Week One: Activity Two

Printing Code

In [1]:

```
##To print code
print("Hello World")
```

```
Hello World
```

In [9]:

```
#Print a message of your choice
print("Welcome to Class!!! :-)")
```

```
Welcome to Class!!! :-)
```

Variables

In [2]:

```
#assign a variable name to a DNA sequence
mydna="ATAC"
```

In [3]:

```
#print mydna
print(mydna)
```

```
ATAC
```

In [10]:

```
#Print the statement "My sequence is: ATAC" by combining the mydna variable and "My sequence is:"
in the print statement
print("My sequence is:",mydna)
```

```
My sequence is: ATAC
```

Data Types

In [9]:

```
#assign the following variable names and check the type
#1,"twenty", 16.8, "dna",1/2
#example
x=1
type(x)
```

Out[9]:

```
int
```

In [11]:

```
#"twenty"
y="twenty"
type(y)
type(y)
```

Out[11]:

```
str
```

In [12]:

```
#16.8
z=16.8
type(z)
```

Out[12]:

```
float
```

In [13]:

```
#"dna"
a="dna"
type(a)
```

Out[13]:

```
str
```

In [15]:

```
#1/2
b=1/2
type(b)
```

Out[15]:

```
float
```

Analysing Data

In [6]:

```
#data
myseq="ATACTATATATATCATAGATACATGACATACAGATACAGATCATAATCATGATACTA"
```

In [7]:

```
#get the length
len(myseq)
```

Out[7]:

```
57
```

In [16]:

```
#how many A's
#use count()
myseq.count("A")
```

Out[16]:

```
26
```

In [18]:

```
#Add the following to myseq "ATATATAT"
mynewseq=myseq+"ATATATAT"
```

In [20]:

```
#replace C with G
correctseq=mynewseq.replace("C","G" )
```

In [21]:

```
#print the first letter
print(correctseq[0])
```

```
A
```

In [22]:

```
#print letters 2 to 4
print(correctseq[1:4])
```

```
TAG
```

In [23]:

```
#print letters 2 to 3
print(correctseq[1:3])
```

```
TA
```

Activity Three

Now we know the basics you should be able to do the following exercise:

The sequence "ATACATGATACAGATCAGATACAGATACGATACATAGATATACTAGATCAT" is included in the notebook, with this sequence carry out the following:

- Assign the sequence the variable name seq1
- Find the length of the sequence and print the following statement, "My DNA has x bases" with x being the length of the sequence.
- Count how many of each base is in seq1
- Replace all the T's with C and call it replaceseq
- The scientist that gave you the sequence forgot the first bases, add "CAGCAG" to the start of seq1
- Extract the first 10 bases

In [38]:

```
#make the seq1 variable
seq1="ATACATGATACAGATCAGATACAGATACGATACATAGATATACTAGATCAT"
```

In [39]:

```
#get the length
len(seq1)
#print the statement with the length
print("My DNA has", len(seq1),"bases")
```

```
My DNA has 51 bases
```

In [28]:

```
#count the number of each base
#A
seq1.count("A")
```

Out[28]:

```
23
```

In [29]:

```
#C
seq1.count("C")
```

Out[29]:

```
8
```

In [30]:

```
#T
seq1.count("T")
```

Out[30]:

```
13
```

In [31]:

```
#G
seq1.count("G")
```

Out[31]:

```
7
```

In [36]:

```
#replace t with c
replaceseq=seq1.replace("T","C")
replaceseq
```

Out[36]:

```
'CAGCAGCAGCAGACACACGACACAGACCAGACACAGACACGACACACAGACACACCAGACCAC'
```

In [34]:

```
#add "CAGCAG" to the start of seq1
seq1="CAGCAG"+seq1
seq1
```

Out[34]:

```
'CAGCAGCAGCAGATACATGATACAGATCAGATACAGATACGATACATAGATATACTAGATCAT'
```

In [35]:

```
#extract the first 10 bases
seq1[0:10]
```

Out[35]:

```
'CAGCAGCAGC'
```