Python Programming

The Python programming language (version less than 3) will be used in our course. This is a miniature introduction written for the game theory student who has never used Python before.

Running Python. Python code can be run in two ways:


2. By installing Python locally on your computer. Apple and Linux machines probably already have Python installed. Windows users can visit www.python.org for installation.

If this option is chosen, code can be run by saving the code as a text file ending in “.py” using a text editor (like “Notepad” and not word processing program like “Word”). Then tell Python to run the code. Visit me if you do not know how to do this.

Select an option and try running this “hello world” program:

```python
# our first program
print "Hello world!"
```

Here are some basic commands in Python:

- `a = 4` Assigns the variable `a` to 4.
- `print a` Outputs `a` to the screen. Without a print statement somewhere, the program may not show anything.
- `s = "Hi Mom"` Assigns the variable `s` to be the string "Hi Mom".
- `a + b` Adds `a` and `b`.
- `a - b` Subtracts `a` and `b`.
- `a * b` Multiplies `a` and `b`.
- `a/float(b)` Divides `a` and `b`. Without `float`, it is rounded division.

Lists. Lists store numbers, strings of letters, and even other lists in a particular order. For example, the list containing 0, 1, and the string "abc" is [0,1,"abc"]. Here are common list commands:

- `range(n)` Gives the list [0,1,2,...,n-1].
- `len(L)` The number of elements in `L`.
- `sum(L)` Sums the elements in `L`.
- `L[i]` The `i`th element of the list `L`, where 0 is the first element, 1 is the second element, and so on.
- `zip(A,B)` A list of pairs (a,b) using interlaced elements from `A` and `B`, in the same way a zipper interlaces teeth.
- `[f(i) for i in L]` Creates a list using function `f` and list `L`.
- `[f(i) for i in L if test]` Creates a list using function `f` and list `L`, selecting only those elements for which test is true.

For example, the code

```python
# A program involving lists
A = range(5)
B = [a*a for a in A]
Z = zip(A,B)
print Z
print [a + b for (a,b) in Z if a != b]
```

produces the output:

```python
[(0, 0), (1, 1), (2, 4), (3, 9), (4, 16)]
[6, 12, 20]
```
**Functions and conditionals.** The `def` command creates functions. The syntax is

```python
def FunctionName(input):
    commands
    return output
```

When a `return` statement is found, `FunctionName` stops and `output` is returned. The indentation before the commands is important! Python needs correct spacing to run.

The syntax for a conditional statement is

```python
if test:
    command 1
    command 2
else:
    command 3
```

If `test` is `True`, then `command 1` and `command 2` are executed. If `test` is `False`, then `command 3` is executed. The `else` statement is optional.

For example, this defines a function giving the absolute value and then prints an example:

```python
def abs(x):
    if x < 0:
        return (-1)*x
    else:
        return x

print [abs(x - 2) for x in range(5)]
```

The output is the list `[2, 1, 0, 1, 2]`.

This is all the Python you need to know to get started in the class! At this point you have enough tools to understand the Python code in the Colonel Blotto assignment, shown below:

```python
# Python code implementing the Colonel Blotto contest
def A_wins(A,B):
    A_score = sum([1 for a,b in zip(A,B) if a > b])
    B_score = sum([1 for a,b in zip(A,B) if a < b])
    return A_score > B_score

def contest(L):
    return [sum([1 for B in L if A_wins(A,B)]) for A in L]

# To run a contest, modify the entries in list L below.
# The output is the scores for the entries in L.
L = [
    [11, 9, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10]
]

print contest(L)
```