Python Programming

Functions and strings

Mariusz Dzieńkowski

Institute of Computer Science
Lublin University of Technology

m.dzienkowski@pollub.pl

Simple built-in functions

Function	*Description*	*Example*
`abs(x)`	Returns the absolute value for x	`abs(-5)` is 5
`max(x1,x2,...)`	Returns the largest among x1, x2,...	`max(2,7,5)` is 7
`min(x1,x2,...)`	Returns the smallest among x1, x2,...	`min(2,6,3)` is 2
`pow(a,b)`	Returns a^b. Same as a**b.	`pow(2,3)` is 8
`round(x)`	Returns an integer nearest to x. If x is equally close to two integers, the even one is rounded.	`round(2.4)` is2 `round(2.5)` is 2 `round(3.5)` is 4
`round(x,n)`	Returns the float value rounded to n digits after the decimal point.	`round(4.466,2)` is 4.47 `round(4.463,2)` is 4.46

- There is not need to import any modules to use these functions.

Mathematical functions

Function	*Description*	*Example*
`fabs(x)`	Returns the absolute value for x as a float.	`fabs(-5)` is 5.0
`exp(x)`	Returns the exponential function of x (e^x).	`exp(1)` is 2.71828
`sqrt(x)`	Returns the square root of x.	`sqrt(4.0)` is 2.0
`sin(x)`, `cos(x)`, `tan(x)`	Returns the sine, cosine, tangent of x. x represents an angle in radians.	`sin(3.14159/2)` is 1 `tan(3.14159/4)` is 1

`log(x)`
`log(x,base)`
	Returns the natural logarithm of x.
	Returns the logarithm of x for the specified base.
		`log(2.71828)` is 1.0
		`log(100,10)` is 2.0
		`round(3.5)` is 4

`degrees(x)`
`radians(x)`
	Converts angle x form radians to degrees.
	Converts angle x from degrees to radians.
		`degrees(1.57)` is 90
		`radians(90)` is 1.57

`pi`, `e`	The mathematical constants pi and e.	

- The Python `math` module provides the mathematical functions listed in the table above.

Problem 1

PI computation – a program that approximates the value of pi using Leibniz' formula

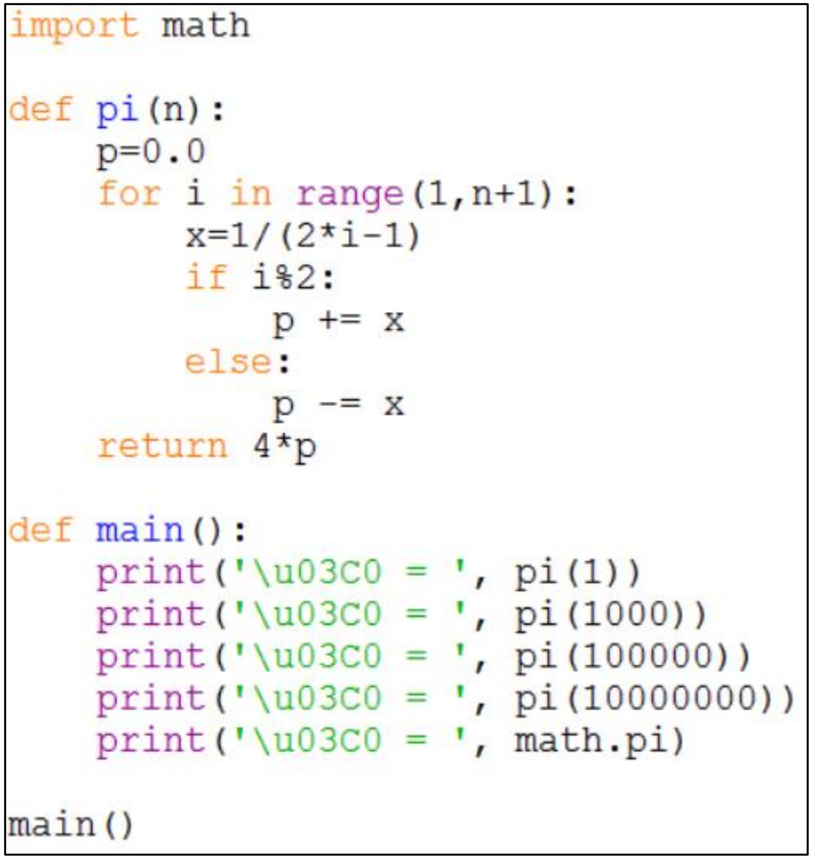
$$\pi = 4\left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \cdots + \frac{(-1)^{i+1}}{2i-1}\right)$$

- Guidelines:
  - Display the π value for i = 1, 1000, 100000, 1000000
  - To display the Greek letter π use the unicode '\u03c0'

- Curiosity:
  - The word "welcome" is translated into Chinese using two characters: 欢 and 迎 The Unicode representations of these two characters are: '\u6b22\u8fce'

Problem 1 - solution

- Source code



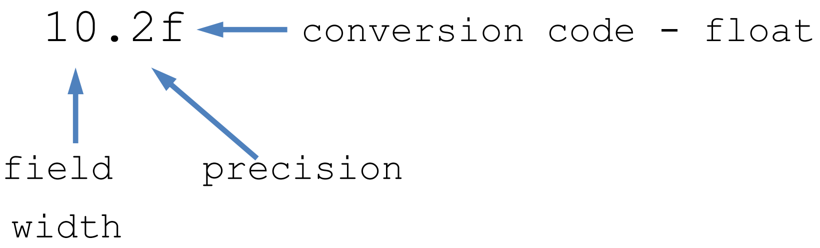
```python
import math

def pi(n):
    p=0.0
    for i in range(1,n+1):
        x=1/(2*i-1)
        if i%2:
            p += x
        else:
            p -= x
    return 4*p

def main():
    print('\u03C0 = ', pi(1))
    print('\u03C0 = ', pi(1000))
    print('\u03C0 = ', pi(100000))
    print('\u03C0 = ', pi(10000000))
    print('\u03C0 = ', math.pi)

main()
```

Problem 2

E number computation – a program that approximates the value of e (mathematical constant) that is the base of the natural logarithm

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \cdots + \frac{1}{i!}$$

- Guidelines:
  - Define functions: `factorial(n),e(n), main()`
  - Display the e value for i = 1 .. 20
  - For float value use the specifier to give the width and precision of the format in the form of `width.precisionf`:



10.2f

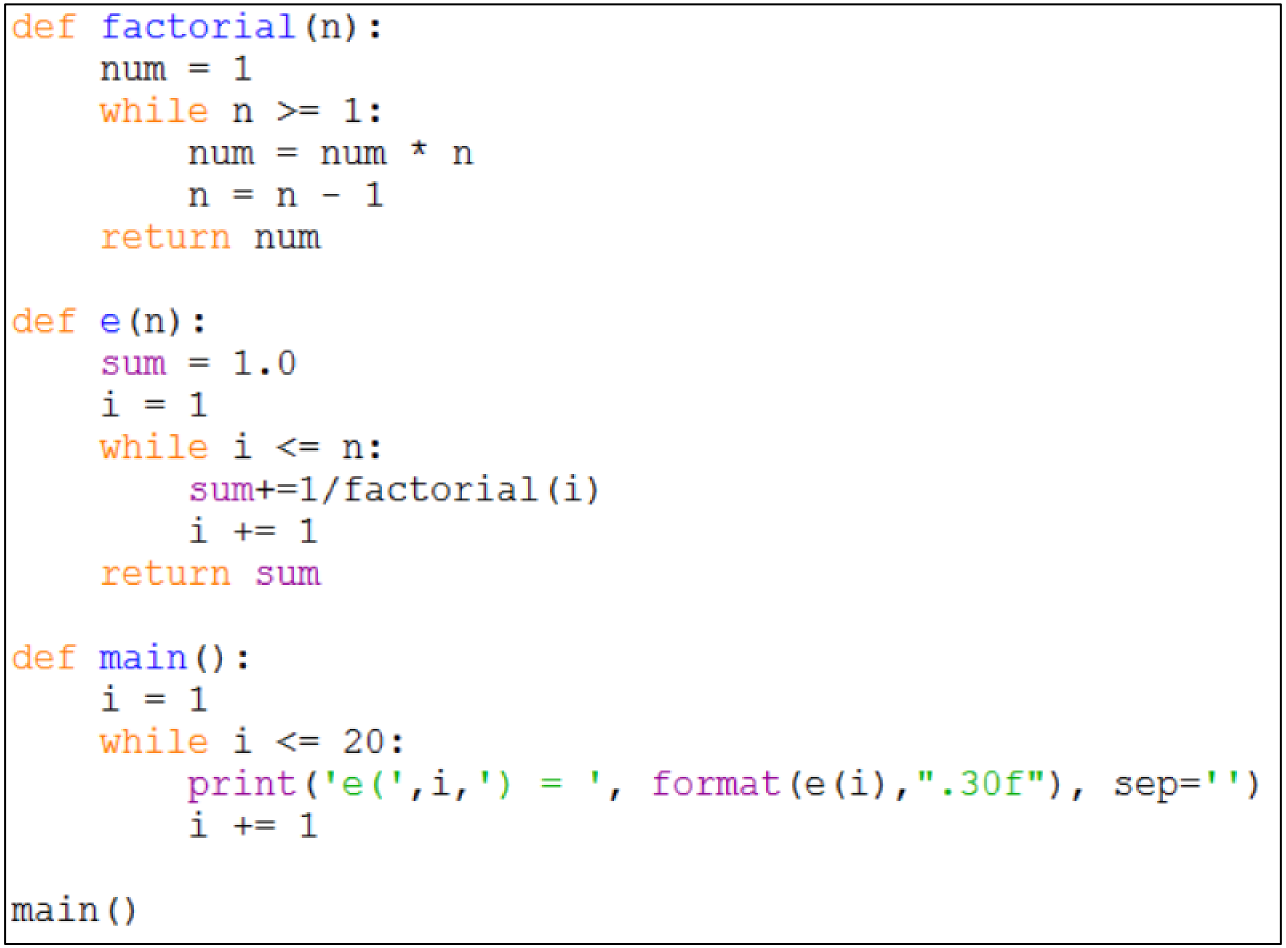
conversion code - float

field width

precision

Problem 2 - solution

- Source code



```
def factorial(n):
    num = 1
    while n >= 1:
        num = num * n
        n = n - 1
    return num

def e(n):
    sum = 1.0
    i = 1
    while i <= n:
        sum+=1/factorial(i)
        i += 1
    return sum

def main():
    i = 1
    while i <= 20:
        print('e(',i,') = ', format(e(i),".30f"), sep='')
        i += 1

main()
```

Problem 3

Mean and standard deviation computation – a program that prompts the user to enter the numbers, and displays the mean and standard deviations of these numbers

$$mean = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{x_1+x_2+\cdots+x_n}{n} \qquad deviation = \sqrt{\frac{\sum_{i=1}^{n} x_i^2 - \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n}}{n-1}}$$

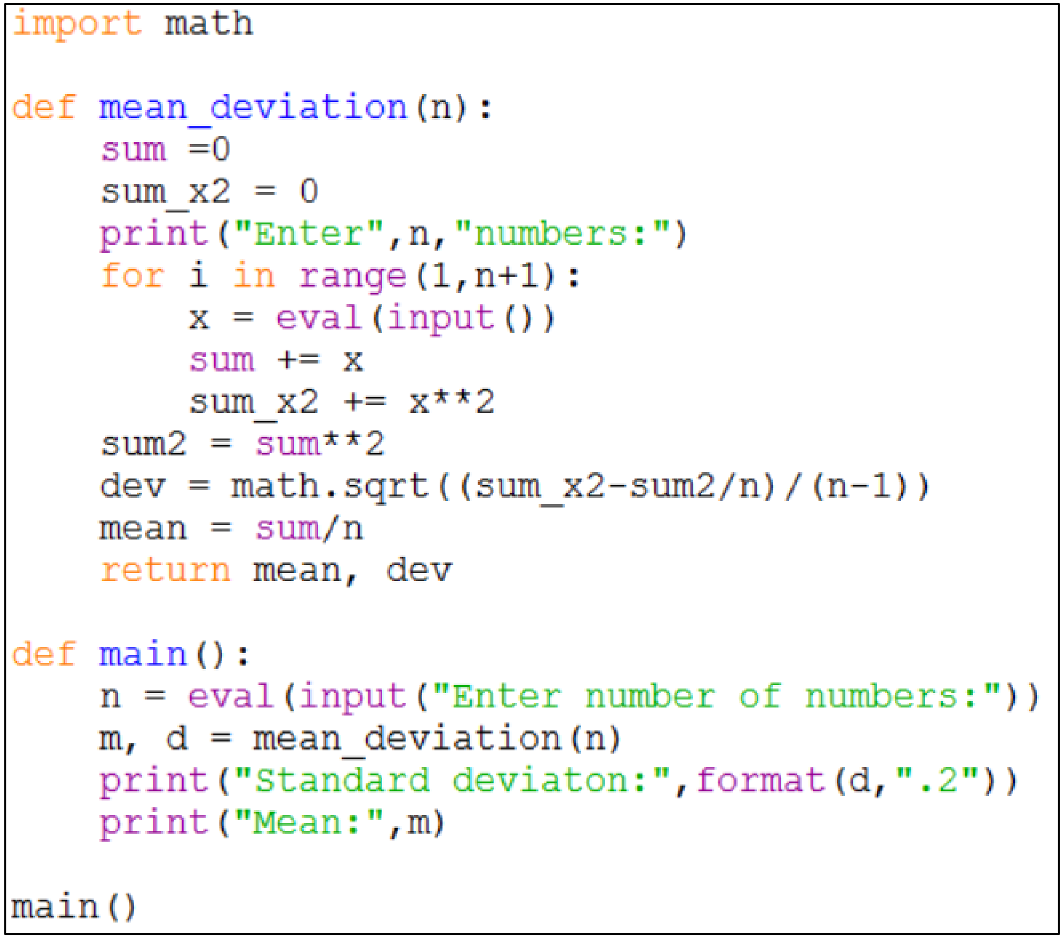
- Guidelines:
  - The mean is the average of the numbers.
  - The standard deviation is a statistic that tells you how tightly all the various data are clustered around the mean in a set of data.
  - A program should contain the following functions:

`mean_deviation(n)` – prompt the user to enter a list of n numbers;
compute and return: the mean and the standard deviation

`main()` – enter the n value, call the mean_deviation function and display results

Problem 3 - solution

- Source code



```
import math

def mean_deviation(n):
    sum =0
    sum_x2 = 0
    print("Enter",n,"numbers:")
    for i in range(1,n+1):
        x = eval(input())
        sum += x
        sum_x2 += x**2
    sum2 = sum**2
    dev = math.sqrt((sum_x2-sum2/n)/(n-1))
    mean = sum/n
    return mean, dev

def main():
    n = eval(input("Enter number of numbers:"))
    m, d = mean_deviation(n)
    print("Standard deviaton:",format(d,".2"))
    print("Mean:",m)

main()
```