Programming for Biologists

Learning Objectives

- Writing a function
- Understanding namespace of variables
- Assertions

Defining Functions

- Function definition statements are compound statements (made up of more than one line of code).
- The first line of each compound statement must have a colon.
- Subsequent lines are indented relative to the first (4 spaces - do not use tabs):

```
def name(parameter-list):
  body
  return value
```

Structure of the function

- The body consists of one or more statements.
- Functions return a value using a return statement.
- The value can be a list or other data structure (more on that later).
- You can use **pass** to do nothing.
- The parameter-list is a list of names separated by commas (can be empty).

```
def name(parameter-list):
  body
  return value
```

More about parameters

- Default values to parameters can be set by assigning values to the parameters in the definition statement.
- If the function is called with no parameter value specified, the default will be used.
- Parameters have to be passed in the specified order unless they are named when called.

```
def name(parameter-list):
  body
  return value
```

Example function

```
def validate_base_sequence(base_sequence):
  seq = base_sequence.upper()
  print(seq)
  return len(seq) == (seq.count('T')+seq.count('C')+
                      seq.count('G')+seq.count('A'))
```

1. Function validate_base_sequence accepts one parameter (base_sequence).
2. The variable seq is assigned an upper case version of the parameter.
3. The return statement evaluates and compares the length of the sequence to the total count of A, T, G, C and then returns the result of the comparison.

Function Execution

```
seq = 'aaat'
result = validate_base_sequence('tatata')
print(result,'\n',seq)
```

Function Results

```
TATATA  <-----From print in function
True    <-----Value of result
 aaat   <-----Value of seq
```

- A variable called **seq** is already present in the environment.
- The function is called using ‘tatata’ as the parameter.
- There is also a variable **seq** in the function and it only exists in the function and as long as the function is being executed.
  - This is called the variable’s namespace.
- Note that the print command in the function prints the value of seq in the function, where as the print in the console prints value of seq in the environment.

Assertions

- At various points of a function we should check assumptions or invariants to see if they are correct and, if not, terminate the code.
- For example, assumptions on the data should be checked at function entry (called "precondition checks") or at function exit (called "postcondition checks").
- An example we saw earlier was checking that a DNA sequence consists only of valid bases.

Assertions

- At other points in the code e.g. in loops, we may know that a certain value should not occur e.g. we may know that the size of a DNA fragment should be larger than a certain size and not not negative, and it is an error if it is.
- Thus, if there is some anomaly during the run of the pipeline it will be detected early and can be corrected lest invalid results are produced.

Notice the different responses.

```
seq="GCT"
assert validate_base_sequence(seq), "DNA sequence must have only A,G,C,T"
```

```
GCT
```

```
seq="GCTN"
assert validate_base_sequence(seq), "DNA sequence must have only A,G,C,T"
```

```
GCTN
---------------------------------------------------------------------------
AssertionError                            Traceback (most recent call last)
<ipython-input-4-425d58b4721b> in <module>()
      1 seq="GCTN"
----> 2 assert validate_base_sequence(seq), "DNA sequence must have only A,G,C,T"

AssertionError: DNA sequence must have only A,G,C,T
```

Exercise

Write a function that will convert Fahrenheit to Celcius.

Call the function **f2c**

Input parameter is f, a numerical value.

Output is the celsius value.