

basics-part1-exercises-with-solutions

October 28, 2015

1 Simple Arithmetic

Write the following expressions in python:

$$\frac{2+2}{4}$$
$$\left(\frac{5-2}{3}\right)^{25}$$
$$(-1)^{3^3} + 2((3+3) \cdot 10^4) \left(\frac{1}{6} \cdot 10^{-4}\right)$$

If it is correct, they should evaluate to 1.

```
In [10]: (2+2)/4
```

```
Out[10]: 1.0
```

```
In [12]: ((5-2)/3)**25
```

```
Out[12]: 1.0
```

```
In [13]: (-1)**(3**3) + 2*((3+3)*10**4)*((1/6) * 10**(-4))
```

```
Out[13]: 1.0
```

Write a simpler expression which gives the same result as z for arbitrary values x and y.

```
In [14]: x = 11
```

```
        y = 3
```

```
        z = (y*(x//y))%y
```

```
In [17]: #The result of the integer division by y is divisible by y, hence z is always zero.
```

```
        z = 0
```

2 Conditions: If

Use the input() function to ask a user to input a number. If the number equals “5”, output “My lucky number”. If the number is larger than 10, output “What a large number!”. In all other cases, output “That’s not my lucky number.”

```
In [21]: number = int(input("Please enter a number: "))
```

```
        if number == 5:
```

```
            print("My lucky number.")
```

```
        elif number > 10:
```

```
            print("What a large number!")
```

```
        else:
```

```
            print("That's not my lucky number.")
```

```
Please enter a number: 11
```

```
What a large number!
```

3 Conditions: while

The code below sets x to a random number between 0 and 100. Write a while-loop which repeatedly asks the user to guess x until the answer is correct. If the answer is not correct, the program tells the user whether the guess was smaller or greater than x .

```
In [22]: import random
         x = random.randint(0,100)

         while True:
             guess = int(input("guess the number: "))
             if guess == x:
                 print("Correct!")
                 break
             elif guess > x:
                 print("No, the number is smaller.")
             elif guess < x:
                 print("No, the number is larger.")
```

```
guess the number: 50
No, the number is larger.
guess the number: 60
No, the number is larger.
guess the number: 70
No, the number is larger.
guess the number: 80
No, the number is larger.
guess the number: 90
No, the number is smaller.
guess the number: 88
No, the number is smaller.
guess the number: 85
No, the number is smaller.
guess the number: 83
Correct!
```

The [Bisection method](#) is a simple way to numerically solve an equation $f(x) = 0$. The input of the algorithm is an interval $[a, b]$, where $f(a)$ and $f(b)$ have different signs, which means that the solution is somewhere in this interval. Then it computes the middle of the interval $c = \frac{a+b}{2}$. If $f(a) \cdot f(c) < 0$ set the new interval to $[a, c]$, if $f(c) \cdot f(b) < 0$ set the new interval to $[c, b]$. Write a while-loop which repeats this bisection until the error is smaller than given tolerance, e.g. $b - a < 10^{-7}$.

As an example, you could try to solve $\sin(x) = \log(x)$. The solution is somewhere between $x = 0$ and $x = 5$. The functions are available if you execute: `from math import sin, log`

```
In [40]: from math import sin, log
         tolerance = 1e-7
         a = 0.01
         b = 5
         while b-a>tolerance:
             c = 0.5*(a+b)
             if (sin(a)-log(a))*(sin(c)-log(c)) < 0:
                 b = c
             else:
                 a = c
         print("Final interval: ",a,b)
```

Final interval: 2.219107122868298 2.2191071972250924

In []:

4 For-loops

Write a for-loop which prints the items of a sequence in reversed order.

```
In [25]: seq = 'reverse me!'
         for i in range(len(seq)):
             print(seq[-i-1])
```

```
!
e
m

e
s
r
e
v
e
r
```

Write a for-loop which finds the minimum and the maximum value of a sequence of numbers.

```
In [27]: seq = [5,3,7,3,56,8,0,6,2]
         minimum = None
         maximum = None

         for s in seq:
             if minimum is None or s<minimum:
                 minimum = s
             if maximum is None or s>maximum:
                 maximum = s
         print("The minimum is: ", minimum)
         print("The maximum is: ", maximum)
```

```
The minimum is: 0
The maximum is: 56
```

Implement the [Bubble Sort](#) algorithm to sort a list of numbers. Starting from the beginning of the list, compare every adjacent pair, swap their position if they are not in the right order. After each iteration, one less element (the last one) is needed to be compared until there are no more elements left to be compared.

```
In [31]: seq = [5,3,7,3,56,8,0,6,2]

         for i in range(len(seq)):
             for j in range(len(seq)-i-1):
                 if seq[j]>seq[j+1]:
                     seq[j],seq[j+1] = seq[j+1],seq[j]
         print(seq)
```

```
[0, 2, 3, 3, 5, 6, 7, 8, 56]
```

The [Golden ratio](#) can be calculated by a continued fraction:

$$\varphi = 1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \ddots}}}} \quad (1)$$

Write a for-loop which evaluates this equation until the 1000'th fraction.

```
In [32]: value = 1
        for i in range(1000):
            value = 1/(1+value)
        print(1+value)
```

1.618033988749895

5 List slicing

Use the string given below and slices to print 'Lorem ipsum', 'elit', 'Lrmism', and 'muspi meroL'.

```
In [21]: s = "Lorem ipsum dolor sit amet, consectetur adipiscing elit."
```

```
In [35]: print(s[:11])
        print(s[-5:-1])
        print(s[:11:2])
        print(s[10::-1])
```

```
Lorem ipsum
elit
Lrmism
muspi meroL
```

```
In [ ]:
```

6 List comprehensions

Create a list $[x_0, x_1, x_2, \dots]$ where $x_n = 2^x$.

```
In [41]: print([2**x for x in range(10)])
```

[1, 2, 4, 8, 16, 32, 64, 128, 256, 512]

Create a list containing all numbers between 0 and 100 which are divisible by 3 and 4 but not divisible by 10.

```
In [42]: print([x for x in range(100) if x%3==0 and x%4==0 and not x%10==0])
```

[12, 24, 36, 48, 72, 84, 96]

Create a list $[x_0, x_1, x_2, \dots]$ where $x_n = \sum_{i=0}^n n$. Hint: Use a nested list comprehension and the sum()-function.

```
In [45]: print([sum([n2 for n2 in range(n1+1)]) for n1 in range(20)])
```

[0, 1, 3, 6, 10, 15, 21, 28, 36, 45, 55, 66, 78, 91, 105, 120, 136, 153, 171, 190]

7 Outlook

```
In [71]: import antigavity
```

```
In [ ]:
```

```
In [ ]:
```