python code kata: Expressions and arithmetic operators

Hello 👋

Great to see you again 👍. This week we are going to practice following elements of Python in our code kata 👇

Kata this week

This week we are going to practice with various numeric types, arithmetic operators, relational operators and conditional programming in Python.

Numeric types in Python

Python defines the following numeric literals:-

Number	Type	Comment
10	Integer	Numeric values without fractional part are integers.
-10	Integer	Integers can be positive or negative.
0.5	Floating Point	Numeric value with the fractional part is floating point. These can be single precision or double precision values.
2.96E-2	Floating Point	Floating point values in exponential notation. 2.96 / 100 = 0.0296

Numeric literals can be written in decimal, hexadecimal and octal notation.

Arithmetic operators in Python

Python defines following arithmetic operators:-

Operator	Usage	Example	Return Value
+	Represent a positive numerical value.	+2	2
	Adds two numerical values.	5 + 2, 12.4 + 45.89	7, 58.29
-	Represent a negative numerical value.	-89	-89
	Subtract two numerical values.	13 - 5, 62.4 - 1.56	8, 60.84
*	Multiplies two numerical values	45 * 23	1035
/	Division between two numerical values	5 / 2	2.5
//	Floor division between two numerical values	5 / 2	2
%	Reminder of a floor divison	5 % 2	1
**	Exponential operator denoting power operation	5 ** 2	25

Output of arithmetic operators is always a numerical value.

Ground Statement

`Ground Statement` are small programming exercises for you to play ⚾ with:

Statement 1

Write following mathematical statmenets in Python

1. $$\frac{(x+y)}{2}$$
2. $$\frac{xy}{2}$$

3. $$(1 + \frac{r}{100})^n$$

4. $$\sqrt{a^2 + b^2}$$

Statement 2

Write a program that reads a number and displays the square, cube, and fourth power. Try to solve it in as many ways as possible.

Statement 3

Write a program that prompts the user for two integers and then prints:
• The sum
• The difference
• The product
• The average
• The distance (absolute value of the difference)
• The maximum (the larger of the two)
• The minimum (the smaller of the two)

Statement 4

Write a program that reads a number between 1,000 and 999,999 from the user and prints it with a comma separating the thousands.

```
Please enter an integer between 1000 and 999999: 23456
23,456
```

Statement 5

Accept an floating value and round it to nearest smaller integer

```
Please enter a floating value : 2.5
2
```

Statement 6

Simulate a postage stamp vending machine. A customer inserts dollar bills into the vending machine and then pushes a “purchase” button. The vending machine
gives out as many first-class stamps as the customer paid for, and returns the change in penny (one-cent) stamps. A first-class stamp cost 44 cents

```
Enter number of dollars: 4
First class stamps:9
Penny stamps:4
```

Hints if you are stuck:-

- How do I will accept values for keyboard
- Some utility functions to check between maximum and minimum
- A list of all python builtin functions

Sensai Says

"What you learn is not what you read or listened to, but rather what you attempted at…"
Progressive learning

If you feel the exercise a little bit difficult to solve do not get disheartened. The whole idea behind these programming exercises is not to solve them but rather attempt them.

Try to attempt them in as many ways as possible, you will learn new techniques that will be very helpful to you in the long run especially in the work field.

We will present you with **solution mail/document** also. The solution will show you various ways to solve a problem and why a technique is better than the last one.

We encourage you to make notes from the solution provided and try to apply what you have learned in future exercise.

What I will gain from these exercises ?

1. A better and faster way to solve exercise.
2. Reusable components like containers, algorithms, etc. that you can apply to the problem at hand rather than designing your own.
3. Confidence and attitude to solve a problem in new ways, instead of trying monotonous techniques.
4. Writing the robust and quality software using **test driven development**.

I need Help 🆘

Still, have some questions related to this exercise. If you still have questions feel free to connect. We will be happy to answer your questions.