**Python Basics: syntax, data types and statements**

**Characteristic features:**

1. Python doesn't require explicit declaration of the data type for a variable.
   *name = "Alex"*
   *age = 40*
   *print(name, age)*

2. Blocks of code are identified by **indentation**. Example: calculation of abs value
   *x = -5*
   *if x > 0 :*
   *print(x)*
   *else :*
   *print(-x)*

3. New operators:
   a. Exponent operator: **
   b. Logical operators: and, or, not
   c. Identicality checking: is, is not
   d. Checking if a value is present in a sequence: in, in not

4. Conditional statement
   *if condition :*
   *s1*
   *else :*
   *s2*

   *if … elif … else*

5. Loop
   *for i in list :*
   *statement*

   *while condition :*
   *statement*

   The range() function can be used to create a list of values over which our loop can iterate. Returns an iterator object.

6. The **pass** statement does nothing
7. Functions
   *def my_function(params) :*
   *statements*
   *return output*

   Parameters can have default values.

8. You can build modules, packages, libraries from functions

**Data types**: int, float, bool, str, Decimal, Fraction, complex numbers (3+5j)
Digits of numbers can be accessed if converted to a string. Strings are immutable. For changing characters in a string, we have to convert it to a list.

**Strings**: we can index characters of strings and slice strings (the start is included, the end is excluded, so `s[:i] + s[i:]` is always equal to s)

*word = 'Python'*
*word[0], word[-1], word[0:2], word[:2], word[4:], word[-2:]*

**Data structures (containers):**

1. List (array of elements): might contain comma-separated items of different types between square brackets (a list can be indexed and sliced the same way as strings).
   Eg.: squares = [1, 4, 9, 16, 25].
   a. We can change the content of a list (squares[0] = 2),
   b. append elements (squares.append(36)), and
   c. remove elements (squares[2:5] = [ ]).
   d. The len() function applies to lists (len(squares)).
   e. We can create nested lists:
   *a = ['a', 'b', 'c']*
   *n = [1, 2, 3]*
   *x = [a, n]*
   So x[0] equals a and x[1] equals n. And x[0][1] equals 'b'.
   f. The llist(obj) function is used to convert an object to list.

   Iterating through a list:
   ```
   for index in range(len(L)):
       value = L[index]
       ...
   for index, value in enumerate(L):
       ...
   ```
   *Iterator function (iterates over all pairs of elements of a list):*
   ```
   def all_pairs(L):
       n = len(L)
       for i in range(n):
           for j in range(i + 1, n):
               yield (L[i], L[j])
   ```
   *How to initialize a list:*
   ```
   n = 5
   squared_numbers = [x ** 2 for x in range(n + 1)]
   t = [0  for _ in range(n) ]
   ```

2. N-tuple: immutable list of elements in brackets, where elements can be of different types.

3. Dictionary: comma-separated key:value pairs between curly brackets, such as {'the': 4, 'bread': 1, 'is': 6} , where the keys and values are separated by a colon.
   Iterating through a dictionary: *for key, value in dic.items():*
   ...

4. Set: unordered, unindexed and unchangeable collection of elements in curly brackets, where duplicates are not allowed

**Input / output:**

Taking input from stdin:

```
import sys
data = sys.stdin.readlines()
print "Counted", len(data), "lines."

inp = input("Type anything")
print(inp)
```

```
import fileinput
with fileinput.input(files = ('first.txt', 'second.txt')) as f:
       for line in f:
              print(line)
```

**Write / read files:**

```
f = open("file.txt", "a")
f.write("Now the file has more content!")
f.close()

#open and read the file after the appending:
f = open("file.txt", "r")
print(f.read())
```

**Exercises:**

1. **Guess my number**

```
from random import seed
from random import randint
# seed random number generator
seed(1)
my_number = randint(0, 10)
user_guess = 0

while user_guess != my_number:
        print('Can you guess my number?')
        user_guess = int(input())                    # You need to cast string input

        if user_guess > my_number :
           print('smaller')
        elif user_guess < my_number :
           print('greater')

print('You have correctly guessed my  number!')
```

2. **How to implement a switch-case in Python**

```
def switch_demo(argument):
    switcher = {                                    # Dictionary definition
        1: "January",
        2: "February",
        3: "March",
        4: "April",
        5: "May",
        6: "June",
        7: "July",
        8: "August",
        9: "September",
        10: "October",
        11: "November",
        12: "December"
    }
    print(switcher.get(argument, "Invalid month"))
```

Call of function : switch_demo(2)

3. **Print Fibonacci series up to n**

```
def fib(n):
    a, b = 0, 1
    while a < n:
        print(a, end=' ')
        a, b = b, a+b
    print()
```

Extension: Count elements of Fibonacci series and print the series up to count n

Return Fibonacci series as list

```
def fib2(n):
    result = [ ]
    a, b = 0, 1
    while a < n:
        result.append(a)
        a, b = b, a+b
    return result
```

4. **List statistics:** write a function for counting each distinct element of a list and print out the statistics to stdout / file / return this list of counts.
5. Write a basic **calculator** program
6. **Slicing a string**. Check if a sentence is correct. A sentence is correct if starts with big capital letter and ends with a punctuation mark.

7. **Translator**: create a dictionary file. Using this file you are able to translate words from one language to another.
8. Greatest pair in list: create all pairs of a list and select the pair with greatest sum.
9. Closest to average: find the element of a list which is closest to the list average.
10. Count determiners (a and an) in a text (list of strings).
11. Check if the elements in a list are monotone increasing / decreasing.