python™

Week 4

>>> Overview

* if else
* returns
* input

>>> if

Like many things in the transition from Java to Python, curly braces are replaced with colons and whitespace, the parentheses are dropped and &&, || and ! change.

Java	python
<	<
>	>
<=	<=
>=	>=
==	==
!=	!=
||	or
&&	and
!	not

Translator.java

```
1  // 1 for english
2  // 2 for german
3  int translator = 1;
4  if (translator == 1) {
5        english();
6  } else if (translator == 2) {
7        german();
8  } else {
9        System.out.println("None");
10 }
```

Notice: "else if" becomes "elif"

translator.py

```
1  translator = 1
2  if translator == 1:
3        english()
4  elif translator == 2:
5        german()
   else:
         print "None"
```

>>> strings

Just like in Java, strings are objects. Lets look at different things that we can do with them:

string methods		
s = "wow"	s.capitalize()	=> "Wow"
s = "wow"	s.endswith("w")	=> True
s = "wow"	s.find("o")	=> 1
s = "wow"	s.islower()	=> True
s = "wOw"	s.isupper()	=> False
s = "wOw"	s.lower()	=> "wow"
s = "hmmm"	s.startswith("hm")	=> True
s = " ack "	s.strip()	=> "ack"
s = "wOw"	s.swapcase()	=> "WoW"
s = "wow"	s.upper()	=> "WOW"

>>> strings as sequences

Like arrays in Java, strings have zero-based indexes, and we can access parts of them with square brackets instead of using Java's substring() and charAt() methods.

Lets look at things we can do if we have the string

s = "shocking"

sequence operations	
s[<index>]	s[3] => "c" s[-1] => "g"
s[<start>:<end>]	s[4:] => "king" s[3:5] => "ck" s[-3:] => "ing"
len(s)	len(s) => 8

Indexing

from the front 0 1 **2** 3 4 5 6 7

"shocking"

from the back -8 -7 -6 -5 **-4** -3 -2 -1

example: s[2:-4] => "oc"

>>> return

Returns in python are straightforward. Simply "return <value>" instead of "return <value>;" and forget about the types.

degrees.py

```
1  def f_to_c(degreesF):
2          degreesC = 5.0 / 9.0 * (degreesF – 32)
3          return degreesC
4
5  #main
6  temp = f_to_c(68)
7  print ("Temperature is: " + str(temp))
```

slope.py

```
1  def slope(x1, y1, x2, y2):
2          return (y2 - y1) / (x2 - x1)
3
4  #main
5  slope = slope(0, 0, 5, 5)
6  print ("Slope is: " + str(slope))
7
```

>>> math in python

Most of the math functions are the same in python. Here is a list with a short description.

In order to use them, we have to import math

The constants are the same as Java, except lower case.

math.pi = 3.1415926...
math.e = 2.7182818..

Random comes from its own class:
import random
random.random()

math functions

ceil(x)
fabs(x)
floor(x)
exp(x)
log(x,[base])
log10(x)
pow(x, y)
sqrt(x)
cos(x)
hypot(x, y)
sin(x)
tan(x)
degrees(x)
radians(x)

>>> input() vs. raw_input()

There are two ways of getting input. The first is input(). It takes in input until enter is hit and then tries to interpret it into python. However, this way only works well for numbers.

```
>>> x = input("yes? ")
yes? y
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<string>", line 1, in <module>
NameError: name 'y' is not defined
>>> x = input("yes? ")
yes? 2
>>> print x
2
>>> x = input("num? ")
num? 2.0
>>> print x
2.0
```

The second way is to use raw_input() which returns the entire line as a string. Once this is done, the string can be split into smaller strings and changed to the desired type.

inputs.py

```
1  #take a number in
2  x = input("x? ")
3  print x
4
5  #take a sentence to tokenize it
6  sent = raw_input("sentence: ")
7  for w in sent.split(" "):
8          print "word: " + w
9
10
```

>>> hours worked

Sample output:

```
Employee 1: How many days? 3
Hours? 1
Hours? 2
Hours? 9
Employee 1's total hours = 11
Employee 2: How many days? 2
Hours? 8
Hours? 10
Employee 2's total hours = 16
Total hours for both = 27
```


```
// Marty Stepp, CSE 142, Autumn 2007
// This program reads the hours worked by two employees and
// outputs the total hours for each and for both employees.

import java.util.*;   // so that I can use Scanner

public class Hours {
   public static void main(String[] args) {
      Scanner console = new Scanner(System.in);
      int totalHours = 0;
      totalHours = totalHours + processEmployee(console, 1);
      totalHours = totalHours + processEmployee(console, 2);
      System.out.println("Total hours for both = " + totalHours);
   }

   // This method reads the hours worked by one employee
   // and returns the total hours.
   public static int processEmployee(Scanner console, int num) {
      System.out.print("Employee " + num + ": How many days? ");
      int days = console.nextInt();

      // cumulative sum for hours worked each day
      int sum = 0;
      for (int i = 1; i <= days; i++) {
         System.out.print("Hours? ");
         int hours = console.nextInt();
         sum += Math.min(8, hours);   // cap to 8 hours in one day
      }
      System.out.println("Employee " + num + "'s total hours = " + sum);
      return sum;
   }
}
```