Prelab5

We encourage you to work together on the prelab. The prelab is not graded but will help you prepare for your lab session. In the prelab, you may find questions to answer. We do not require you to provide us the answers, but we do recommend you to try to answer these questions. If you have any questions on the material in the prelab, first check the book and recitation slides, if you do not find your answer please email your recitation TA or the course instructors.

In this prelab you will learn some of the decision control features of Python.

While Loops

While loops, are used for repeating the execution of certain sections of code until a defined condition is met.

For example,

```python
a = 0
while a < 5 :
    a += 1
    print (a)
```

And the output is,

```
1
2
3
4
5
```

So what does the program do? First it sees the line `a = 0` and makes a zero. Then it sees while `a < 5`: and so the computer checks to see if `a < 5`. The first time, the computer sees that `a` is zero. Since it is less than 5, it runs the tabbed statements. From the statement `a += 1`, `a` is now 1 which is still less than 5, so it runs the tabbed statements again. In other words, while `a` is less than 5 the computer will run the tabbed in statements.

The following example works as a simple login procedure that allows the user to enter a password until he can successfully log in.

```python
# Waits until a password has been entered. Use control-C to break out without
# the password.

# Note that this must not be the password so that the
# while loop runs at least once.
password = "foobar"
```
#note that != means not equal

while password != "unicorn":
    password = raw_input("Password: ")
print ("Welcome in")

## For loops

Python’s `for` loop iterates over the items of any sequence (a list or a string), in the order that they appear in the sequence.

Rewriting the example while loop above, using the `for` loop, the program looks like,

```python
for num in range(1, 6):
    print (num)
```

And the output is,

```
1
2
3
4
5
```

The `for` loop goes through each number in the sequence and sets `num` to each element in the sequence. Inside the body of the loop, each element is printed.

The following example shows a function that uses a `for` loop to calculate the factorial of a number.

```python
def factorial(n):
    p = 1
    for n in range(1, n+1):
        p = p * n
    return p

>>> factorial(6)
720
```

Everything that can be done with `for` loops can also be done with `while` loops but `for` loops give an easy way to go through all the elements in a list or to do something a certain number of times.

From:  
[http://courses.cs.purdue.edu/](http://courses.cs.purdue.edu/) - Computer Science Courses

Permanent link:  
[http://courses.cs.purdue.edu/cs17700:prelab5](http://courses.cs.purdue.edu/cs17700:prelab5)

Last update: 2012/09/17 16:26