College of San Mateo
CIS 117 Python Programming

Python 3 vs Python 2

##

Be aware that there are some significant differences between Python 3 and earlier versions. For beginner Python programmers, the main ones are that the print statement of Python 2.x is now a print function in Python 3, (brackets are required after the word print (see program #01-01 below),the raw_input function in Python 2.x is replaced by the input function in Python 3,and an integer division such as 2/3 in Python 2.x is now a real division in Python 3.

For experienced programmers, also check out the range() and string formatting differences outlined here:
http://inventwithpython.com/appendixa.html

Check out the program at the bottom of this page which shows how to force printing on the same line in Python 3 compared to the old method in Python 2.x

NOTE: For CIS 117 we will use Python 3
##

Examples of differences between Python 2 and Python 3...

```
#01-01.py
# This is a Python 3 program

print ("Hello World!")
```

```
#01-01.py
# This is a Python 2.x program

print "Hello World!"
```

```
#01-04.py
# This is a Python 3 program
```

```
prompt  = "Enter a some text "
thetext = input(prompt)
print ("This is what you entered:")
print (thetext)
```

```
#01-04.py
# This is a Python 2.x program

prompt  = "Enter a some text "
thetext = raw_input(prompt)
print "This is what you entered:"
print thetext
```

```
################################################
################################################

How to force output to continue on the same line

################################################
################################################
```

```
# This is a Python 2.x program
# The , (e.g. print a,) forces the next output to continue on the same
line
# Compare the code to the output below it

a = 1
b = 2
c = 3
print a
print b
print c
print a,
print b,
print c
print a,b,c

'''
The above program gives this output in Python2.x

>>>
1
2
3
1 2 3
1 2 3
>>>

'''
```

```
##########################################
```

Now compare the program above to an equivalent program in Python 3:

```
# This is a Python 3.x program
# The , (e.g. print (a,) does not behave in the same way as it does in
Python 2.x
# Compare the code to the output below it

a = 1
b = 2
c = 3
print (a)
print (b)
print (c)
print (a,)
print (b,)
print (c)
print (a,b,c)

'''
The above program gives this output in Python 3.x

>>>
1
2
3
1
2
3
1 2 3
>>>

'''
```

```
##########################################
```

In Python 3.x, if you want to continue output on the same line,
the program should be altered to look like this:

```
# This is a Python 3.x program
# to continue output on the same line
# Compare the code to the output below it


a = 1
b = 2
c = 3
print (a)
print (b)
print (c)
print (a, end=" ")
print (b, end=" ")
print (c)
print (a,b,c)

'''
The above program gives this output in Python 3.x

>>>
1
2
3
1 2 3
1 2 3
>>>

'''


See the explanation here:
http://docs.python.org/3.0/whatsnew/3.0.html

# For experienced programmers, also check out
# the range() and string formatting differences outlined here:
# http://inventwithpython.com/appendixa.html

# For experienced programmers, also check out
# IDLE's debugging tools at:
# http://inventwithpython.com/chapter7.html
```