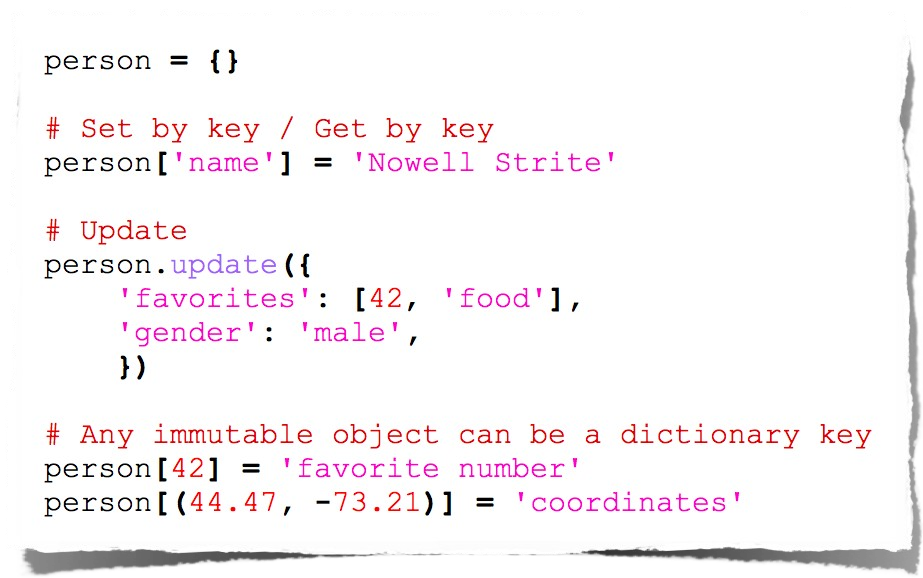
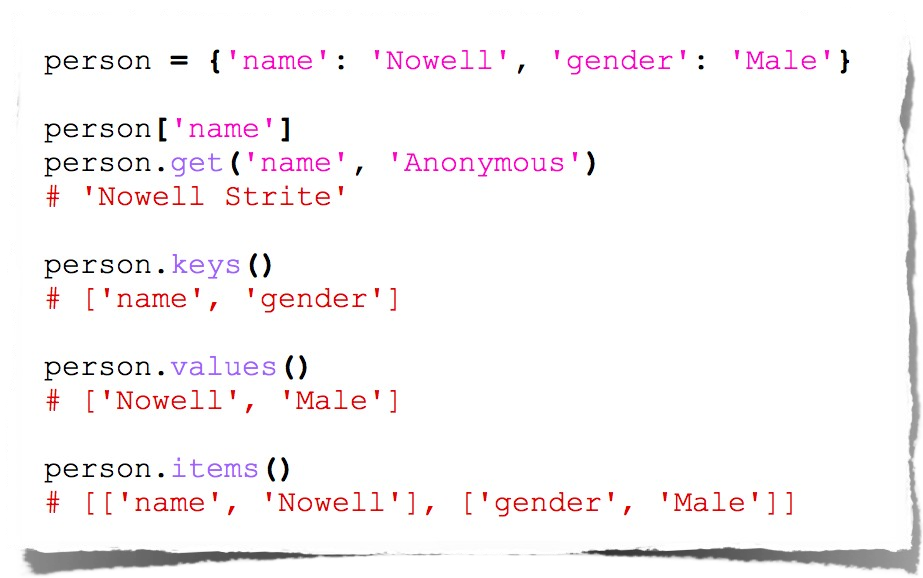
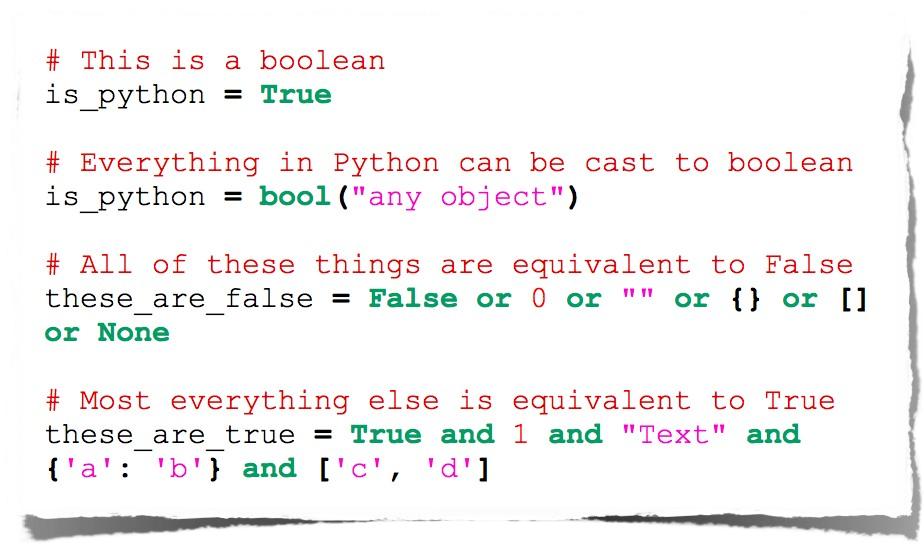
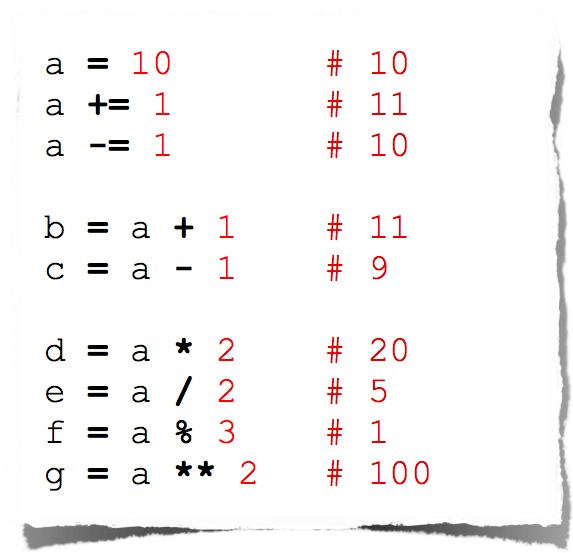
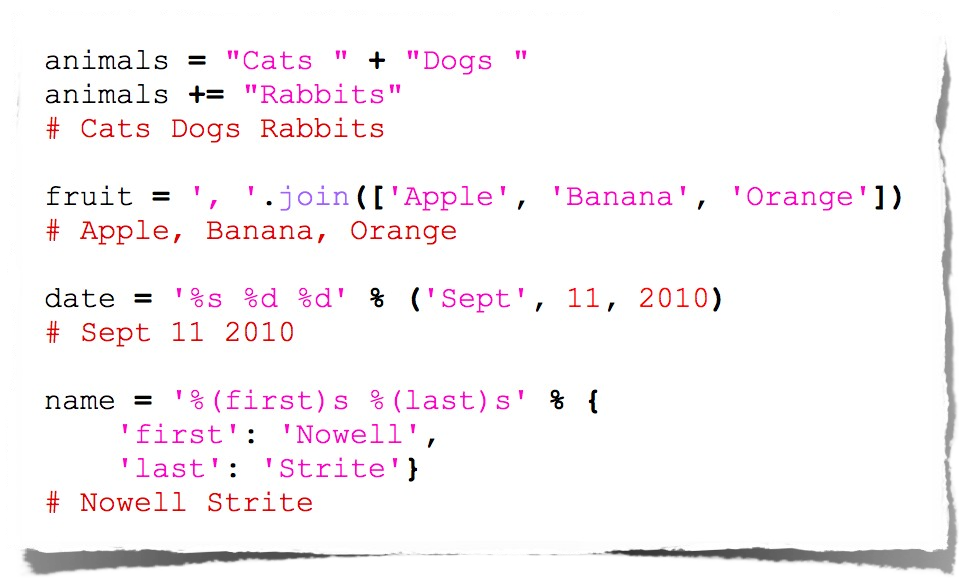
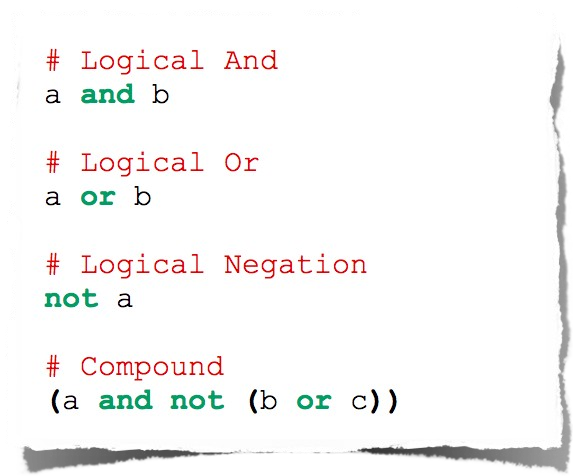
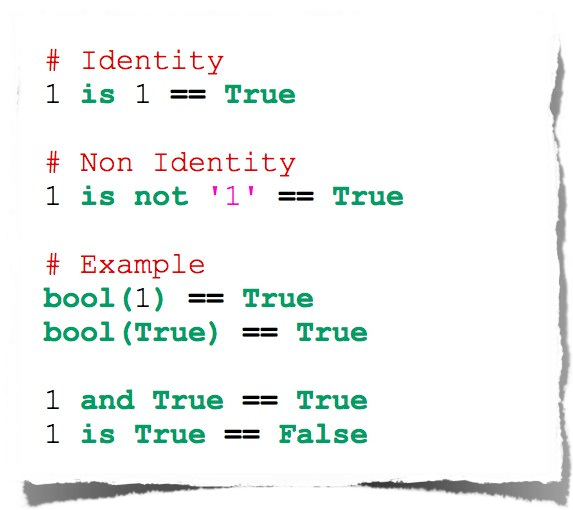
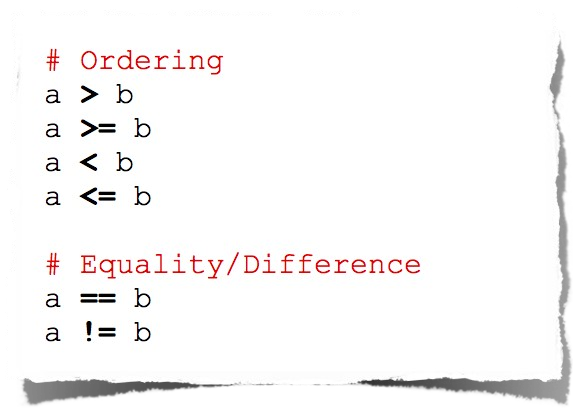
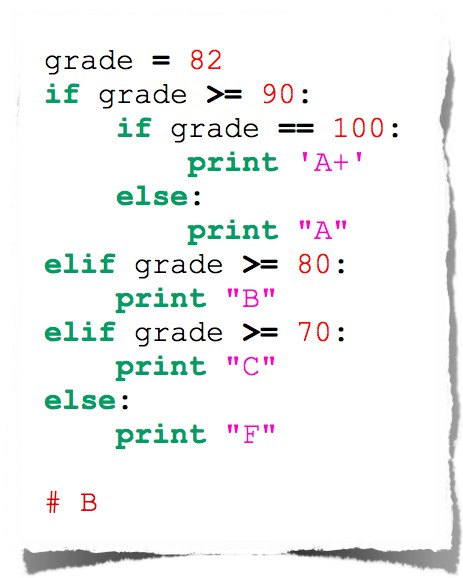
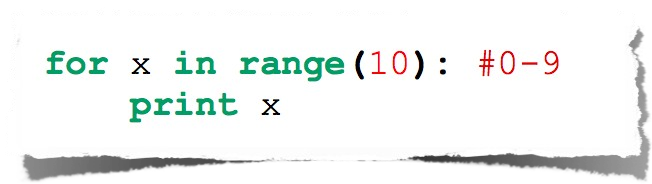
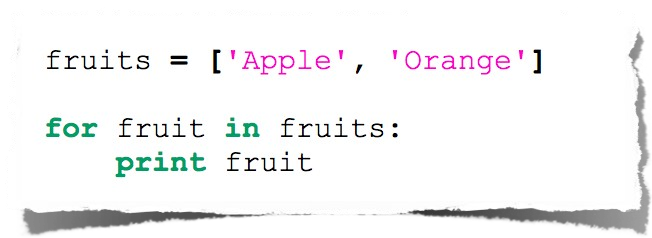
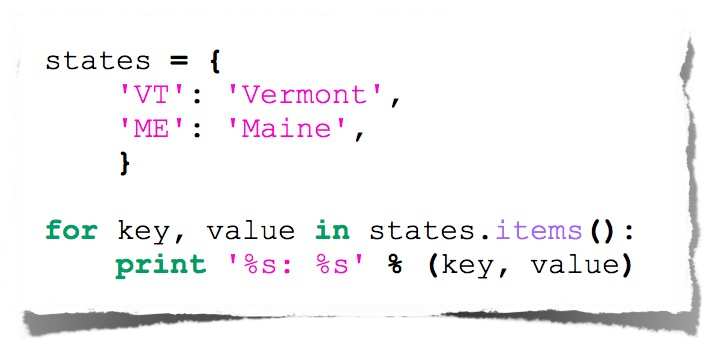
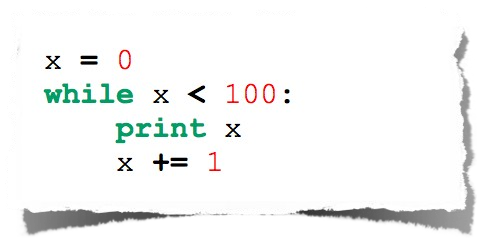
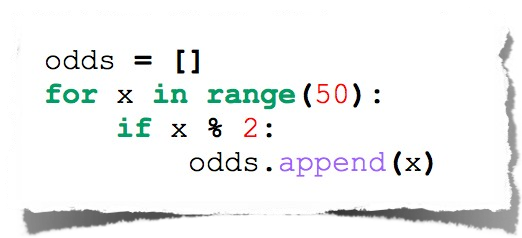
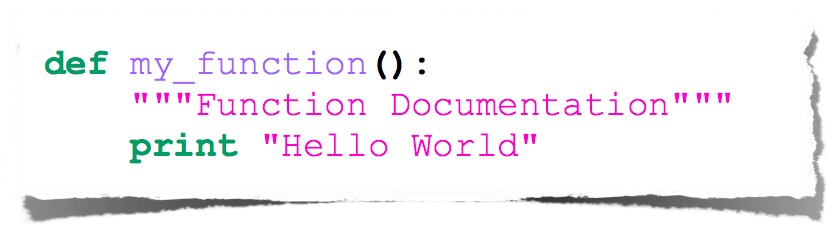
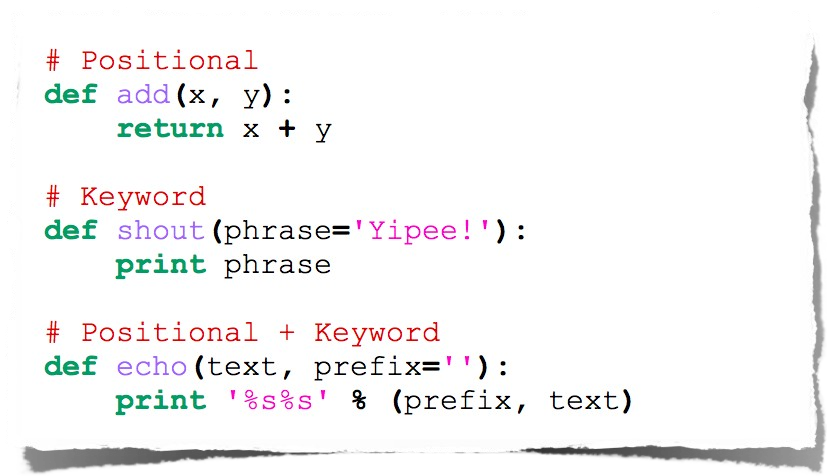
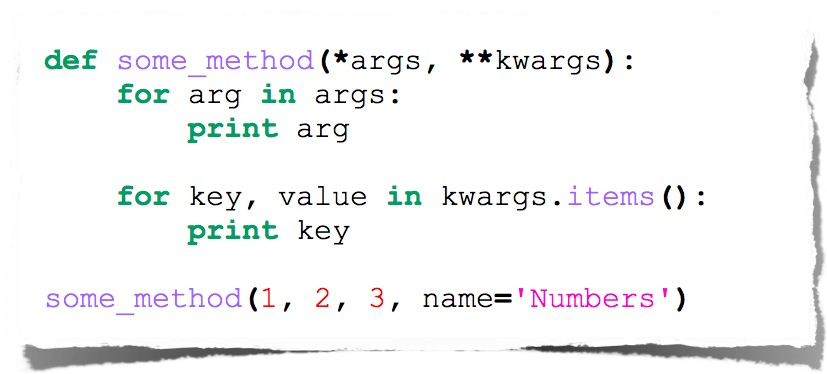
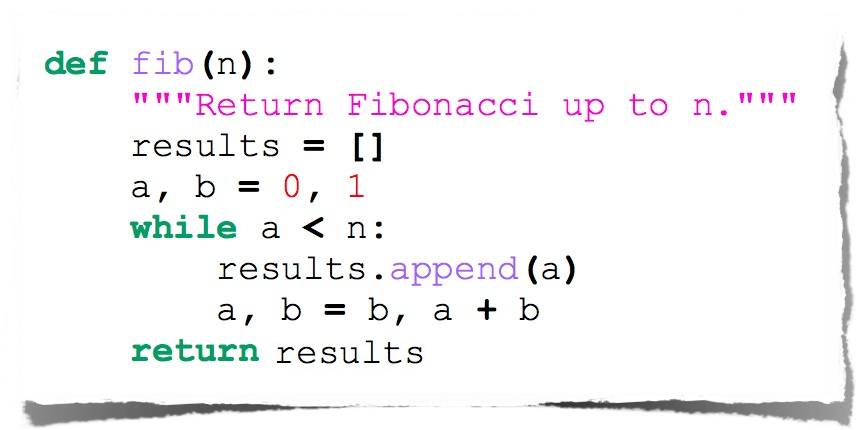
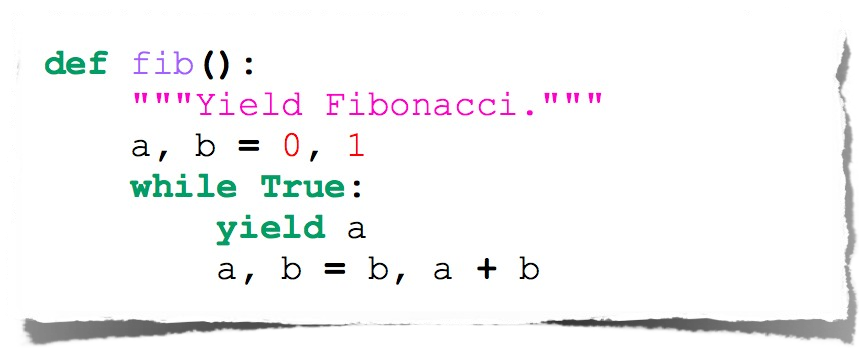
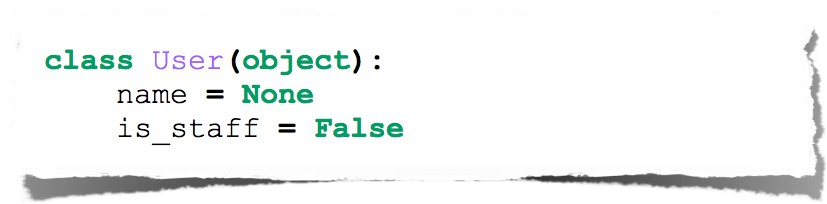
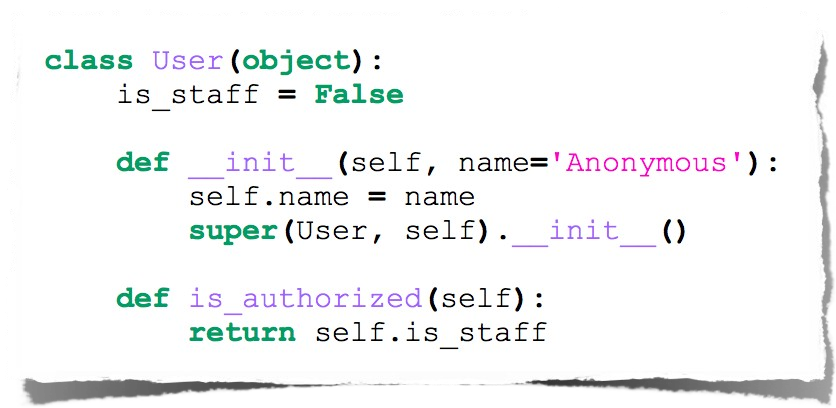
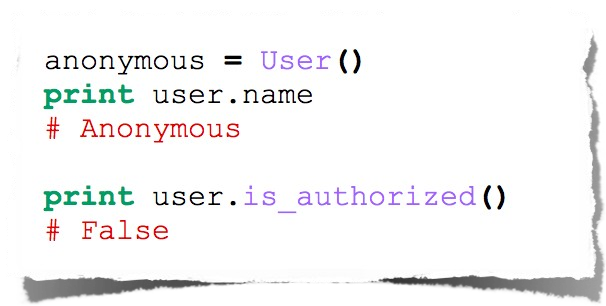
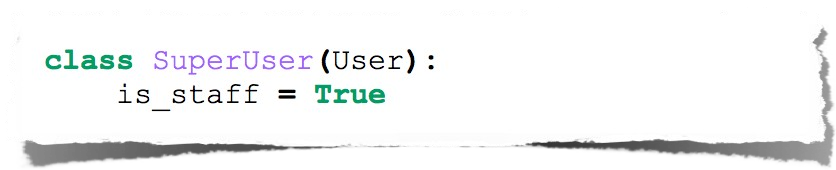
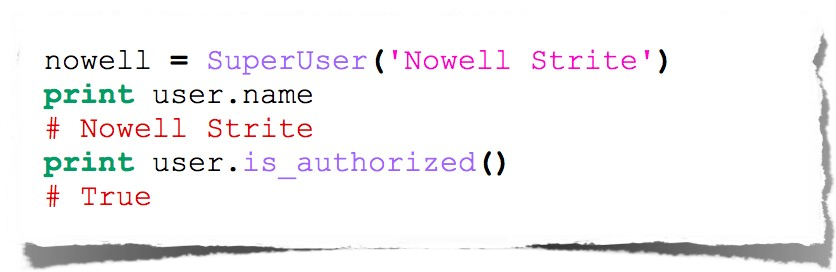
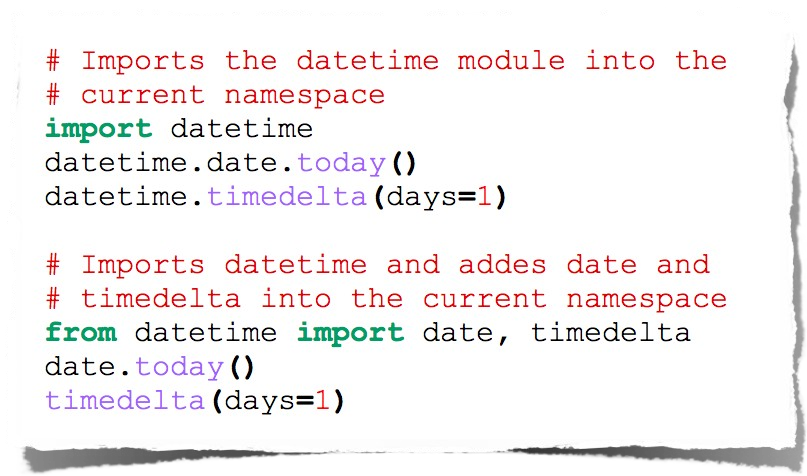
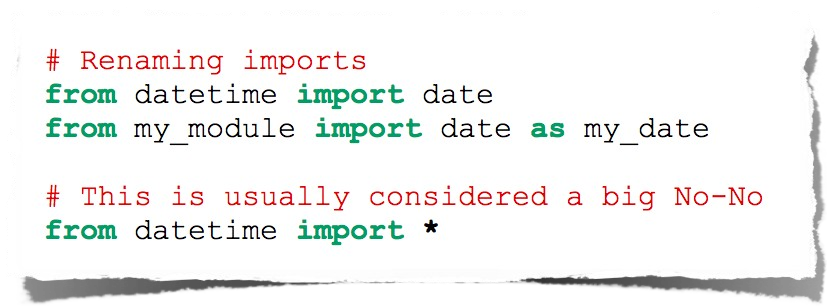
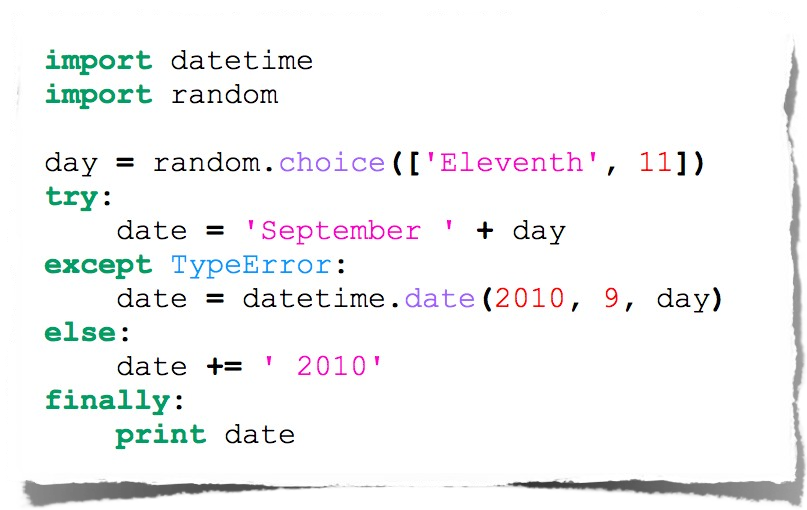
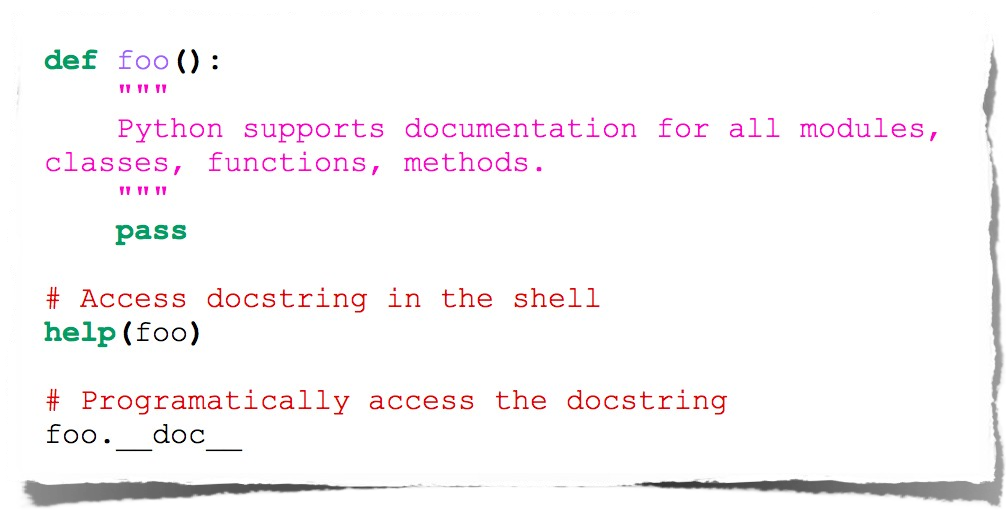
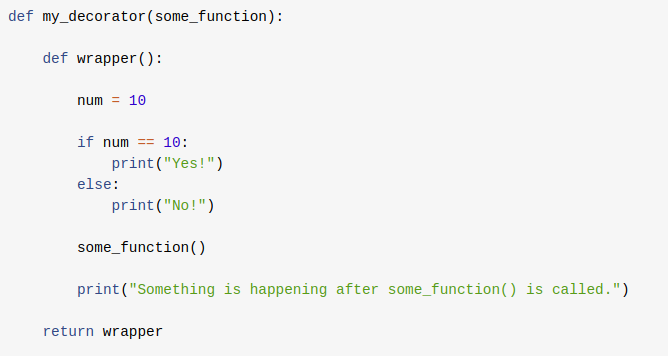
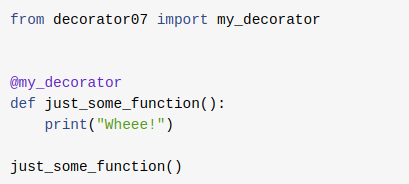
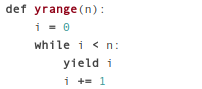
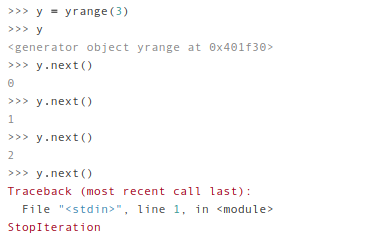
**Python**

1. What is Python
   1. Multi-purpose(Web, GUI, Scripting)
   2. Object oriented
   3. Interpreted
   4. Strongly typed and Dynamically typed
   5. Focus on readability and productivity
   6. Features
      1. Batteries included
      2. Everything is an Object
      3. Interactive Shell
      4. Strong Introspection
      5. Cross Platform
      6. CPython, Jython, IronPython, PyPy
   7. Who uses python
      1. Google.
      2. NASA
      3. Instagram
      4. Shopify
   8. Releases
      1. Created in 1989 by Guido Van Rossum
      2. Python 1.0 released in 1994
      3. Python 2.0 released in 2000
      4. Python 3.0 released in 2008
      5. Current version 3.6.4
2. Syntax
   1. 
   2. Indentation
      1. 
         1. The else here actually belongs to the 2nd if statement
      2. 
         1. The else here actually belongs to the 2nd if statement
      3. 
      4. 
         1. Python code
3. Comments
   1. 
4. Data Types
   1. Strings
      1. 
   2. Numbers
      1. 
   3. Null
      1. 
   4. Lists
      1. 
      2. 
   5. Dictionaries
      1. 
   6. Dictionary methods
      1. 
   7. Booleans
      1. 
5. Operators
   1. Arithmetic
      1. 
   2. String manipulations
      1. 
   3. Logical comparisons
      1. 
   4. Identity comparisons
      1. True
      2. Arithmetic comparisons
         1. 
6. Control flow
   1. Conditionals
      1. 
   2. Loops - For
      1. 
      2. Expanded for loop
         1. 
      3. While loop
         1. 
7. List comprehension
   * 1. 
     2. 
8. Functions
   1. Basic function
      1. 
   2. Function arguments
      1. 
   3. Arbitrary arguments
      1. 
   4. Fibonacci
      1. 
      2. 
9. Classes
   1. Class declaration
      1. 
   2. Class attributes
      1. 
   3. Class methods
      1. 
   4. Class instantiation and attribute access
      1. 
   5. Class inheritance
      1. 
      2. 
   6. Pythons way
      1. No real private attributes/functions
      2. Private attributes start (but do not end) with double underscores.
      3. Special class methods start and end with double underscores.
      4. \_\_init\_\_, \_\_doc\_\_, \_\_cmp\_\_, \_\_str\_\_
10. Imports
    1. Allows code isolation and re-use
    2. Adds references to variables/classes/functions/etc. into current namespace
    3. 
11. Error handling
    1. 
12. Documentation
    1. Docstrings
       1. 
13. Tools
    1. Web frameworks
       1. Django
       2. Flask
       3. Pylons
       4. turboGears
       5. Zope
       6. Grok
    2. IDE’s
       1. Emacs
       2. Vim
       3. Komodo
       4. PyCharm
       5. Eclipse
    3. Package management
    4. Easy\_install pip
    5. Pip install django
14. Decorators
    1. 
    2. 
15. Lambda functions
    1. Anonymous function is a function that is defined without a name, in Python anonymous functions are defined using the lambda keyword.
    2. lambda arguments: expression
    3. double = lambda x: x \* 2
16. Generators
    1. Python generators are a simple way of creating iterators.
    2. 
    3. 

**Links**:

<https://www.tutorialspoint.com/python/index.htm>

**Assignments:**

**Set 1:**

1. With a given integral number n, write a program to generate a dictionary that contains (i, i\*i) such that is an integral number between 1 and n (both included). and then the program should print the dictionary.

Suppose the following input is supplied to the program:

8

Then, the output should be:

{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}

1. Write a program which accepts a sequence of comma-separated numbers from console and generate a list and a tuple which contains every number.

Suppose the following input is supplied to the program:

34,67,55,33,12,98

Then, the output should be:

['34', '67', '55', '33', '12', '98']

('34', '67', '55', '33', '12', '98')

1. Define a class which has at least two methods:

getString: to get a string from console input

printString: to print the string in upper case.

~~Also please include simple test function to test the class methods.~~

1. Write a program that accepts a comma separated sequence of words as input and prints the words in a comma-separated sequence after sorting them alphabetically.

Suppose the following input is supplied to the program:

without,hello,bag,world

Then, the output should be:

bag,hello,without,worl

1. Write a program to generate and print another tuple whose values are even numbers in the given tuple (1,2,3,4,5,6,7,8,9,10).
2. Define a class named Circle which can be constructed by a radius. The Circle class has a method which can compute the area.
3. Assuming that we have some email addresses in the "username@companyname.com" format, please write program to print the user name of a given email address. Both user names and company names are composed of letters only.

If the following email address is given as input to the program:

john@google.com

Then, the output of the program should be:

john

1. Get a list of numbers from users and print the smallest odd number.
2. Read and print number of lines, words and characters in the given file.

**Set 2:**

1. Let 'a' be the list of users who likes a post! I want to get displayed as below.
   1. eg 1 :- a = []
   2. Output : Nobody likes This
   3. eg 2 :- a = ['Alice']
   4. Output : Alice likes This
   5. eg 3 :- a = ['Alice','Bob']
   6. Output : Alice and Bob likes This
   7. eg 4 :- a = ['Alice', 'Bob', 'Charls']
   8. Output : Alice, Bob and Charls likes This
   9. eg 5 :- a = ['Alice', 'Bob', 'Charls','Denny']
   10. Output : Alice, Bob and 2 others likes This
   11. eg 6 :- a = ['Alice', 'Bob', 'Charls','Denny','Emely']
   12. Output : Alice, Bob and 3 others likes This
2. Learn below packages:
   1. os
   2. sys
   3. re
   4. math
   5. random
   6. urllib2
   7. smtplib
   8. datetime
   9. zlib
   10. unittest
   11. requests
   12. scrapy
   13. BeautifulSoup
   14. nltk
   15. nose

**Set 3**

1. Write a Python program to print yesterday, today, tomorrow.
2. Get email input from user and check it is valid or not(Output: valid/ Invalid)
3. Write a program to find all mobile number inside a string.
4. Fetch data from the URL https://www.w3schools.com/xml/simple.xml and print name, price of each item.
5. Create a class with functions to add, subtract, multiply and division. Also write unit test for each functions.
6. Write a function to send email to given email address.

**Set 4**

1. Fetch image url, country name from the URL <http://example.webscraping.com/> using beautifulsoup

**Set 5:**

1. Learn pep8 - <https://www.python.org/dev/peps/pep-0008/>
2. Create a Python SDK for few API’s mentioned here <https://jsonplaceholder.typicode.com/>
   1. Features:
      1. Post
         1. Get posts
         2. Get posts based on the user
         3. Get post detail
      2. Comment
         1. Get comments
         2. Get comments based on the post
      3. Write unittest for minimum two functions
   2. **Note**: Use facebook-SDK for reference <https://github.com/mobolic/facebook-sdk/>