Python Variables and Operators

FPT Education
FPT UNIVERSITY

- 0. Your first program
- 1. Number and Boolean
- 2. String and Date
- 3. Decimal, Fraction, Statistics and Math Module
- 4. Arithmetic and Comparison Operators
- 5. Problem Solving : Stock Market Analysis

Your first program

FPT Education
FPT UNIVERSITY

- The print() function

```
print("Hello, World!")
```

- A function (in this context) is a separate part of the computer code able to:
  - Cause some effect (e.g., send text to the terminal, create a file
  - Evaluate a value (e.g., the square root of a value or the length of a given text) and return it as the function's result;
- Where do the functions come from?
  - From Python itself; (it is built-in); modules; write them yourself,

•the word print;
•an opening parenthesis;
•a quotation mark;
•a line of text: Hello, World!;
•another quotation mark;
•a closing parenthesis.

Your first program

- What happens when Python encounters an invocation like this one below?
  - `function_name(argument)`
    - First, Python checks if the name specified is legal ;
    - Second, Python checks if the function's requirements for the number of arguments allows you to invoke the function in this way;
    - Third, Python leaves your code for a moment and jumps into the function you want to invoke; of course, it takes your argument(s) too and passes it/them to the function;
    - Fourth, the function executes its code, causes the desired effect (if any), evaluates the desired result(s) (if any) and finishes its task;
    - Finally, Python returns to your code (to the place just after the invocation) and resumes its execution.

Your first program

- The print() function - the escape and newline characters
  - The backslash (\) has a very special meaning when used inside strings - this is called the escape character \n.
  - The print() function - using multiple arguments
  - Passing the arguments into the print() function is the most common in Python, and is called the positional way
  - The print() function - the keyword arguments

```
print("My", "name", "is", "Monty", "Python.", sep="-")
print("My", "name", "is", sep="_", end="*")
print("Monty", "Python.", sep="*", end="*\n")
```

Your first program

- The input() function comes with an optional parameter: the prompt string. It allows you to write a message before the user input, e.g.:
  - name = input("Enter your name: ")
  - print("Hello, " + name + ". Nice to meet you!")
- When the input() function is called, the program's flow is stopped, the prompt symbol keeps blinking (it prompts the user to take action when the console is switched to input mode) until the user has entered an input and/or pressed the Enter key.
- The result of the input() function is a string. You can add strings to each other using the concatenation (+) operator (* - replication)

```
num_1 = input("Enter the first number: ") # Enter 12
num_2 = input("Enter the second number: ") # Enter 21

print(num_1 + num_2) # the program returns 1221
```

Variables

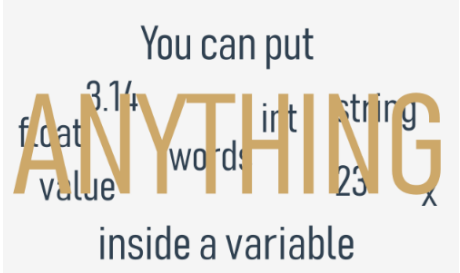
- How to store the results of these operations, in order to use them in other operations
  - It offers special "boxes" (containers) for that purpose, and these boxes are called variables
  - What does every Python variable have?
    - a name;
    - a value (the content of the container)

```
Adiós_Señora, sûr_la_mer, Einbahnstraße, переменная.
```

- Keywords
  - ['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']

Variables

- What can you put inside a variable?



- A variable comes into existence as a result of assigning a value to it. Unlike in other languages, you don't need to declare it in any special way.
- If you assign any value to a nonexistent variable, the variable will be automatically created. You don't need to do anything else.
- The creation (or otherwise - its syntax) is extremely simple: just use the name of the desired variable, then the equal sign (=) and the value you want to put into the variable.

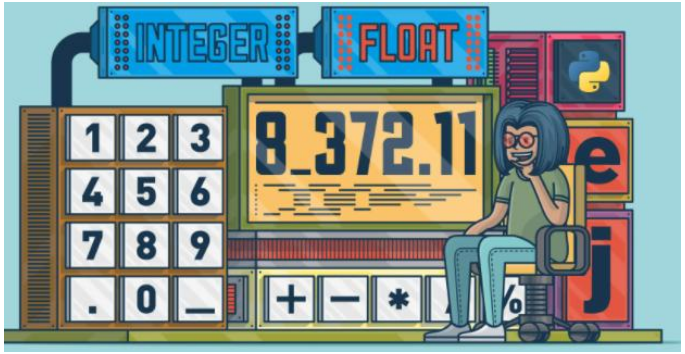
Variables

- *A Python variable is a reserved memory location to store values. In other words, a variable in a python program gives data to the computer for processing.*
- *Every value in Python has a datatype. Different data types in Python are Numbers, List, Tuple, Strings, Dictionary, etc. Variables can be declared by any name or even alphabets like a, aa, abc, etc.*
- *Example :*

```
a=100
print (a)
```

Number and Boolean

- Number
  - Integer : In Python, value of an integer is not restricted by the number of bits and can expand to the limit of the available memory
  - Floating Number: In Python, float is used to represent real numbers and is written with a decimal point dividing the integer and fractional parts



FPT Education
FPT UNIVERSITY

- Number
- Complex Number: A complex number is a number that can be expressed in the form a + bi, where a and b are real numbers, and i represents the imaginary unit, satisfying the equation i^2 = –1

Convert Complex Number from Rectangular Form to Polar (Trigonometric) Form

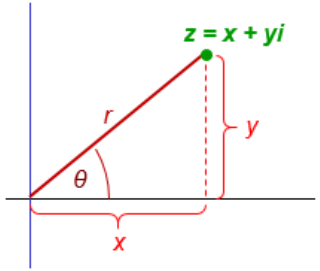
$z = x + yi$ (rectangular form)

$r = |z| = \sqrt{x^2 + y^2}$

$x = r\cos\theta$

$y = r\sin\theta$

$z = r(\cos\theta + i\sin\theta)$ (polar form)



Number and Boolean

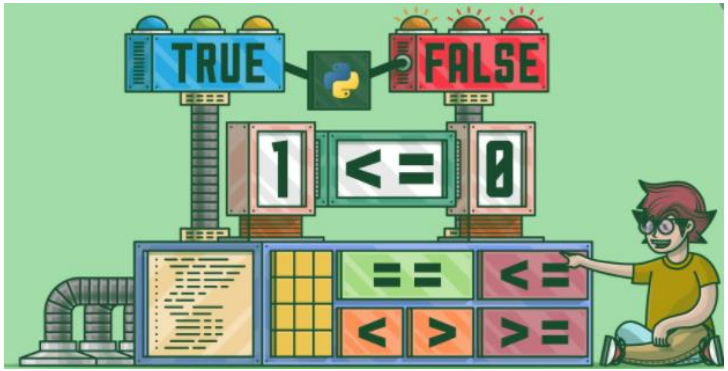
- Questions and answers
  - A programmer writes a program and the program asks questions.
  - A computer executes the program and provides the answers.
  - The program must be able to react according to the received answers.
- To ask questions, Python uses a set of very special operators.
  - Equal, less than or equal to, greater than or equal to, greater than

Boolean

- Boolean
- Booleans represent one of two values: True or False

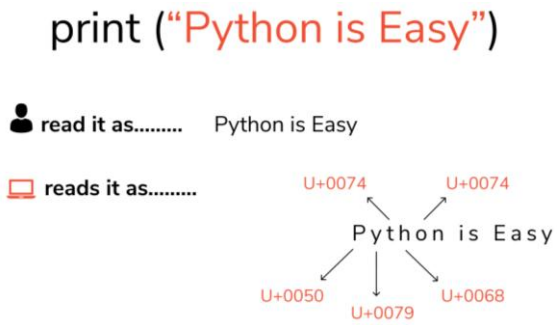
```
>>> a = 10
>>> b = 10
>>> c = 20
>>>
>>> a == b
True
>>> a != b
False
>>> c > a
True
>>> c < a
False
>>> c <= 20
True
>>> c >= 20
True
```

A	B	A AND B	A OR B	NOT A
False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False



String and Date

- String
  - Strings are Arrays. Like many other popular programming languages, strings in Python are arrays of bytes representing unicode characters
- Datetime
  - The datetime module supplies classes for manipulating dates and times.




```
date_string = "21 June, 2018"

... .. ...

date_object = datetime.strptime(date_string, "%d %B, %Y")
```

3. Decimal, Fraction, Statistics and Math Module

FPT Education
FPT UNIVERSITY

Decimal Module

- The decimal module provides support for fast correctly-rounded decimal floating point arithmetic.

Fractions Module

- A Fraction instance can be constructed from a pair of integers, from another rational number, or from a string. Fraction instances are hashable, and should be treated as immutable.

3. Decimal, Fraction, Statistics and Math Module

FPT Education
FPT UNIVERSITY

Math Module

- The Python math module is an important feature designed to deal with mathematical operations. It comes packaged with the standard Python release and has been there from the beginning

Statistics Module

- The statistics module provides functions to mathematical statistics of numeric data. The following popular statistical functions are defined in this module.

FPT Education
FPT UNIVERSITY

Math Module

Function	Returns (Description)
abs(x)	The absolute value of x: the (positive) distance between x and zero.
ceil(x)	The ceiling of x: the smallest integer not less than x. **math module is required.**
cmp(x, y)	-1 if x < y, 0 if x == y, or 1 if x > y. **Deprecated** in Python 3; Instead use **return (x>y)-(x<y).**
exp(x)	The exponential of x: e^x **math module is required.**
fabs(x)	The absolute value of x. **math module is required.**
floor(x)	The floor of x: the largest integer not greater than x. **math module is required.**
log(x)	The natural logarithm of x, for x> 0. **math module is required.**
log10(x)	The base-10 logarithm of x for x> 0. **math module is required.**

max(x1, x2,...)	The largest of its arguments: the value closest to positive infinity.
min(x1, x2,...)	The smallest of its arguments: the value closest to negative infinity.
modf(x)	The fractional and integer parts of x in a two-item tuple. Both parts have the same sign as x. The integer part is returned as a float. **math module is required.**
pow(x, y)	The value of x**y.
round(x [,n])	x rounded to n digits from the decimal point. Python rounds away from zero as a tie-breaker: round(0.5) is 1.0 and round(-0.5) is -1.0.
sqrt(x)	The square root of x for x > 0. **math module is required.**

Operators and their priorities

- The phenomenon that causes some operators to act before others is known as the hierarchy of priorities.
- Operators and their bindings
  - The binding of the operator determines the order of computations performed by some operators with equal priority, put side by side in one expression.
- Most of Python's operators have left-sided binding, which means that the calculation of the expression is conducted from left to right.

```
print(9 % 6 % 2)
```

- Operators and their bindings: exponentiation

```
print(2 ** 2 ** 3)
```

Operators and their priorities

- A unary operator is an operator with only one operand,
- A binary operator is an operator with two operands
- Subexpressions in parentheses are always calculated firs

Priority	Operator	
1	`**`	
2	`+`, `-` (note: unary operators located next to the right of the power operator bind more strongly)	unary
3	`*`, `/`, `//`, `%`	
4	`+`, `-`	binary

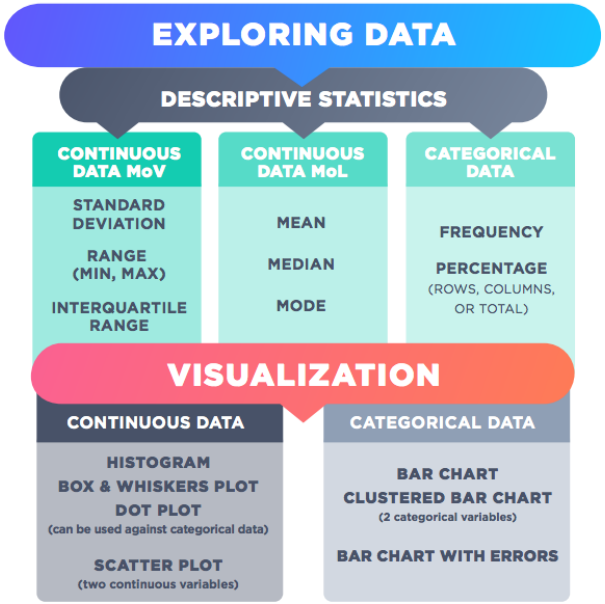
```
print(2 * 3 % 5)
```

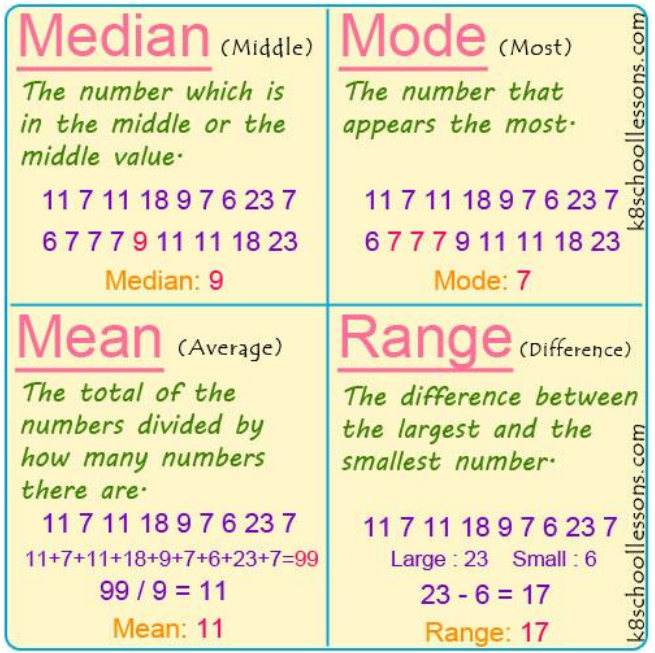
Shortcut operators

```
i = i + 2 * j  ⇒  i += 2 * j

var = var / 2  ⇒  var /= 2

rem = rem % 10  ⇒  rem %= 10

j = j - (i + var + rem)  ⇒  j -= (i + var + rem)

x = x ** 2  ⇒  x **= 2
```

FPT Education
FPT UNIVERSITY

Statistics Module



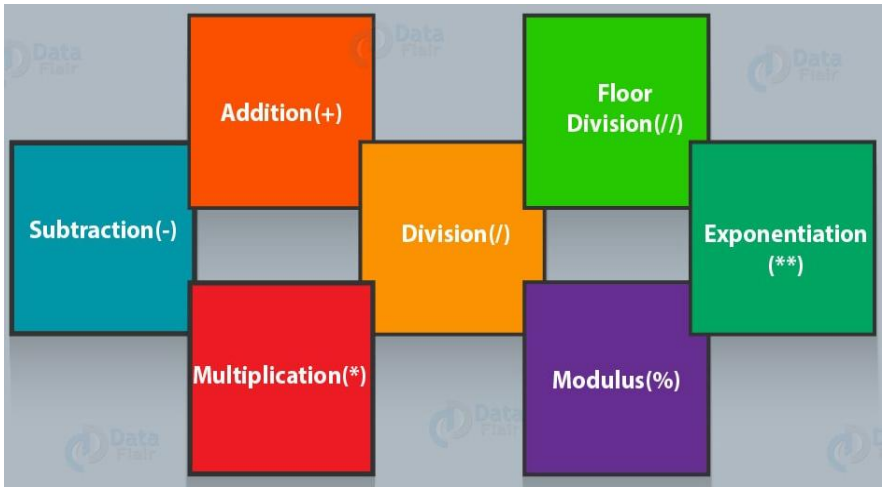


4. Arithmetic and Comparison Operators

FPT Education
FPT UNIVERSITY

Arithmetic and Comparison Operators

- Arithmetic Operators are used to perform mathematical calculations. ... Comparison Operators are used to perform comparisons.

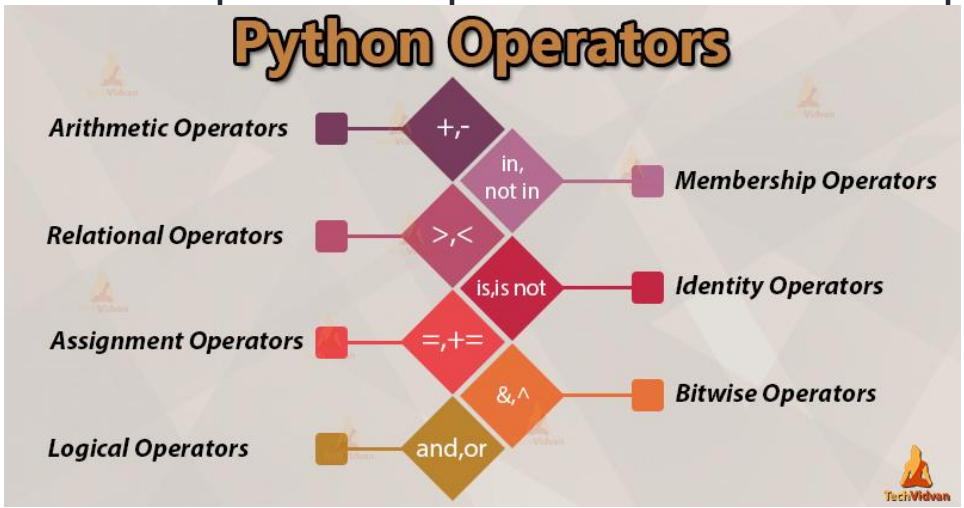


4. Arithmetic and Comparison Operators

FPT Education
FPT UNIVERSITY

Arithmetic and Comparison Operators

- Arithmetic Operators are used to perform mathematical calculations. ... Comparison Operators are used to perform comparisons.



in and not in operators

- Python offers two very powerful operators, able to look through the list in order to check whether a specific value is stored inside the list or not.
  - The first of them (in) checks if a given element (its left argument) is currently stored somewhere inside the list (the right argument) - the operator returns True in this case.
  - The second (not in) checks if a given element (its left argument) is absent in a list - the operator returns True in this case.

```
my_list = [0, 3, 12, 8, 2]

print(5 in my_list)
print(5 not in my_list)
print(12 in my_list)
```

5. Problem Solving : Stock Market Analysis

- In stock trading, the high and low refer to the maximum and minimum prices in a given time period. Open and close are the prices at which a stock began and ended trading in the same period. Volume is the total amount of trading activity. Adjusted values factor in corporate actions such as dividends, stock splits, and new share issuance. (OHLCV)