python-exercices-01

September 4, 2016

1 Head-first Python

Author: christophe@pallier.org

1.0.1 Can you guess what the following program does?

```
In [ ]: if 2 + 2 == 5:
            print("Oops")
        else:
            print("Fine")
            print("2 + 2 = 4")
```

Remark: Python uses **indentation** rather than delimiters to regroup series of instructions into a block

1.0.2 Can you guess what the following program does?

```
In [ ]: name = raw_input("What is your name? ")
        print("Hello " + name + "!")
```

Remarks: * '+' as the string concatenation operator * (in Python3: raw_input -> input)

1.0.3 Can you guess what the following program does?

```
In [ ]: x = raw_input("Enter a number: ")
        if int(x) % 2 == 0:  # '%' is the modulo operator
            print(x + " is even")
        else:
            print (x + " is odd")
```

1.0.4 Can you guess what the following program does?

```
In [ ]: while True:
            x = raw_input('Enter a number ("q" to stop)')
            if x == 'q':
                break  # interupts the while loop
            if int(x) % 2 == 0:  # '%' is the modulo operator
                print(x + " is even")
```

```
        else:
            print (x + " is odd")
```

1.0.5 Can you guess what the following programs do?

```
In [ ]: x = 1
        while x <= 10:
            print(x)
            x = x + 1
```

```
In [ ]: reg1, reg2 = 5, 2
        while reg1 > 0:
            reg2 += 1
            reg1 -= 1
        print(reg1, reg2)
```

1.0.6 Using a 'while' loop, write a program which computes the sum of the first 'n' integers.

...

1.0.7 A few remarks about python lists

```
In [ ]: # the elements do not have to be of the same type
        a = [3, 4, 'dog']
        print(a[0]) # change the index 0 into 1, then 2
```

```
In [ ]: # you can even embbed lists inside lists
        sentence = [['the', 'dog'], [ 'bites', [['the', 'cat']]]]
        sentence[1][1]
```

```
In [ ]: a = [1, 2, 3]
        b = a
        a[1] = 666
        b
```

This is because `a` and `b` point to the same memory area!

```
In [ ]: a = [1, 2, 3]
        b = a[:]  # makes a copy in memory
        a[1] = 666
        b
```

```
In [ ]: a = [1, 2, 3]
        b = a
        a = [4, 5, 6]
        b
```

1.0.8 Can you guess what the following program does?

```
In [ ]: sum = 0
        for i in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]:
                sum = sum + i
        print(sum)
```

Remark: we could have used the expression 'for x in range(11)'

1.0.9 Write a program that prints the multiplication tables from 1 to 10

...

Remarks:

```
In [ ]: # python's for has some nifty features
        for n, item in enumerate(['alpha', 'beta', 'gamma']):
            print n, item

        for _ in range(10):
            print('All work and no play make Jack a dull boy!')
```

1.0.10 Defining functions in python with 'def'

```
In [ ]: def somme(n):
            sum = 0
            for i in range(n + 1):
                sum = sum + i
            return sum

        print(somme(10))
        print(somme(20))
```

1.0.11 Write a function that returns the average of a list of numbers

...

1.0.12 Given the following function, write a program that prints the prime numbers below 100

```
In [36]: def is_factor(a, b):
             return b % a == 0

         print(is_factor(3, 27))
         print(is_factor(3, 28))
```

```
True
False
```

1.0.13 Guess what the following program does

```
In [ ]: def prod(x):
                p = 1
                for e in x:
                        p = p * e
                return p

        prod([4, 5, 6])
```

1.0.14 Modify the following program to draw regular polygon of 'n' sides

```
In [41]: import turtle
         turtle.forward(100)
         turtle.left(120)
         turtle.forward(100)
         turtle.left(120)
         turtle.forward(100)
         turtle.done()
```

1.0.15 Functions that call themselves are called 'recursive'

```
In [ ]: def fact(n):
            if n == 0:
                return 1
            else:
                return n * fact(n - 1)

        print(fact(10))
```

1.0.16 Write a function that returns the n^{th} Fibonnaci number

F(0)=1; F(1)=1; F(n)=F(n-1)+F(n-2)

1.0.17 Study the following program

```
In [ ]: import random   # module importation
        n = random.randint(1, 99)
        guess = input('Guess a number between 1 and 99 ? ')

        while guess != n:
                if guess > n:
                        print('Too large!')
                if guess < n:
                        print('Too Small!')
                guess = input('New guess?')

        print("Correct!")
```

1.0.18 Write a program where *you* think about a number and the computer tries to find it

...

1.0.19 Write a program which, given a string of '0' and '1' representing a number in binary format, prints the value of this number

...

1.0.20 Write a program which, given a number in decimal, prints its binary representation

...

1.0.21 Write a simulator of a Register Machine a la Rodrego

The input is a a list of instructions, e.g. [['inc',1,1], ['deb',1,0,3], ['end']] (no need for line number)