



Ondrej Kosztolanyi of Slovakia and fellow competitors collapse to the ground after the men's 10km pursuit during the Winter Youth Olympic Games on Jan. 16 in Seefeld, Austria. (Julian Finney/Getty Images)

If-then-else statements

Lecture 9 - COMP 364

February 5, 2010, updated 2012

Derek Ruths

Other containers: strings and files!

- **Strings** and **files** also implement the container interface to a certain extent!

- `s = "Hello World"`

What is `s[2]`? Intuitively, what do `s[0:5]` and `s[3:]` represent?

```
f = open('species.v9.0.txt')
for line in f:
    print line[:5]
```

```
for c in "Mus Musculus": # What are we iterating over?
    print c.lower()
```

Other ways to obtain lists

- Lists are everywhere: obtaining the argument list for a program, obtaining the list of lines in a file, creating a new list through iteration...
- Some functions that will give us lists:
 - `split(...)` will deconstruct a line according to a specified delimiter
 - `f.readlines()`

Recall...

Task 1: Working with all the elements of a list

- Often we will want to do something to each element in a list
 - In the case of “gene <start>..<>end>”, extract <start> and <end> and subtract them to get the length of a gene
 - In the case of a bunch of data points, add them together and divide by the number of data points
- Version 1: Print out all the elements
- Version 2: Print out all the elements, one element per line
- Version 3: Compute the sum of a list of numbers
- Version 4: Compute the average of a list of numbers

Task 2: Working with specific elements of a list

- Often (potentially even more often than simple looping we will want to do something to a subset of the element in the list
 - In the case of “gene <start>..<>”, we might want to print out only genes longer than 100 bps
 - We might want to average only data points that are less than some threshold
 - Count the number of data points that are in a specific range
- Version 1: Print out all strings in a list that are less than 5 characters long
- Version 2: Count the number of elements less than 10
- Version 3: Compute the sum of all numbers less than 20
- Version 4: Average all numbers less than 15

If-then-else statements

- **If:** Designate code that is executed only when a specific condition is True
- **Else:** Code that is executed when a specific condition is False
- Reminder of boolean conditions:

```
a < b  
b >= c  
b == c  
d != a  
len(mylist) > 5
```

If-then-else statements (2)

- Always starts with the if condition, but **else block is optional**.

```
if len(sys.argv[1]) > 5:
    print sys.argv[1]
else:
    print 'Too short!'
```

```
if len(sys.argv[1]) > 5:
    print sys.argv[1]
```

The else part is optional!

```
if mylist[0] != "McGill":
    print "Something is wrong, was expecting McGill, got", mylist[0]
else:
    print "Everything OK"
```

Containment

- The 'in' operator will test for membership of an element inside a sequence
- Strings: Test if a “substring” is present
Lists: Test if a certain element is in the list.

```
monty = 'Egg and spam'
has_spam = "spam" in monty
print has_spam
```

```
if 'spam' in monty:
    print "I've got spam!"
else:
    print "Where's my spam!?"
```

```
a = ['AAA', 'TTA', 'ATA']
print 'ATA' in a
print 'TAA' in a
```


Combining Iteration and If-Else

- Go through a **comma-separated file (csv)**, see if it's a core or peripheral species, and build one list for each category.
- Do “head” on the file to see what format we expect

```
## taxon_id,STRING_type,STRING_name_compact,official_name_NCBI
394,periphery,Rhizobium sp. NGR234,Rhizobium NGR234
882,core,Desulfovibrio vulgaris Hildenborough,Desulfovibrio vulgaris
Hildenborough
883,periphery,Desulfovibrio vulgaris Miyazaki,Desulfovibrio vulgaris
Miyazaki F
...
```

Combining Iteration and If-Else

```
core_list = []
periphery_list = []
for line in open('species.csv'):
    mylist = line.split(',')
    stype = mylist[1] # Can be core or periphery
    sname = mylist[2] # Name of species
    if stype == "core":
        core_list.append(sname)
    elif stype == "periphery":
        periphery_list.append(sname)

print "Core: ", len(core_list)
print "Periphery: ", len(periphery_list)
```

elif and flow control

- **elif** is a combination of else and if. Makes end result more readable.

```
foundit = False
for line in open('genes.fasta'):
    if 'ABAT01000005.1' in line:
        foundit = True
    else:
        if 'ABAT01000006.1' in line:
            foundit = False
        else:
            if foundit:
                print line
```

```
foundit = False
for line in open('genes.fasta'):
    if 'ABAT01000005.1' in line:
        foundit = True
    elif 'ABAT01000006.1' in line:
        foundit = False
    elif foundit:
        print line
```

Extra: string formatting

- When you want to print complex strings, use string substitution
- Main substitution types: %s (string), %d (digits/integers), %f (float)

```
print "This sequence is %d nucleotides long" % len(line)
print "For taxon ID %d, Species name is: %s" % (mylist[0],
mylist[1])
```