

# Python for biologists

## Lesson 4

### User data input and 'while' loops

```
def complementary(seq):  
    nt_comp = {  
        'A' : 'T',  
        'C' : 'G',  
        'G' : 'C',  
        'T' : 'A',  
    }  
    for nt in seq:  
        compseq += nt_comp[nt]  
    return compseq
```



Alvaro Sebastián Yagüe

[www.sixthresearcher.com](http://www.sixthresearcher.com)



@SixthResearcher

# Data input, conversion and validation

## Data input

---

The **input function** shows a given text between parenthesis (for ex. a question) asking the user to write some data and push *Enter*. The entered data will be stored into a variable.

**variable = input(<string>)**

```
>>> name = input("What's your name? ")
What's your name? Michael
>>> print("Nice to meet you " + name + "!")
Nice to meet you Michael!
>>> age = input("Your age? ")
Your age? 45
>>> if int(age)<30:
    print(name+", you are still young, only "+age+" years old.")
elif int(age)>=30:
    print(name+", you are getting old, already "+age+" years old.")

Michael, you are getting old, already 45 years old.
```

## Data conversion

---

The value retrieved by the 'input' function is a *string*.

- If we want to use it as a number we will need to use the function 'int' to convert it into a **integer** (number without decimals).
- Or the function 'float' to convert the string into a floating point number (equivalent to a decimal for most of the cases).
- In the other side, if we want to convert a number into a string we will use 'str' .

```
integer = int(<string>)  
float   = float(<string>)  
string  = str(<number>)
```

## Data conversion

---

```
>>> number = input("Write a number: ")
Write a number: 23
>>> number**2
Traceback (most recent call last):
  File "<pyshell#35>", line 1, in <module>
    number**2
TypeError: unsupported operand type(s) for ** or pow(): 'str' and 'int'
>>> int(number)**2
529
>>> float(number)**2
529.0
```

- The function 'input' retrieves a string value.
- Arithmetic operations demand numbers.

```
>>> number = 123
>>> print("We have the number "+number)
Traceback (most recent call last):
  File "<pyshell#43>", line 1, in <module>
    print("We have the number "+number)
TypeError: Can't convert 'int' object to str implicitly
>>> print("We have the number "+str(number))
We have the number 123
```

- The function 'print' only accepts as argument a string.

## Data validation

---

If there is an error in the execution of the code, the program will stop. Sometimes we will want the program to continue. To skip an error we can use **'try'** and **'except'** statements, that in common language could be translated like: **'in case of error do the following'**.

```
>>> number = input("Write a number: ")
Write a number: Hello
>>> try:
    number = float(number)
    print(str(number)+' is a number.')
except:
    print(str(number)+' is not a number.')

Hello is not a number.
```

<b>try:</b>	<code>&lt;code&gt;</code>
<b>except:</b>	<code>&lt;code&gt;</code>

# Let's do bad things...

```
>>> number = 123
>>> try:
    first_char = number[1]
except:
    print("Numbers cannot be treated as lists or strings.")

Numbers cannot be treated as lists or strings.
>>> dict1 = {'A':1, 'B':2, 'C':3}
>>> try:
    dict1[2]
except:
    print("Dicts cannot be treated as lists.")

Dicts cannot be treated as lists.
>>> try:
    dict1.sort()
except:
    print("Dicts cannot be sorted.")

Dicts cannot be sorted.
>>> try:
    print "Hello"
except SyntaxError:
    print("You forgot the parenthesis.")

SyntaxError: Missing parentheses in call to 'print'
```

**But try...except  
doesn't work  
for everything!**



# 'While' loops



# While loops

---

A **loop** is a block of code that is executed several times.

**While loops** are repeated indefinitely 'while' a stated condition remains *True*.

```
>>> number = 0
>>> while number != '5':
    number = input("Please enter 5: ")
```

```
Please enter 5: hi
Please enter 5: five
Please enter 5: piec
Please enter 5: 5
```

```
>>> temperature=15
>>> while temperature<20:
    temperature=temperature+1
    print(str(temperature)+" degrees")
```

```
16 degrees
17 degrees
18 degrees
19 degrees
20 degrees
```

**while** <condition>:  
    <code>

# We can mix loops and conditional statements

```
>>> temperature=0
>>> while temperature<35:
    temperature=temperature+5
    if temperature<15:
        print(str(temperature)+" degrees is cold")
    elif temperature<25:
        print(str(temperature)+" degrees is warm")
    else:
        print(str(temperature)+" degrees is hot")

5 degrees is cold
10 degrees is cold
15 degrees is warm
20 degrees is warm
25 degrees is hot
30 degrees is hot
35 degrees is hot
```

# While loops

```
>>> from random import randint
>>> number = randint(1,100)
>>> prediction = 0
>>> while prediction != number:
    prediction = int(input("New number: "))
    if prediction > number :
        print("Number too large")
    elif prediction < number :
        print("Number too small")
    else:
        print("Congratulation. You made it!")
```

```
New number: 50
Number too small
New number: 75
Number too small
New number: 87
Number too small
New number: 93
Number too large
New number: 90
Number too large
New number: 88
Congratulation. You made it!
```

Let's play to  
guess the  
number 😊

```
>>> while prediction != number:
    prediction = int(input("New number: "))
    if prediction > number :
        print("Number too large")
    elif prediction < number :
        print("Number too small")
    else:
        print("Congratulation. You made it!")

New number: Hello
Traceback (most recent call last):
  File "<pyshell#19>", line 2, in <module>
    prediction = int(input("New number: "))
ValueError: invalid literal for int() with base 10: 'Hello'
```

# Make the code more robust

```
>>> prediction=0
>>> while prediction != number:
    prediction = input("New number: ")
    try:
        prediction = int(prediction)
        if prediction > number :
            print("Number too large")
        elif prediction < number :
            print("Number too small")
        else:
            print("Congratulation. You made it!")
    except:
        print(prediction,"is not a number, try again...")
        prediction = 0
```

```
New number: Hello
Hello is not a number, try again...
New number: 70
Number too small
New number: 88
Congratulation. You made it!
```

## While loops

---

Or sing a song!

```
>>> lines = list()
>>> line = 'something'
>>> while line != '':
    line = input('Next line: ')
    lines.append(line)
```

```
Next line: We all live in a yellow submarine,
Next line: yellow submarine, yellow submarine.
Next line:
```

```
>>> song = list()
>>> song.extend(lines*2)
>>> while song:
    line = song.pop(0)
    print(line)
```

```
We all live in a yellow submarine,
yellow submarine, yellow submarine.
```

```
We all live in a yellow submarine,
yellow submarine, yellow submarine.
```

# Exercise: Calculating the factorial of a number

## Exercise: Calculating the factorial of a number

---

Write a 'while' loop to calculate the factorial of a number.

### Factorial

---

From Wikipedia, the free encyclopedia

In [mathematics](#), the **factorial** of a [non-negative integer](#)  $n$ , denoted by  $n!$ , is the [product](#) of all positive integers less than or equal to  $n$ . For example,

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120.$$

# Python for scientists

## Next lesson... 'for' loops and functions

```
def complementary(seq):  
    nt_comp = {  
        'A' : 'T',  
        'C' : 'G',  
        'G' : 'C',  
        'T' : 'A',  
    }  
    for nt in seq:  
        compseq += nt_comp[nt]  
    return compseq  
for nt in seq:  
    compseq += nt_comp[nt]  
return compseq
```

