Techtorial

PYTHON INTRO

Python

What Is Python ?

Python is an **object oriented computer programming language** often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems.

Techtorial

Python

Advantages of Python

1- Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance.

2- Python supports modules and packages, which encourages program modularity and code reuse.

3- The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Note

This advantages exactly taken from python.com

Techtorial

Python

Advantages of Python

- Easy to Read, Learn and Write. Python is a high-level programming language that has English-like syntax. ...
- Improved Productivity. Python is a very productive language. ...
- Interpreted Language. ...
- Dynamically Typed. ...
- Free and Open-Source. ...
- Vast Libraries Support. ...
- Portability.

Techtorial

Note

This advantages are less bookish good for telling in the interviews.

Coding Basics

Interpreter and Compiler

- Compiler and Interpreter are two different ways to translate a program from programming or scripting language to machine language.
- A compiler takes entire program and converts it into object code which is typically stored in a file. The object code is also referred as binary code and can be directly executed by the machine after linking. Examples of compiled programming languages are C and C++.
- An Interpreter directly executes instructions written in a programming or scripting language without previously converting them to an object code or machine code. Examples of interpreted languages are Perl, Python and Matlab.

Techtorial

Coding Basics

Interpreter and Compiler

- Following are some interesting facts about interpreters and compilers.
- 1) Both compilers and interpreters convert source code (text files) into tokens, both may generate a parse tree, and both may generate immediate instructions. The basic difference is that a compiler system, including a (built in or separate) linker, generates a stand alone machine code program, while an interpreter system instead performs the actions described by the high level program.

Coding Basics

- 2) Once a program is compiled, its source code is not useful for running the code. For interpreted programs, the source code is needed to run the program every time.
- 3) In general, interpreted programs run slower than the compiled programs.
- 4) Java programs are first compiled to an intermediate form, then interpreted by the interpreter.

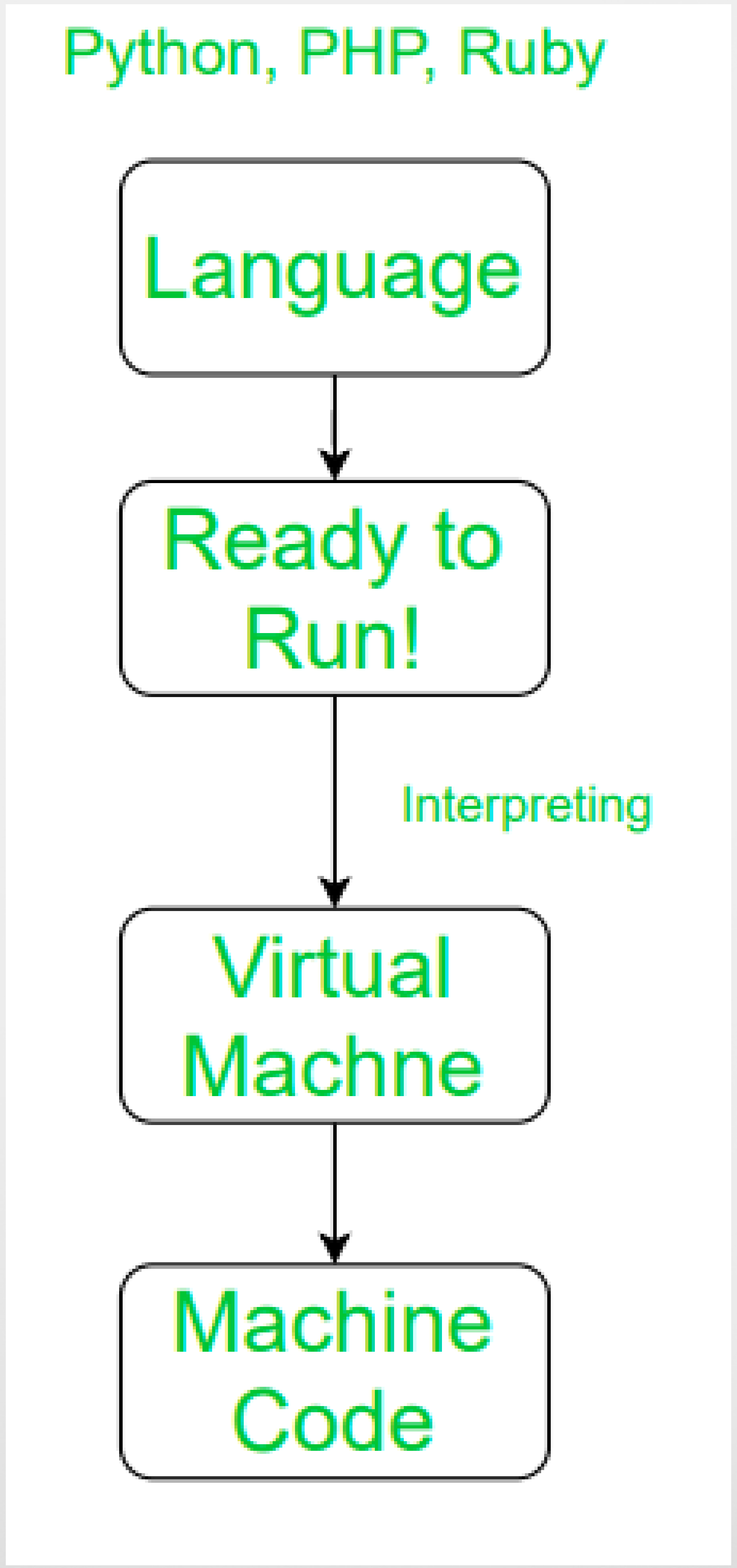
Techtorial

Python

What is Interpreted Language?

An interpreted language is a programming language which are generally interpreted, without compiling a program into machine instructions. It is one where the instructions are not directly executed by the target machine, but instead read and executed by some other program. Interpreted language ranges – JavaScript, Perl, Python, BASIC, etc.

Interpretation



Techtorial

Note

Image to demonstrate interpretation structure.

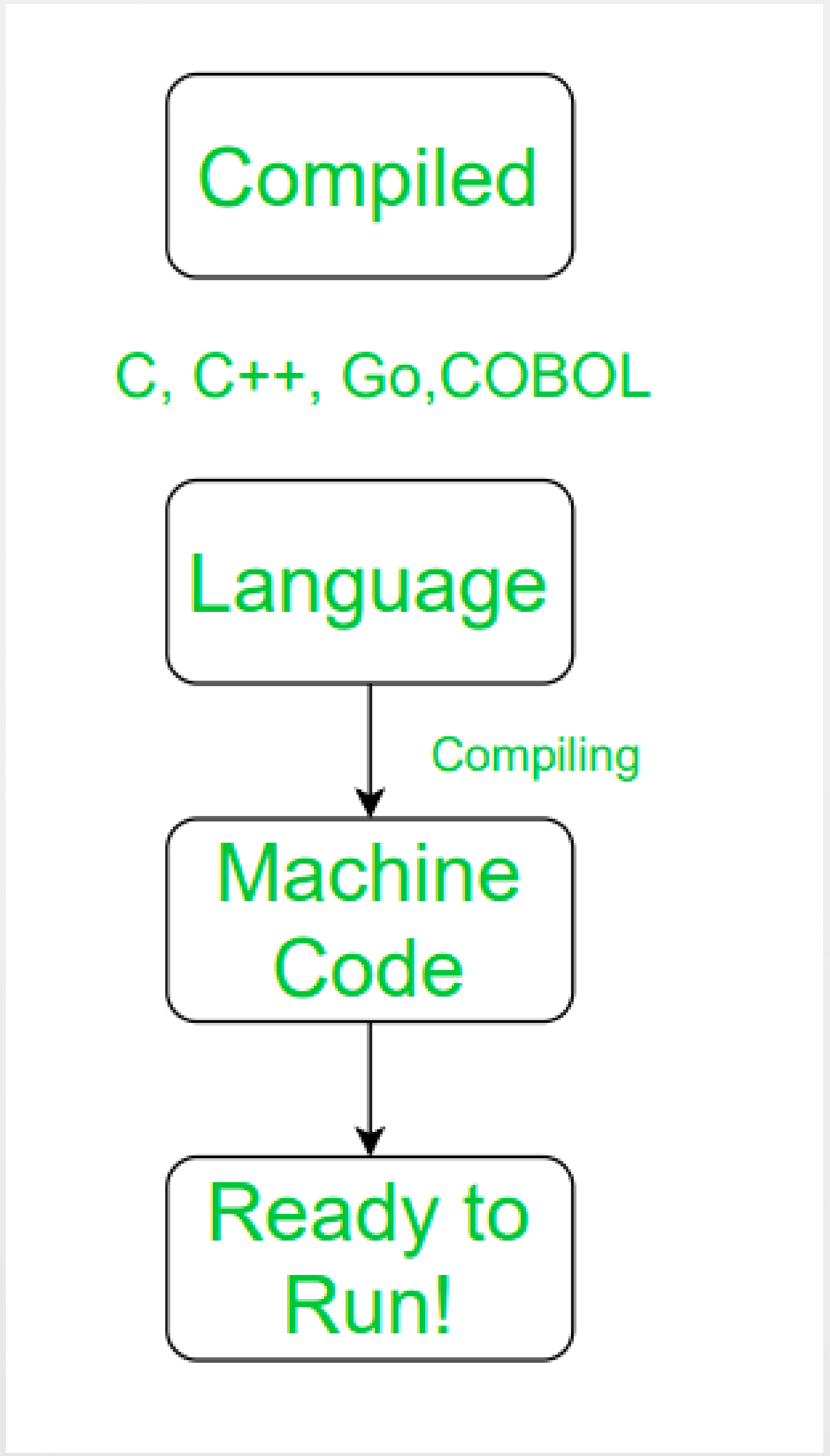
Compilation

What is Compiled Language?

A compiled language is a programming
language which are generally compiled and not interpreted.
It is one where the program, once compiled, is expressed in
the instructions of the target machine;
this machine code is undecipherable by humans.
Types of compiled language
– C, C++, C#, CLEO, COBOL, etc.

Techtorial

Compilation



Techtorial

Note

Image to demonstrate compilation.

Print

print() function

For the first steps, let's try to print something! To make Python print something, use the print() function. To print your desired text, you should place it inside brackets "..." or '...', otherwise Python will try to find a variable with that name (as to be discussed later). Thus, if you want to print a sentence, you should write and compile the following piece of code:

```
print("Some message")
```

Techtorial

Comment
s #

Comments

Firstly, it's important to note that the bigger the code is, the harder it is to understand the separate piece of code, even if it was written by yourself. Therefore, to solve it, we need to leave comments in our code such as instructions of what this piece of code does or something understandable to another person.
To mark something as a comment, begin the line with #

Techtorial

Comments

For example,

Input:

```
print('Python 1')
print('Python 2')
```

Output:

```
Python 1
Python 2
```

But if we place the hash # before print, this line will not be compiled.

Input:

```
#print('Python 1')
print('Python 2')
```

Output:

```
Python 2
```

Techtorial

Simple Math

Simple Math

Python also can calculate math expressions. The syntax is very simple: use

- + for addition,
- - for subtraction,
- * for multiplication, and
- / for division.

This time, we are also going to use print( ) if we want our result to be displayed. Lastly, there is no need to use "..." brackets, as we want to calculate math expression.

Techtorial

Simple Math

Example

```
# Addition
print(1039 + 948)

# Subtraction
print(10484 - 9274)

# Division
print(95873/472)
```

TASK

Calculate the following expressions and print the results:

1. **2580 plus 198.**
2. **23 multiplied by 42.**
3. **1512 divided by 27.**

Techtorial

Numeric
Data
Types

Numeric Data Types

Let's start our acquaintance with the simplest
and most popular data type - numeral. But, what is it?
A numeric data type is a data type that stores numbers.
It is worth noting that this data type is immutable - that is,
if we have already defined a variable as a number,
we will no longer be able to change this variable unless we redefine it.

Techtorial

Numeric Data Types

Create an Integer

```
# Creating an integer
a = 5
print(f'This is an integer = {a}')
```

Techtorial

Numeric Data Types

Create a Float

```
# Creating a floating point number
b = 10.6
print(f'This is a floating point number = {b}')
```

Techtorial

Numeric Data Types

Create a Complex Number

```
c = 43 + 6j
print(f'This is a complex number = {c}')
```

Techtorial

Extra Notes

Explanation to Printing

In the example, we can see an unusual moment for us, namely **print(f'This is an integer = {a}')**

These are the so-called **f-strings**. f-strings allow us to print strings with variables in a **single pair of quotes.**

Let's look at the syntax. All text we want to print must **be enclosed in quotation marks preceded by an f**, and any variables we want to print within that text **must be enclosed in curly braces.**

Techtorial

Math

There are several more useful math operations available in Python.

They are:

- // - integer division.
- % - the remainder of a division.
- ** - exponentiation.

Math

Please note, that we can't use the traditional ^ symbol for exponentiation in Python as it gives a different result.
The syntax for these operations is the same as the simplest arithmetic, for instance

18 // 4 is 4,

since we take the integer part of division (the division result is 4.5).

14 % 5 is 4

since we can subtract number 5 two times and there will be number **4 left**.

4**3 will be 64**

since the result of multiplying 4*4*4.(The value of 4 to the 3rd power)

Techtorial

Math

TASK

- First, calculate the integer part of the division of 234 by 32
- Second, find the remainder of the division of 356 by 17
- Third, find the value of 5 to the 6th power

Techtorial

Math

When are these operations useful ?

For example, we can modulate the situation:
let's say you have 50 dollars and need to buy as many packs of cookies as you can (each costs 6 dollars). You need to calculate the maximum number of packs you can buy and how much money will be left. With these operations, it can be easily done.
50//6 equals 8 since you subtract 6 as maximum 8 times. This will be the number of snacks we can buy.
50%6 equals 2 since 2 dollars will be left after subtracting 6 eight times.

Techtorial

TASK

Techtorial

Tasks

```
1:   print("Code start")
2:  # Replace name.
3:  print("Hello, my name is", "name")
4:  print("and I'm learning Python.")
5:  # Calculate math expression
6:  print((___)___)
7:  print("Code end")
```

1. On the third line within the print function, replace "name" with your name.
2. Perform an integer division of 2894 by 274 and raise the result to the 4 power (line 6).
3. Comment on the first and the seventh lines so they will not be shown in the console.