Python Programming

Introduction

Things To Know

- Python is one of the most popular Powerful Programming language. It is created by ***Guido van Rossum*** in 1991.
- It is the most demanding programming language in the USA job market with the highest 74 K job posting in January 2021. Also, Python ranked third with a $120 K yearly salary.

Syntax

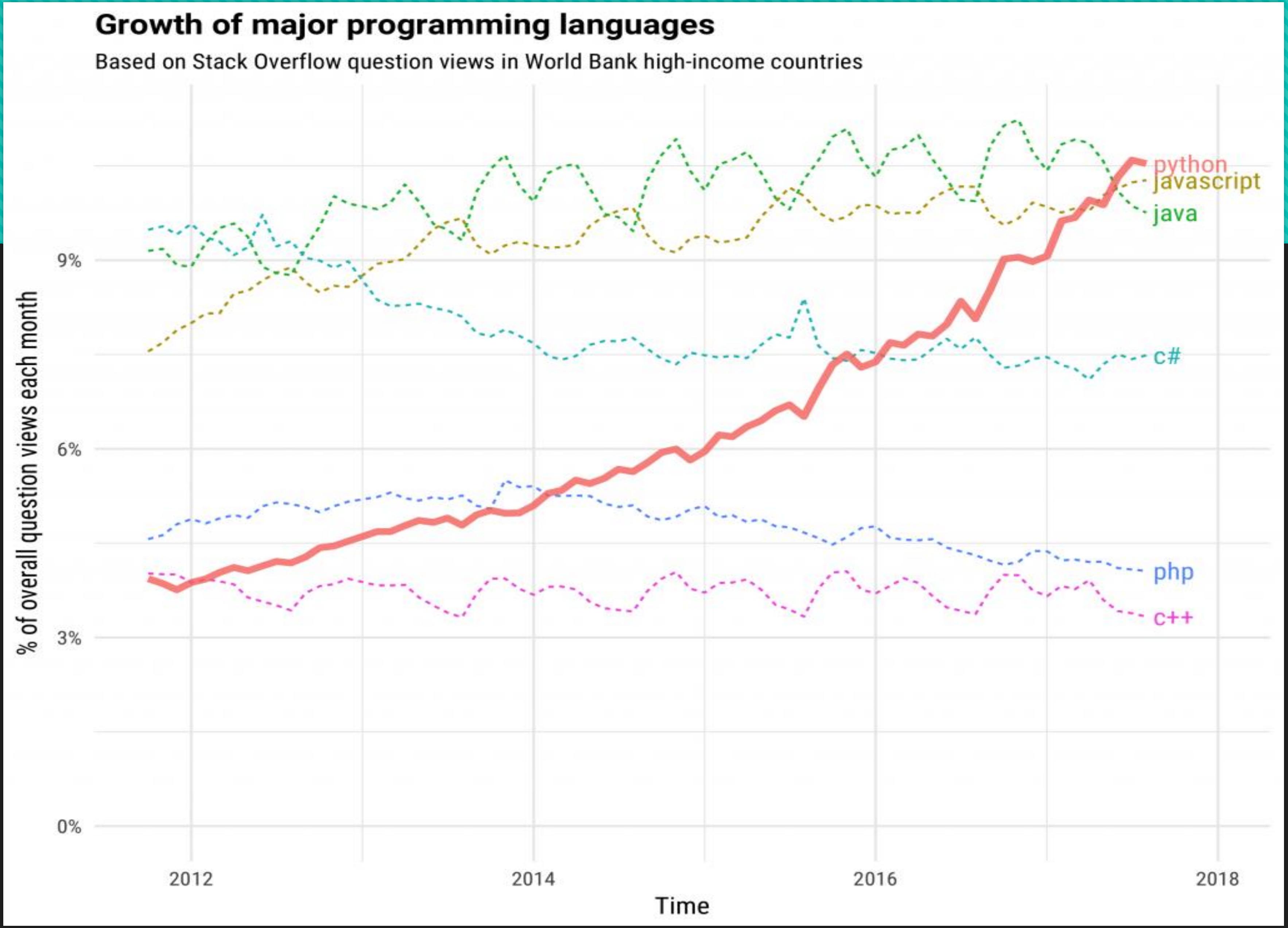
- It is an interpreted, high-level and general-purpose programming language.
- It has a very **simple** and **easy** syntax which makes it easy to learn and understand.
- Documentation of python: docs.python.org

Applications

- 1.Software Development
- 2. Web Development
- 3.Game Development
- 4. Artificial Intelligence and Machine Learning
- 5. Data Science
- And a lot more

Why to learn Python?

- Python is an most ***popular*** programming language right now.
- It is very **easy** to learn and understand.
- You will be paid really a ***good salary*** if you have **good skills**.
- It supports both **object-oriented programming** and **structured programming**.
- It offers a lot of functionalities using standard **libraries** which allows the easy implementation of ***complex applications***.
- A lot of cross-language operations can be performed easily on Python because of its **portable** and **extensible** nature.
- It has a **large community** that is particularly helpful for new Python programmers.
- Python can be used to build **graphical user interfaces** or **desktop applications**.
- Python is used in **Scripting** and **Automation**.


Growth of major programming languages
Based on Stack Overflow question views in World Bank high-income countries
% of overall question views each month
9%
6%
3%
0%
2012
2014
2016
2018
Time
python
javascript
java
c#
php
c++

Requirements to Learn Python

- Install **Python 3** into your PC.
- **IDE** for coding python language in a better way.
- 4-6 GB RAM is needed for smooth functioning.
- Windows 8 or 10 is recommended, so that we can Code without any System Errors / Problems.

Print() statement

- The print() function prints the given object to the standard output device (screen) or to the text stream file.
- **Parameters** : objects, sep, end, file and flush.
- Simply, print() will create a new line without any parameters.
- Syntax : print(*objects, sep = ‘ ’, end = ‘ ’, file = sys.stdout, flush= False)
- Visit for more reference: https://www.programiz.com/python-programming/methods/built-in/print

Escape Sequences - \

Code	*Result*
\’	‘
\"	“
\\	\
\n	New Line
\r	Carriage Return
\t	Tab

Comments In Python

- Any sentence or word or letter or blank_line with a **#** at the beginning is called a comment.
- Comments are **ignored** by Python, these are used to make code readable and understandable to the reader.
- ***Multi-line Comments*** : Anything with in ‘’’   ‘’’   is a multi-line comment or a **Doc-string**.

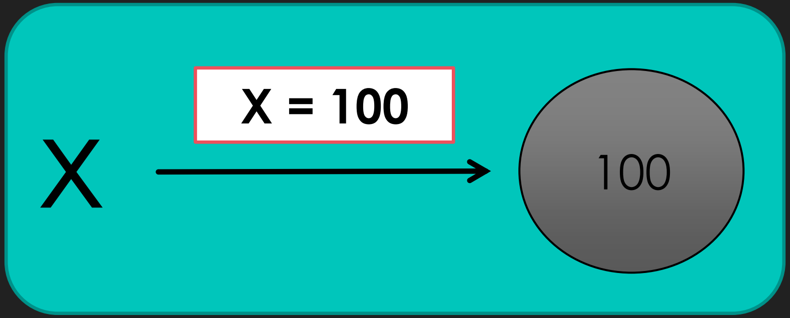
Data Types In Python

- *Numbers* – *Integers, Floating-point numbers, Complex numbers*
- *Strings* – Anything inside ‘ ’ or “ ” is a string.
- *Lists* – Any element/s inside [ ] is a list.
- *Tuples* – Any element/s inside ( ) is a tuple.
- *Dictionaries* – Any element/s inside { } is a dictionary. It is a collection of keys : values.
- *Sets* – It is similar to dictionaries, but they don’t contain any values.

Variables In Python

- Variables are containers for storing data values.
- Rules for declaring a variable name are :

1. Variable names are case-sensitive.
2. A **variable name** must start with a letter or the underscore character.
3. A **variable name** cannot start with a number but numeric values can be used in the middle or last.



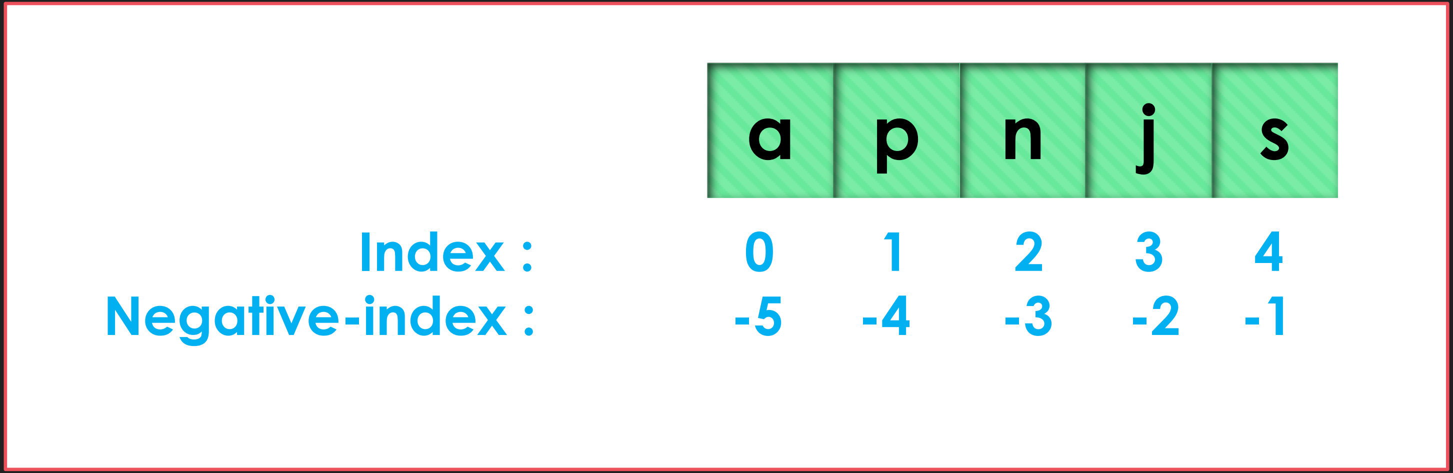
Mutable and Immutable

- Mutable means which can be changed.
- Numbers, Lists, Dictionaries, Sets are mutable.
- Immutable means which can not be changed.
- Strings and Tuples are immutable.

Indentation

- **Four whitespaces** are used for indentation and are preferred over **tabs**.
- If you **don’t give** proper indentations then, you will get an **error**.

Indexing



Slicing

- Syntax : variable_name(start, end, step-value)

Start	Optional. An integer number specifying at which position to start the slicing. Default is 0.
End	An integer number specifying at which position to end the slicing.
Step-Value	Optional. An integer number specifying the step of the slicing. Default is 1.

True and False

- True is equal to 1.
- False is equal to 0.

Operators

- Arithmetic operators
- Assignment operators
- Comparison operators
- Logical operators
- Identity operators
- Membership operators
- Bitwise operators

1. Arithmetic operators

Operators	Name
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus
**	Exponentiation
//	Floor division

2. Assignment operators

Operators
=
+=
-=
*=
/=
%=
//=
**=
&=
!=
^=
>>=
<<=

3. Comparison operators

Operators	Name
==	Equal
!=	Not Equal
>	Greater Than
<	Less Than
>=	Greater Than Equal to
<=	Less Than Equal to

4. Logical operators

Operators	Desc.
and	Returns True if both statements are true
or	Returns True if one of the statements is true
not	Reverse the result, returns False if the result is true

5. Identity operators

Operators	Desc.
is	Returns True if both variables are the same object
is not	Returns True if both variables are not the same object

6. Membership Operators

Operators	Desc.
in	Returns True if a sequence with the specified value is present in the object
not in	Returns True if a sequence with the specified value is not present in the object

7. Bitwise operators

Operators	Name
&	AND
|	OR
^	XOR
~	NOT
<<	Zero fill left shift
>>	Signed right shift

Conditional Statement – if, elif, else

- These are used when you have **different choices** but want execute **one which satisfies the condition**.
- **IF the ‘ if ’ condition is True then it will be executed but if the ‘ if ‘ statement is False then ‘ elif ’ statement will be executed and if both are False then ‘ else ’ statement will be executed.**

For Loop

- A `for` loop is used for ***iterating*** over a sequence (that is either a list, a tuple, a dictionary, a set, or a string).
- **Iterate : perform again**

While Loop

- As long as the given condition is **True** , the while loop will continue to execute the statement , but when the condition get **False** it breaks and the program stops.

Continue and Break Statements

- The **break** statement in Python terminates the current loop and resumes execution at the next statement.
- The **continue** statement in Python returns the control to the beginning of the while loop. And it forgets / rejects the conditions or statements below it.

Functions

- **Functions**: A **function** is a block of code which only runs when it is called.
- You can define a function by using **def <name_of_function><parameters>:**
- A **parameter** is the variable listed inside the parentheses in the function definition.
- Calling a function - **<name_of_function><arguments>**
- An **argument** is the value that is sent to the function when it is called.

Try and Except (Exception Handling)

- If you have some *suspicious* code that may raise an exception, you can defend your program by placing the suspicious code in a **try:** block.
- After the try: block, include an **except:** statement, followed by a block of code which handles the problem as elegantly as possible.
- The **finally** block is a place to put any code that must execute, whether the try-block raised an exception or not.

Best websites to go for doc:

- https://www.w3schools.com/python/
- https://www.tutorialspoint.com/python/

Thanks For Reading ☺