Data manipulation tools

Python as a calculator

print(2+2)

Output is: 2

Basic operators

Operator

Is symbol of the programming language, which is able to operate on the values.

the + (plus) sign is the operator which is able to add two numbers, giving the result of the addition.

Others can be: **+**

**-**

**\*** is a **multiplication** operator

**/** is a **division** operator

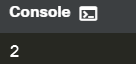
**//** is a **intiger division** operator  
 The result of integer division is always rounded to the nearest integer value that is less than the real

**If it’s a - number it will be - number higher**



**%**

**The result of the operator is a remainder left after the integer division**



**\*\***

**The result produced by the division operator is always a float**

**Is this a problem? Yes, it is. It happens sometimes that you really need a division that provides an integer value, not a float.**

How not to divide

As you probably know, **division by zero doesn't work**.

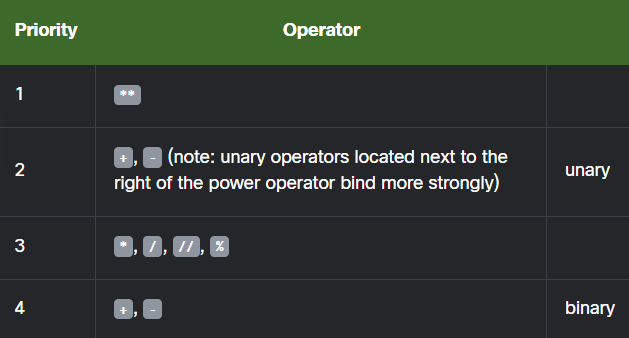
Do not try to:

perform a division by zero;

perform an integer division by zero;

find a remainder of a division by zero.

List of priorities



Operators and parentheses

Of course, you're always allowed to use **parentheses ‘()’**, which can change the natural order of a calculation.

**SECTION SUMMARY**

Expression

Is a combination of values which evaluates to a certain value, e.g., 1 + 2.

Operators

special symbols or keywords which are able to operate on the values and perform (mathematical) operations, e.g., the \* operator multiplies two values: x \* y.

Arithmetic operators in Python:

+ (**addition**),

- (**subtraction**),

\* (**multiplication**),

/ (**classic division ‒ always returns a float**),

% (**modulus ‒ divides left operand by right operand and returns the remainder of the operation**, e.g., 5 % 2 = 1),

\*\* (**exponentiation** ‒ **left operand raised to the power of right operand**, e.g., 2 \*\* 3 = 2 \* 2 \* 2 = 8),

// (**floor/integer division** ‒ **returns a number resulting from division, but rounded down to the nearest whole number**, e.g., 3 // 2.0 = 1.0)

Unary operator

is an **operator with only one operand**, e.g., -1, or +3.

Binary operator

**operator with two operands**, e.g., 4 + 5, or 12 % 5.

the hierarchy of priorities:

**the \*\* operator** (exponentiation) has the highest priority;

**then the unary + and -** (note: a unary operator to the right of the exponentiation operator binds more strongly, for example 4 \*\* -1 equals 0.25)

**then: \*, /, and %,**

**and finally, the lowest priority: binary + and -**.

Subexpressions in parentheses

**are always calculated first**, e.g., 15 - 1 \* (5 \* (1 + 2)) = 0.

**The exponentiation operator uses right-sided binding**, e.g., 2 \*\* 2 \*\* 3 = 256.