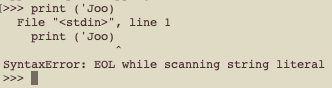
Part 1:

* If you are trying to print a string, what happens if you leave out one of the quotation marks or both and why?

If I leave one of the quotation marks like the following example:

>>> print ('Joo)



The program prints an error because the quotation is part of the syntax, I think this problem happens because the program didn’t know where is the end of the words I want to print. I think we use the quotation to tell the compiler here are the beginning and the end of the words I want to print.

* You can use a minus sign to make a negative number like -2. What happens for each of the following and why?

If I use two plus signs like the following example:

>>> 2++2



Like in math the program will deal with the operator as the following:

The program will consider the first ‘+’ as a sign for making the addition and the second ‘+’ as a sign for saying that the second number is a positive number, then the program will make the addition so the answer will be 4.

If I use two minus signs like the following example:

>>> 2—2



Like in math the program will deal with the operator as the following:

The program will consider the first ‘-’ as a sign for making the subtraction and the second ‘-’ as a sign for saying that the second number is a negative number, then the program will make the subtraction so the answer will be 4.

If I use plus then minus signs like the following example:

>>> 2+-2



Like in math the program will deal with the operator as the following:

The program will consider the first ‘+’ as a sign for making the addition and the second ‘-’ as a sign for saying that the second number is a negative number, then the program will make the addition so the answer will be 0.

If I use minus then plus signs like the following example:

>>> 2-+2



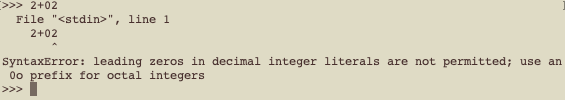
Like in math the program will deal with the operator as the following:

The program will consider the first ‘-’ as a sign for making the subtraction and the second ‘+’ as a sign for saying that the second number is a positive number, then the program will make the subtraction so the answer will be 0.

* In math notation, leading zeros are OK, as in 02. What happens if you try this in Python? Why?

If I put 0 before a number like the following example:

>>> 2+02

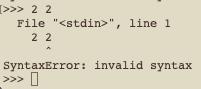


An error will appear because python isn’t like our traditional math it has its syntax that doesn’t allow us to put 0 before an integer.

* What happens if you have two values with no operator and a space between them? Why?

If I put two numbers with space between them without an operator like the following example:

>>> 2 2



An error will appear as we say in the previous example python has its syntax that he understands, and he didn’t know what to do with the numbers so he prints an error.

#### Part 2:

#### What happens if I divided by 0, how python will deal with that?

#### >>> 1/0

#### 

#### It prints an error like every calculator does and as I expected.

#### What I learn is that machines can not deal with division by zero like humans, so I should avoid any mistake of this kind so I didn’t get errors.

#### What happens if I put a math operator in the print statement?

#### >>> print('2\*2')

#### 

#### The program prints the operation as it is without operating it.

#### What I learn from this experiment is that python deal with anything inside the quotation marks of the print statement as its and doesn’t process it or do anything with it, so if I want to do any operations I should do it outside the print statement first and then print it.

#### What happens if I divided very big numbers is python would give me the exact number or he will give us the nearest?

#### >>> 123456789123456789/123456789123456789

#### >>> 123456789123456789/123456789123456780

#### 

#### At this time, I give the computer two operators to see how it will behave, In the first one I divided a very big number by itself and it correctly divides it and gave us the result ‘1’, but in the second one I gave it the exact same numbers but I changed the last digit in the second number so the result of the operator doesn’t be whole ‘1’ number, but the computer here gave us ‘1’.

#### What I learn from this experiment is python gives the nearest number, so if I want to do complicated math operations and want accurate results python will fail me in that.

#### In the end, I really enjoyed having this experience and dealing with python and the whole things I have learned so far.