Python Programming

LECTURES BY

DR. AVINASH GULVE

Lecture -1

- History of python
- Features of python

History of Python

Created in 1989 by Guido van Rossum

- Created as a scripting language for administrative tasks
- Based on *All Basic Code (ABC)* and *Modula-3*
  - Added extensibility
- Named after comic troupe Monty Python

Released publicly in 1991

- Growing community of Python developers
- Evolved into well-supported programming language

History of Python

Modules

- Reusable pieces of software
- Can be written by any Python developer
- Extend Python's capabilities

Python Web site at www.python.org

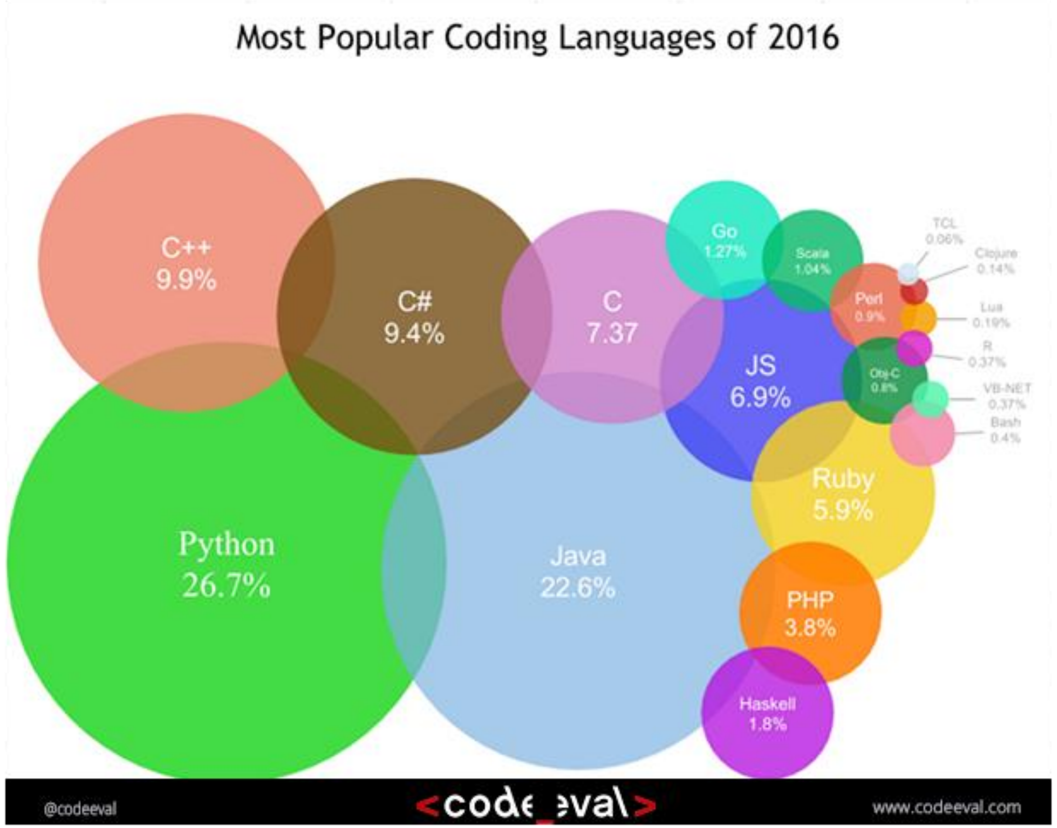
- Primary distribution center for Python source code, modules and documentation

History of Python

Python

- Designed to be portable and extensible
  - Originally implemented on UNIX
  - Programs often can be ported from one operating system to another without any change
- Syntax and design promote good programming practices and surprisingly rapid development times
  - Simple enough to be used by beginning programmers
  - Powerful enough to attract professionals

History of Python



Python today

Developed a large and active scientific computing and data analysis community

Now one of the most important languages for

- Data science
- Machine learning
- General software development

Packages: NumPy, pandas, matplotlib, SciPy, scikit-learn, statsmodels

What is python?

Object oriented language

Supports dynamic data type

Focused on development time

High-level internal object data types

It’s free (open source)!

Interpreted language

Independent from platforms

Simple and easy grammar

Automatic memory management

Python features

no compiling or linking	rapid development cycle
no type declarations	simpler, shorter, more flexible
automatic memory management	garbage collection
high-level data types and operations	fast development
object-oriented programming	code structuring and reuse, C++
embedding and extending in C	mixed language systems
classes, modules, exceptions	"programming-in-the-large" support
dynamic loading of C modules	simplified extensions, smaller binaries
dynamic reloading of C modules	programs can be modified without stopping

Python features

universal "first-class" object model	fewer restrictions and rules
run-time program construction	handles unforeseen needs, end-user coding
interactive, dynamic nature	incremental development and testing
access to interpreter information	metaprogramming, introspective objects
wide portability	cross-platform programming without ports
compilation to portable byte-code	execution speed, protecting source code
built-in interfaces to external services	system tools, GUIs, persistence, databases, etc.

Setting Up Python on Windows

Go to http://www.python.org and get the latest distribution

- Online tutorials
- Python related websites

Use the distribution on the CD ROM

- Examples from the book

Use all the defaults when installing

Run **Idle** from the Start Menu.

General Information

- Unlike C/C++ or Java, Python statements do not end in a semicolon
- In Python, indentation is the way you indicate the scope of a conditional, function, etc.
- No braces!
- Python is interpretive, meaning you don’t have to write programs.
- You can just enter statements into the Python environment and they’ll execute

The Python Shell

- Because Python is interpretive, you can do simple things with the shell
- You should have a `>>>` prompt
- Type in:

```
print(“hello, world”)
```

- You have written your first Python program
- This is good for simple calculations but not for real programming
- For programming, we’ll use Idle
- Idle will give you access to a shell but also to an IDE for writing and saving programs

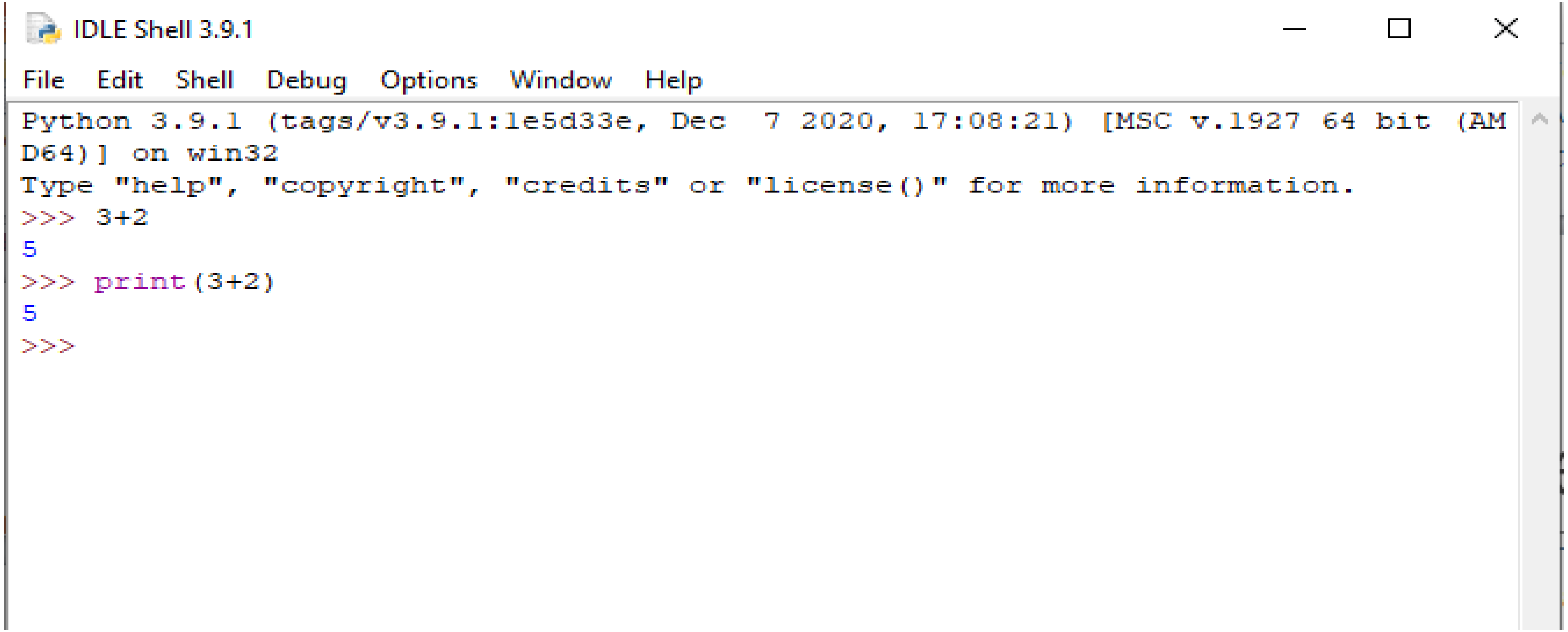
The Python Shell - IDLE

- For programming, we'll use Idle
- Idle will give you access to a shell but also to an IDE for writing and saving programs
- IDLE is an **I**ntegrated **D**eve**L**opment **E**nvironment for Python, typically used on Windows
- Multi-window text editor with syntax highlighting, auto-completion, smart indent and other.

The Python Shell

- Great for learning the language
- Great for experimenting with the library
- Great for testing your own modules
- Two variations: IDLE (GUI), python (command line)

The Python Shell



```
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec  7 2020, 17:08:21) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 3+2
5
>>> print(3+2)
5
>>>
```