Decisions and Loops

V. Orgogozo
Dec 9 2011

In Terminal, open new screen

Open a Python Interactive prompt

$ **python**

```
Python 2.6.1 (r261:67515, Aug  2 2010, 20:10:18)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "credits" or "license" for
more information.
>>> dna = 'atcgatc'
```

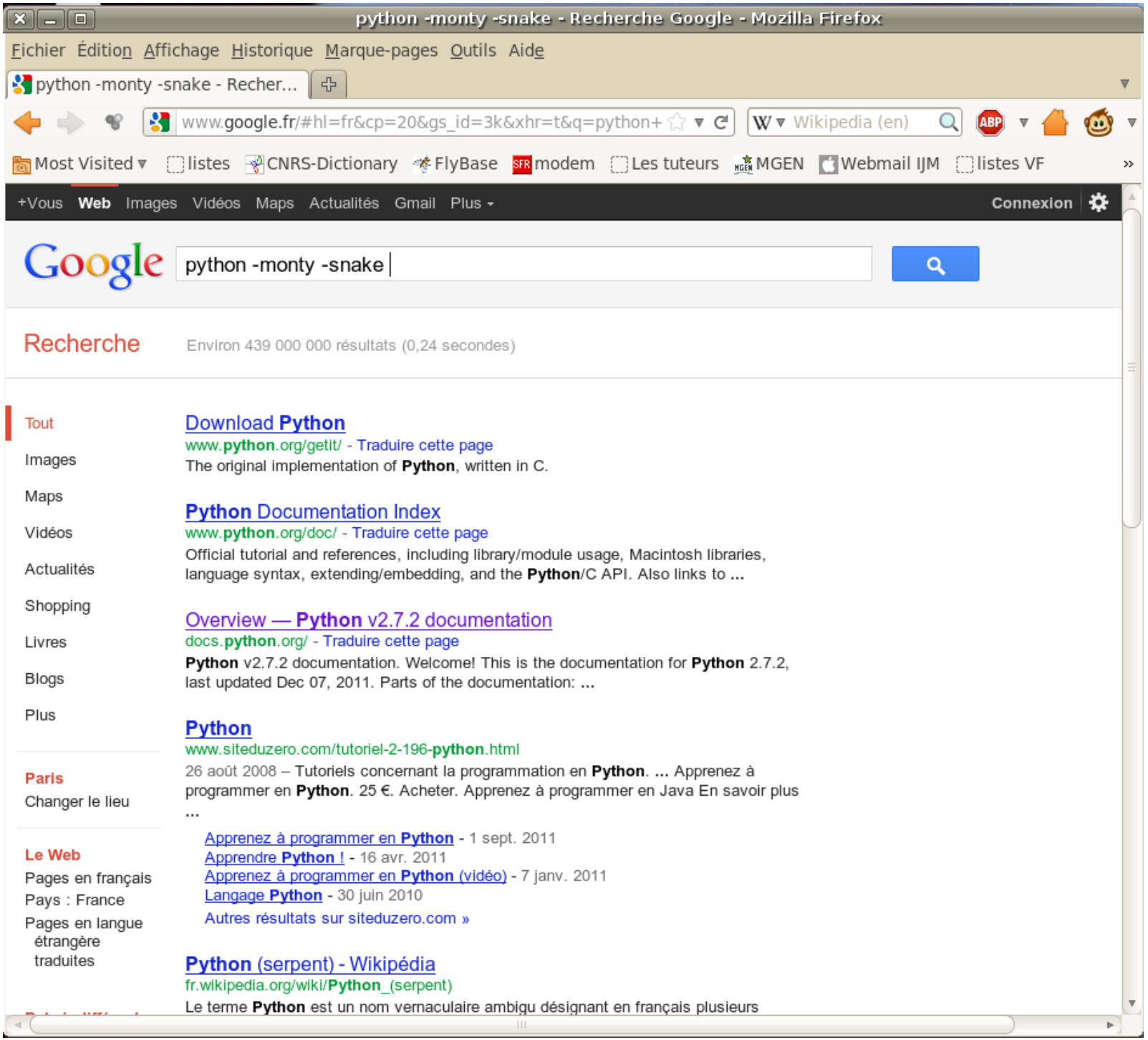
>>> dir(dna) shows all “methods” for use with strings

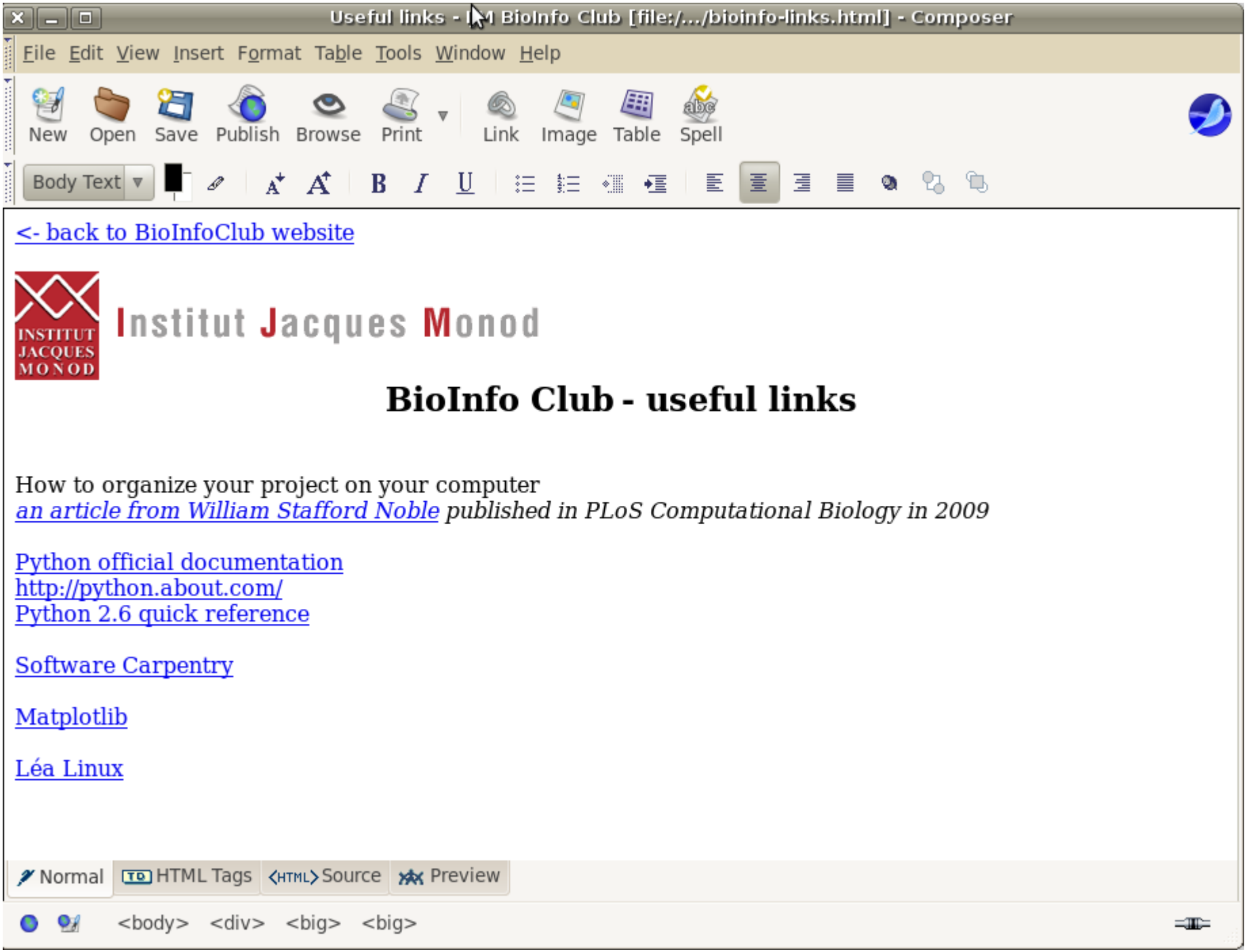
Methods are functions built into each type of variable. “dot notation”

```
>>> dna.count('atc')
2
```

>>> help(strg) shows all commands for use with strings

Ctrl+D to quit Python Interactive prompt


python -monty -snake - Recherche Google - Mozilla Firefox
Fichier Édition Affichage Historique Marque-pages Outils Aide
python -monty -snake - Recher...
www.google.fr/#hl=fr&cp=20&gs_id=3k&xhr=t&q=python+
Wikipedia (en)
Most Visited listes CNRS-Dictionary FlyBase modem Les tuteurs MGEN Webmail IJM listes VF
+Vous Web Images Vidéos Maps Actualités Gmail Plus Connexion
Google
python -monty -snake
Recherche
Environ 439 000 000 résultats (0,24 secondes)
Tout
Images
Maps
Vidéos
Actualités
Shopping
Livres
Blogs
Plus
Paris
Changer le lieu
Le Web
Pages en français
Pays : France
Pages en langue étrangère traduites
Download Python
www.python.org/getit/ - Traduire cette page
The original implementation of Python, written in C.
Python Documentation Index
www.python.org/doc/ - Traduire cette page
Official tutorial and references, including library/module usage, Macintosh libraries, language syntax, extending/embedding, and the Python/C API. Also links to ...
Overview — Python v2.7.2 documentation
docs.python.org/ - Traduire cette page
Python v2.7.2 documentation. Welcome! This is the documentation for Python 2.7.2, last updated Dec 07, 2011. Parts of the documentation: ...
Python
www.siteduzero.com/tutoriel-2-196-python.html
26 août 2008 – Tutoriels concernant la programmation en Python. ... Apprenez à programmer en Python. 25 €. Acheter. Apprenez à programmer en Java En savoir plus ...
Apprenez à programmer en Python - 1 sept. 2011
Apprendre Python ! - 16 avr. 2011
Apprenez à programmer en Python (vidéo) - 7 janv. 2011
Langage Python - 30 juin 2010
Autres résultats sur siteduzero.com »
Python (serpent) - Wikipédia
fr.wikipedia.org/wiki/Python_(serpent)
Le terme Python est un nom vernaculaire ambigu désignant en français plusieurs


Useful links - IJM BioInfo Club [file:/.../bioinfo-links.html] - Composer
File Edit View Insert Format Table Tools Window Help
New Open Save Publish Browse Print Link Image Table Spell
Body Text
<- back to BioInfoClub website
INSTITUT JACQUES MONOD
Institut Jacques Monod
BioInfo Club - useful links
How to organize your project on your computer
an article from William Stafford Noble published in PLoS Computational Biology in 2009
Python official documentation
http://python.about.com/
Python 2.6 quick reference
Software Carpentry
Matplotlib
Léa Linux
Normal HTML Tags Source Preview
<body> <big> <big>

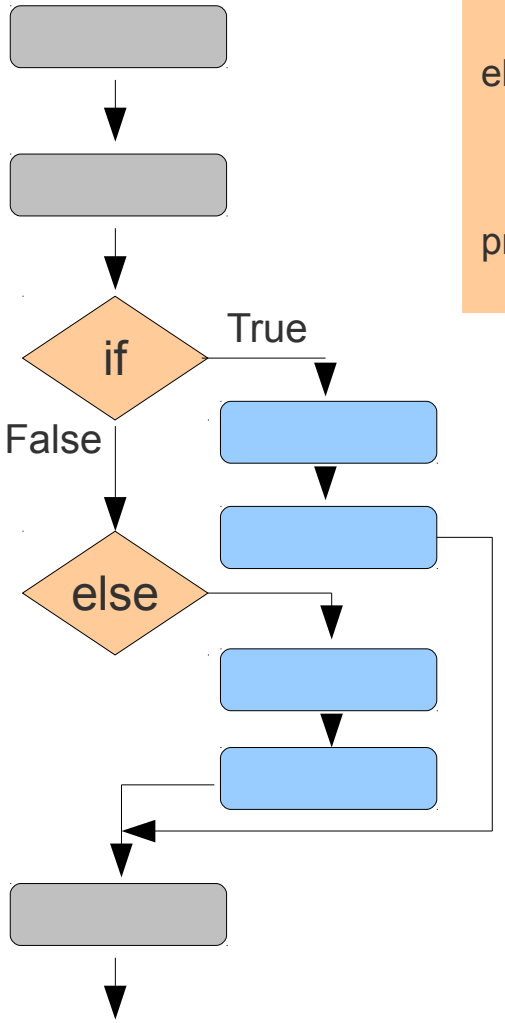
Goal:

calculate melting temperature of a given DNA sequence

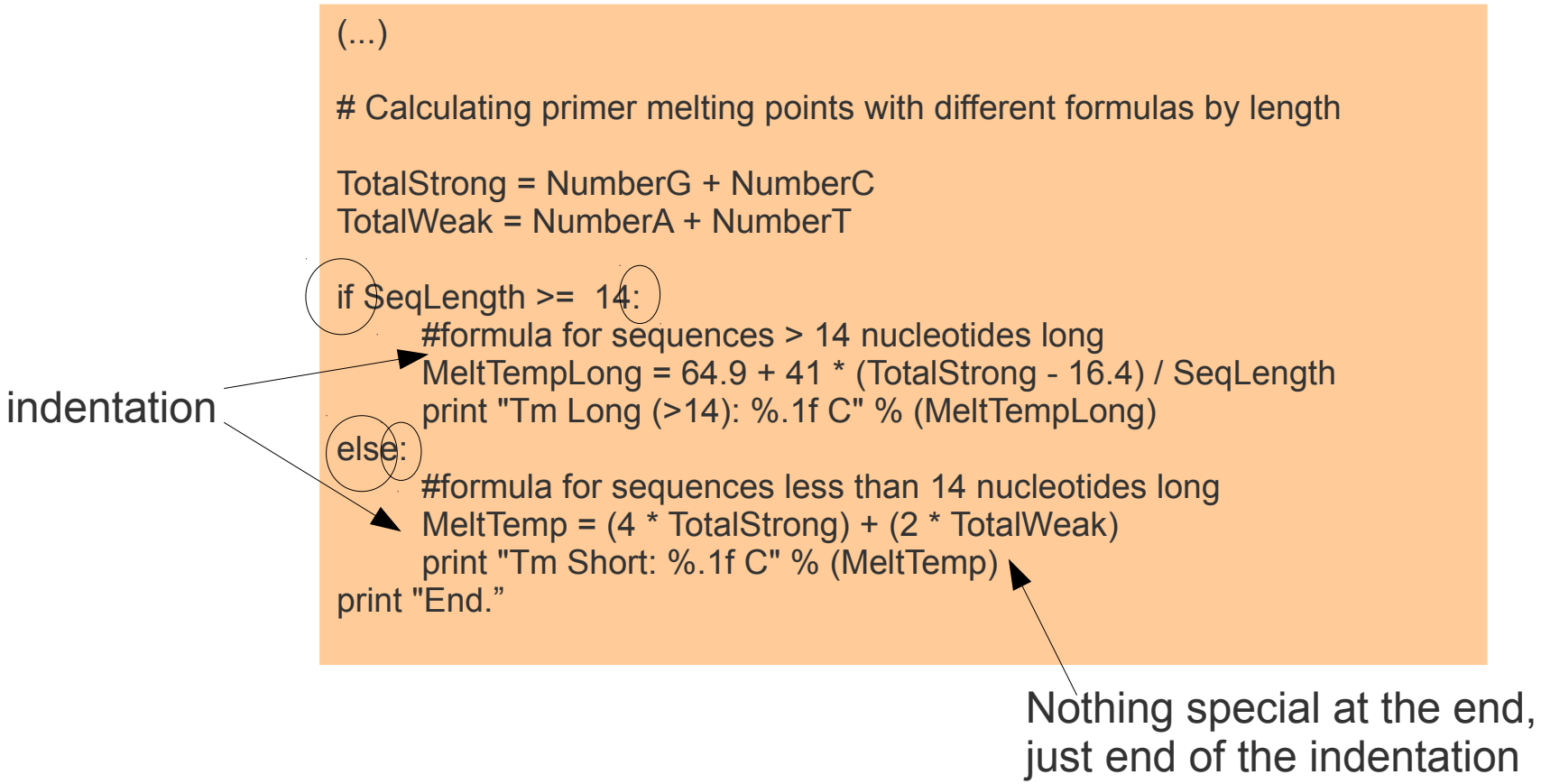
Open the dnacalc1.py file

```
#! /usr/bin/env python

DNASeq = "ATGTCTCATTCAAAGCA"

# gather user input for sequence
# this overrides the definition of DNASeq above
# DNASeq = raw_input("Enter a sequence: ")
DNASeq = DNASeq.upper()  # convert to uppercase for .count() function
DNASeq = DNASeq.replace(" ","") # remove spaces

print 'Sequence:', DNASeq

# below are nested functions: first find the length, then make it float

SeqLength = float(len(DNASeq))

print "Sequence Length:", SeqLength

NumberA = DNASeq.count('A')
NumberC = DNASeq.count('C')
NumberG = DNASeq.count('G')
NumberT = DNASeq.count('T')

# Old way to output the Numbers
# print "A:", NumberA/SeqLength
# print "C:", NumberC/SeqLength
# print "G:", NumberG/SeqLength
# print "T:", NumberT/SeqLength

# Calculate percentage and output to 1 decimal
print "A: %.1f" % (100 * NumberA / SeqLength)
print "C: %.1f" % (100 * NumberC / SeqLength)
print "G: %.1f" % (100 * NumberG / SeqLength)
print "T: %.1f" % (100 * NumberT / SeqLength)

# End of Chapter 8
```

'if-else' statement




```
(...)

# Calculating primer melting points with different formulas by length

TotalStrong = NumberG + NumberC
TotalWeak = NumberA + NumberT

if SeqLength >=  14:
        #formula for sequences > 14 nucleotides long
        MeltTempLong = 64.9 + 41 * (TotalStrong - 16.4) / SeqLength
        print "Tm Long (>14): %.1f C" % (MeltTempLong)
else:
        #formula for sequences less than 14 nucleotides long
        MeltTemp = (4 * TotalStrong) + (2 * TotalWeak)
        print "Tm Short: %.1f C" % (MeltTemp)
print "End."
```



Indentation is a requirement in Python.

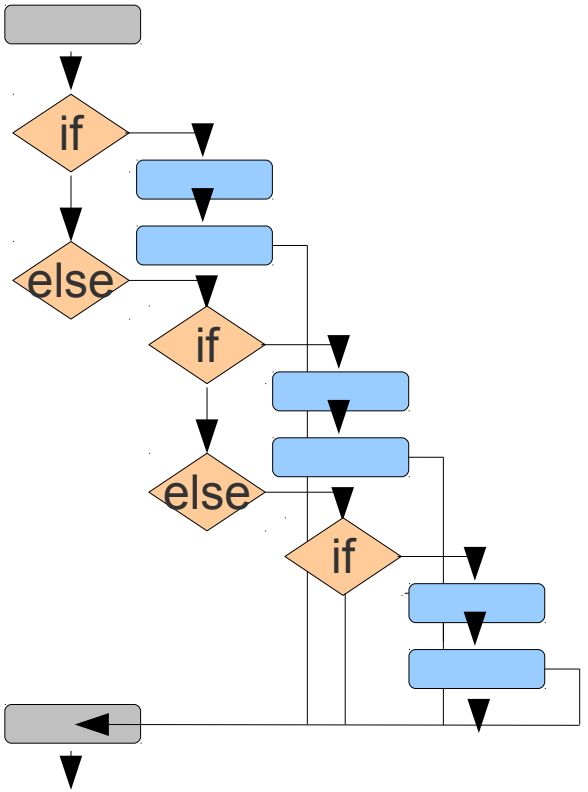
Either spaces (always same number, typically four) or tab.

	Operators	Is true if...
Comparison operators	X == y	X is equal to y
	X != y	X is not equal to y
	X > y	X is greater than y
	X >= y	X is greater than or equal to y
Logical operators	A and B	Both A and B are true
	A or B	Either A or B are true
	not B	B is false (inverts the value of B)
	(not A) or B	A is false or B is true
	not (A or B)	A and B are both false

Be careful not to put a single = sign in an “if” statement

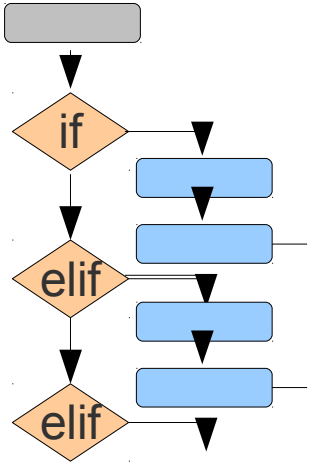
Many indentations when there are 3 cases

```
If X == 1:
        Value = “one”
else:
                if X == 2:
                                Value=”two”
                else:
                                if X == 3:
                                                Value=”three”
```



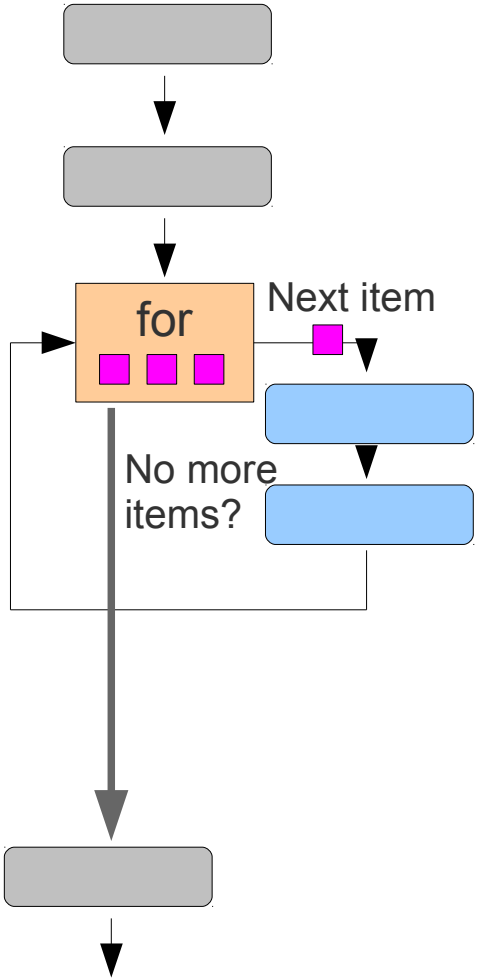
Better = use elif

```
If X == 1:
        Value = “one”
elif X == 2:
        Value=”two”
elif X == 3:
        Value=”three”
```



Goal:
calculate molecular weight of a given protein sequence

We will use the for loop



We will use a list

```
MyCollection= [1,2,3,4,5]
MyCollection=['A','T','G','C','T']
MyCollection=list('ATGCT')
```

In Python, the for loop can operate directly on each character of a string without conversion to a list.

Write a new script entitled proteincalcTEST.py

```
#!/ usr/bin/env python

ProteinSeq="FDILSATFTYGNR"

for Aa in ProteinSeq:
        print Aa
        print '-'
```

What does it do?

We will generate a dictionary with each amino acid's single letter code and its corresponding molecular weight.

```
MyDictionary = {key1:value1, key2:value2}

MyDictionary = {'A':89,09, 'R':174,20, 'N':...}

MyDictionary = {
'A':89.09,
'R':174.20,
'N':
(…)
'X':0,
'-':0,
'*':0 }
```

(1) use the aminoacid.html file that can be found on the web (also in your example folder).

(2) copy only the useful lines into your text editor

(2) copy only the useful lines into your text editor

```
<tr><td>Alanine</td><td>Ala</td><td>A</td><td>89.09</td></tr>
<tr><td>Arginine</td><td>Arg</td><td>R</td><td>174.20</td></tr>
<tr><td>Asparagine</td><td>Asn</td><td>N</td><td>132.12</td></tr>
<tr><td>Aspartic acid</td><td>Asp</td><td>D</td><td>133.10</td></tr>
<tr><td>Cysteine</td><td>Cys</td><td>C</td><td>121.15</td></tr>
<tr><td>Glutamine</td><td>Gln</td><td>Q</td><td>146.15</td></tr>
<tr><td>Glutamic acid</td><td>Glu</td><td>E</td><td>147.13</td></tr>
<tr><td>Glycine</td><td>Gly</td><td>G</td><td>75.07</td></tr>
<tr><td>Histidine</td><td>His</td><td>H</td><td>155.16</td></tr>
<tr><td>Isoleucine</td><td>Ile</td><td>I</td><td>131.17</td></tr>
<tr><td>Leucine</td><td>Leu</td><td>L</td><td>131.17</td></tr>
<tr><td>Lysine</td><td>Lys</td><td>K</td><td>146.19</td></tr>
<tr><td>Methionine</td><td>Met</td><td>M</td><td>149.21</td></tr>
<tr><td>Phenylalanine</td><td>Phe</td><td>F</td><td>165.19</td></tr>
<tr><td>Proline</td><td>Pro</td><td>P</td><td>115.13</td></tr>
<tr><td>Serine</td><td>Ser</td><td>S</td><td>105.09</td></tr>
<tr><td>Threonine</td><td>Thr</td><td>T</td><td>119.12</td></tr>
<tr><td>Tryptophan</td><td>Trp</td><td>W</td><td>204.23</td></tr>
<tr><td>Tyrosine</td><td>Tyr</td><td>Y</td><td>181.19</td></tr>
<tr><td>Valine</td><td>Val</td><td>V</td><td>117.15</td></tr>
<tr><td>Unknown</td><td>Xaa</td><td>X</td><td>0.0</td></tr>
<tr><td>Gap</td><td>Gap</td><td>-</td><td>0.0</td></tr>
<tr><td>Stop</td><td>End</td><td>*</td><td>0.0</td></tr>
```

(3) use regular expressions to change this text into:

HINT: use [\d\.]+
The beginning of your query should be:
.+(.)

```
'A':89.09,
'R':174.20,
'N':132.12,
'D':133.10,
'C':121.15,
'Q':146.15,
'E':147.13,
'G':75.07,
'H':155.16,
'I':131.17,
'L':131.17,
'K':146.19,
'M':149.21,
'F':165.19,
'P':115.13,
'S':105.09,
'T':119.12,
'W':204.23,
'Y':181.19,
'V':117.15,
'X':0.0,
'-':0.0,
'*':0.0,
```

(2) copy only the useful lines into your text editor

```
<tr><td>Alanine</td><td>Ala</td><td>A</td><td>89.09</td></tr>
<tr><td>Arginine</td><td>Arg</td><td>R</td><td>174.20</td></tr>
<tr><td>Asparagine</td><td>Asn</td><td>N</td><td>132.12</td></tr>
<tr><td>Aspartic acid</td><td>Asp</td><td>D</td><td>133.10</td></tr>
<tr><td>Cysteine</td><td>Cys</td><td>C</td><td>121.15</td></tr>
<tr><td>Glutamine</td><td>Gln</td><td>Q</td><td>146.15</td></tr>
<tr><td>Glutamic acid</td><td>Glu</td><td>E</td><td>147.13</td></tr>
<tr><td>Glycine</td><td>Gly</td><td>G</td><td>75.07</td></tr>
<tr><td>Histidine</td><td>His</td><td>H</td><td>155.16</td></tr>
<tr><td>Isoleucine</td><td>Ile</td><td>I</td><td>131.17</td></tr>
<tr><td>Leucine</td><td>Leu</td><td>L</td><td>131.17</td></tr>
<tr><td>Lysine</td><td>Lys</td><td>K</td><td>146.19</td></tr>
<tr><td>Methionine</td><td>Met</td><td>M</td><td>149.21</td></tr>
<tr><td>Phenylalanine</td><td>Phe</td><td>F</td><td>165.19</td></tr>
<tr><td>Proline</td><td>Pro</td><td>P</td><td>115.13</td></tr>
<tr><td>Serine</td><td>Ser</td><td>S</td><td>105.09</td></tr>
<tr><td>Threonine</td><td>Thr</td><td>T</td><td>119.12</td></tr>
<tr><td>Tryptophan</td><td>Trp</td><td>W</td><td>204.23</td></tr>
<tr><td>Tyrosine</td><td>Tyr</td><td>Y</td><td>181.19</td></tr>
<tr><td>Valine</td><td>Val</td><td>V</td><td>117.15</td></tr>
<tr><td>Unknown</td><td>Xaa</td><td>X</td><td>0.0</td></tr>
<tr><td>Gap</td><td>Gap</td><td>-</td><td>0.0</td></tr>
<tr><td>Stop</td><td>End</td><td>*</td><td>0.0</td></tr>
```

Search for:

```
.+(.)</td><td>([\d\.]+)</td></tr>
```

Replace by:

```
'$1':$2,
```

(3) use regular expressions to change this text into:

```
'A':89.09,
'R':174.20,
'N':132.12,
'D':133.10,
'C':121.15,
'Q':146.15,
'E':147.13,
'G':75.07,
'H':155.16,
'I':131.17,
'L':131.17,
'K':146.19,
'M':149.21,
'F':165.19,
'P':115.13,
'S':105.09,
'T':119.12,
'W':204.23,
'Y':181.19,
'V':117.15,
'X':0.0,
'-':0.0,
'*':0.0,
```

HINT: use [\d\.]+
The beginning of your query should be:
.+(.)

(4) copy and paste in your script file

```
#!/ usr/bin/env python

ProteinSeq="FDILSATFTYGNR"

AminoDict={
'A':89.09,
'R':174.20,
'N':132.12,
(…)
'-':0.0,
'*':0.0}

for Aa in ProteinSeq:
    print Aa, AminoDict[Aa]

MolWeight = 0
for Aa in ProteinSeq:
    MolWeight = MolWeight + AminoDict[Aa]
Print “Molecular Weight: %.1f” % (MolWeight)
```

Improvements

Use the .get() function so that unknown characters are taken into account

`AminoDict.get('A')` is equivalent to **`AminoDict['A']`**
Here use **`AminoDict.get(Aa,0.0)`**

The .keys() and .values() functions might be useful as well.

`print AminoDict.values()` return them in an unpredictable order
use `sorted(AminoDict.values())` for the alphabetical order

Check the compositioncalc1.py file

What does it do?

```
#! /usr/bin/env python
DNASeq = "ATGTCTCATTCAAAGCA"
SeqLength = float(len(DNASeq))

BaseList = list(set(DNASeq))
for Base in BaseList:
    Percent = 100 * DNASeq.count(Base) / SeqLength
    print "%s: %4.1f" % (Base,Percent)
```