

In [1]:

```
#Question1: Who in the hell is Guido Van Rossum? Have Python print the answer, using jupyter
print("Answer1:He is the person that developed the python programming language/n
(per Wikipedia https://en.wikipedia.org/wiki/Guido\_van\_Rossum, accessed 14 Jan 2016")
```

Answer1:He is the person that developed the python programming language/n (per Wikipedia https://en.wikipedia.org/wiki/Guido_van_Rossum, accessed 14 Jan 2016

In [2]:

```
#Question2: Explain (what it is) and execute the most famous first programming exercise. Where does this #come from? Feel free to just add your answer to the iPython notebook as markdown-formatted text.
print("Answer2: The most famous first programming exercise is to print the words 'Hello world'. This phrase was apparently first used as a program output by Brian Kernighan in the documentation for the BCPL language./n per http://stackoverflow.com/questions/602237/where-does-hello-world-come-from")
print("Hello world")
```

Answer2: The most famous first programming exercise is to print the words 'Hello world'. This phrase was apparently first used as a program output by Brian Kernighan in the documentation for the BCPL language./n per <http://stackoverflow.com/questions/602237/where-does-hello-world-come-from>
Hello world

In [3]:

```
#Question3: Produce the sum of the year of your birthday and the year of this class
yearBirthday = 1990
yearClass = 2016
Answer = yearBirthday + yearClass
print(Answer)
```

4006

In [4]:

```
#Question4: Divide 3 by 2 and return the expected result. Is this what you expected? Why or why not?
print(3/2)
#Yes, the answer 1.5 is what I expected because I know that three halves is equal to 1 1/2
```

1.5

In [5]:

```
#Question5: Print the numbers 0 to 50 by 5  
#I use 51 because upper end of range is exclusive  
Answer = (list(range(0,51,5)))  
print(Answer)
```

```
[0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50]
```

In [6]:

```
#Question6: Draw and print 10 discrete numbers from a uniform distribution spanning the interval  
#[0,1000000]  
import numpy as np  
#I am using 1000001 because this value is exclusive  
print(np.random.randint(0,1000001,10))
```

```
[607585 568842 773171 331181 63784 410298 317268 803937 762567 285288]
```

In [7]:

```
#Question7: Python is what is known as a zero-indexed programming language. Illustrate what this means.  
print("This means that the first element in a list, for instance, is the considered the 0th element")  
#I create list below  
list = [1,2,3,4,5]  
#Then I take the index of the number 1 in the list, which is 0  
print(list.index(1))
```

This means that the first element in a list, for instance, is the considered the 0th element
0

In [8]:

```
import this
```

The Zen of Python, by Tim Peters

```
Beautiful is better than ugly.  
Explicit is better than implicit.  
Simple is better than complex.  
Complex is better than complicated.  
Flat is better than nested.  
Sparse is better than dense.  
Readability counts.  
Special cases aren't special enough to break the rules.  
Although practicality beats purity.  
Errors should never pass silently.  
Unless explicitly silenced.  
In the face of ambiguity, refuse the temptation to guess.  
There should be one-- and preferably only one --obvious way to do it  
.  
Although that way may not be obvious at first unless you're Dutch.  
Now is better than never.  
Although never is often better than *right* now.  
If the implementation is hard to explain, it's a bad idea.  
If the implementation is easy to explain, it may be a good idea.  
Namespaces are one honking great idea -- let's do more of those!
```

In []: