



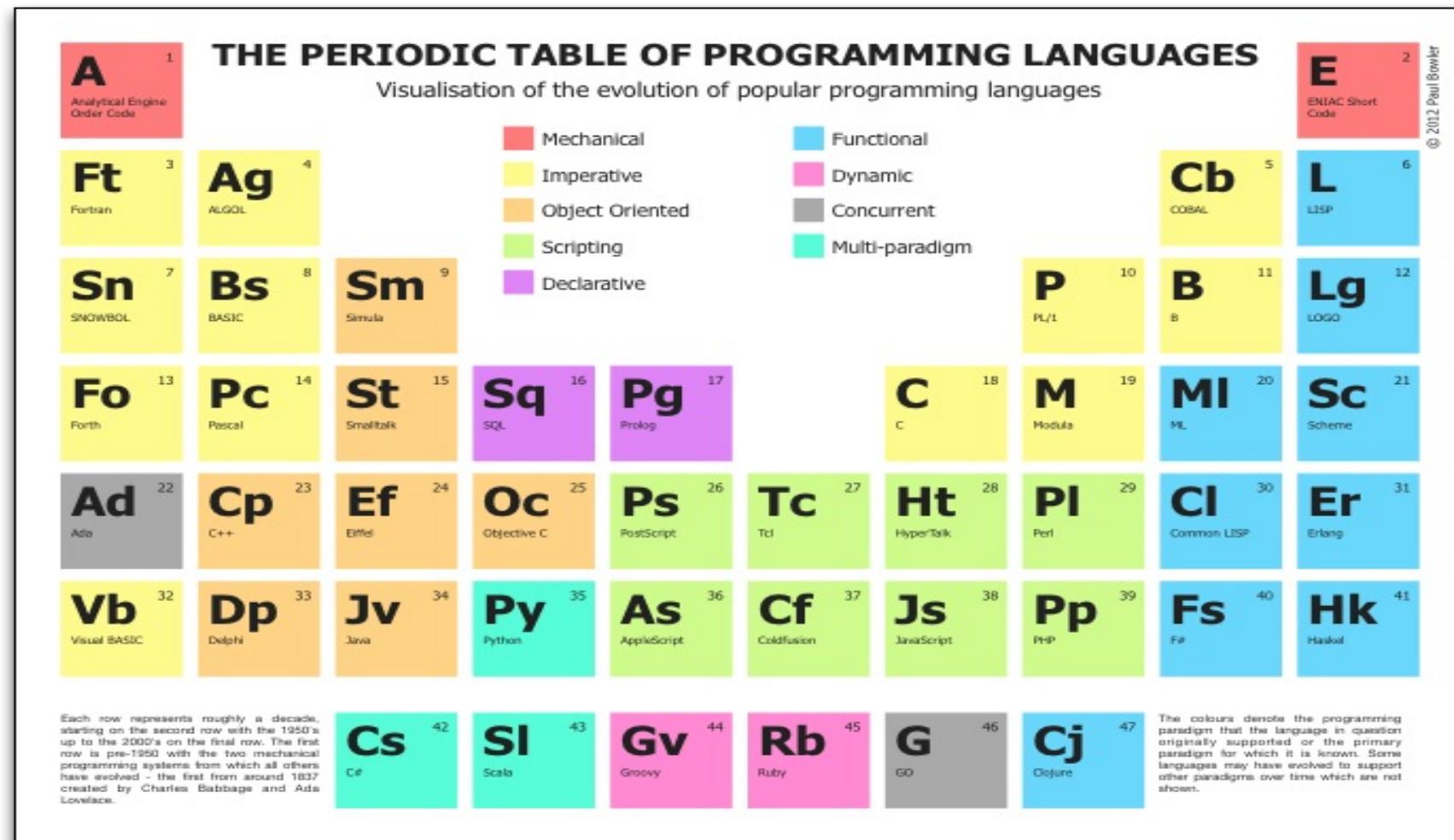
Programming for Fintech

Garbel Wong

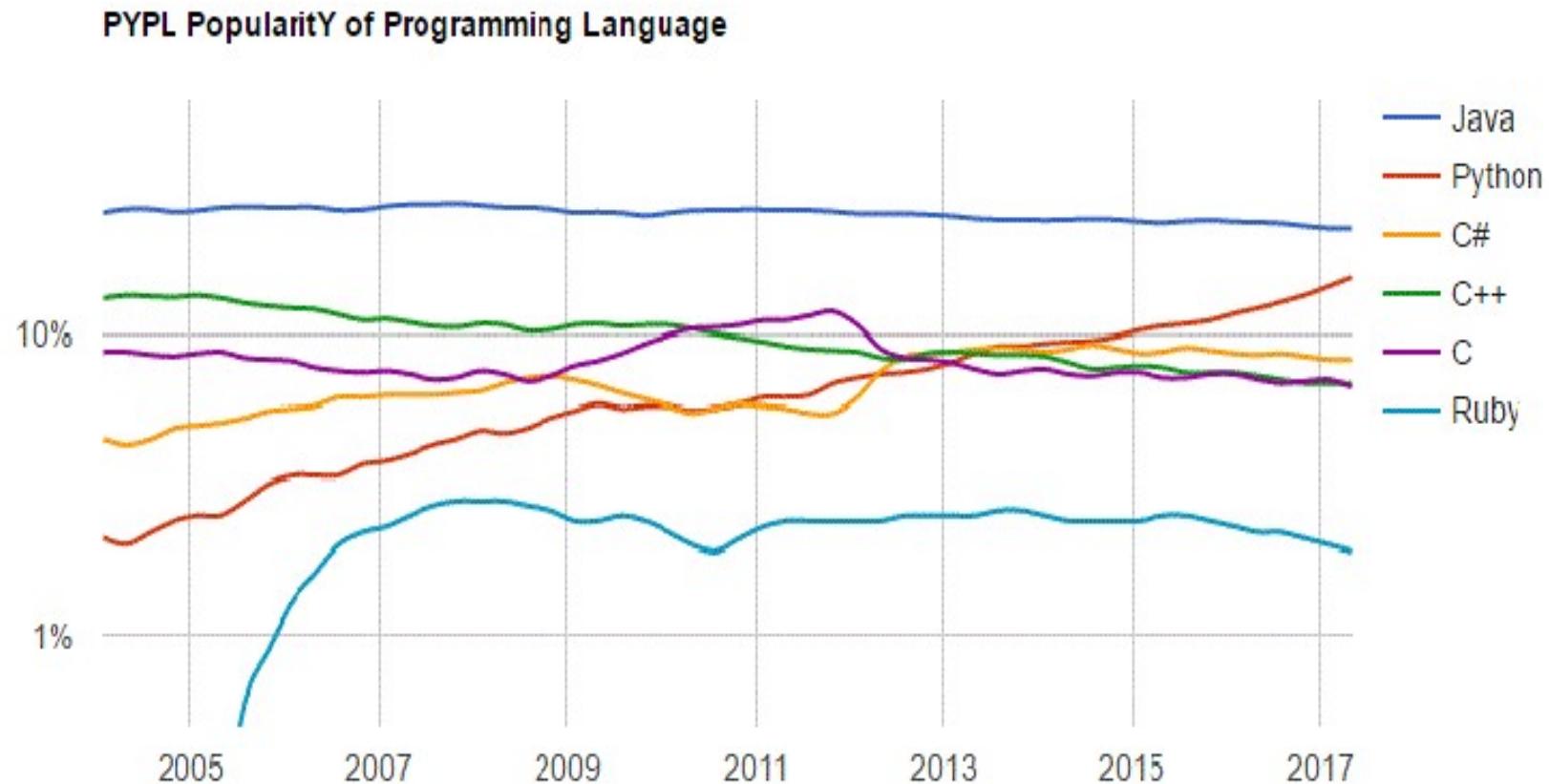
Popularity

- From supercomputers to Instagram, from labs to financial firms
- You can do machine learning, nature language processing, web pages and social networks
- Universities like Imperial College, M.I.T., Stanford, Harvard, NYU have switched to Python for programming courses in computer science and financial engineering

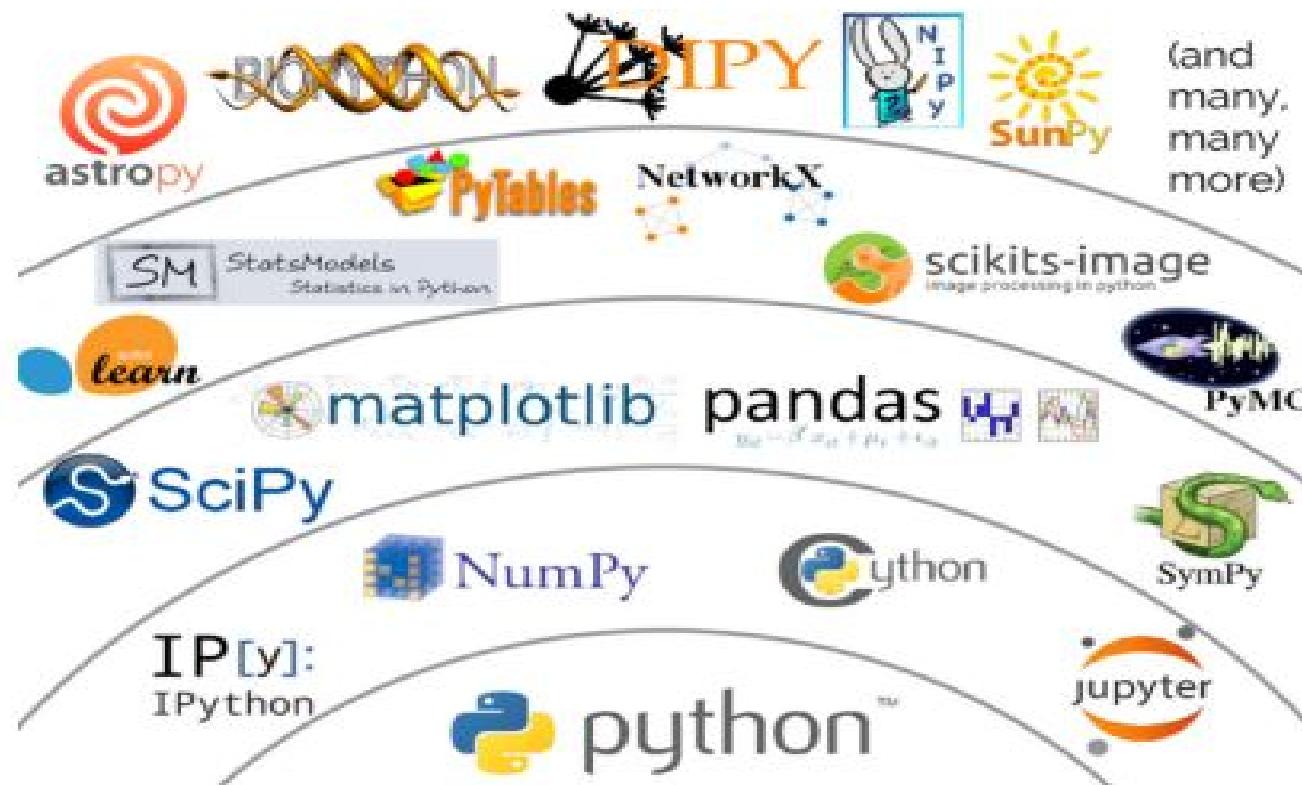
Periodic Table



Trend of Popularity

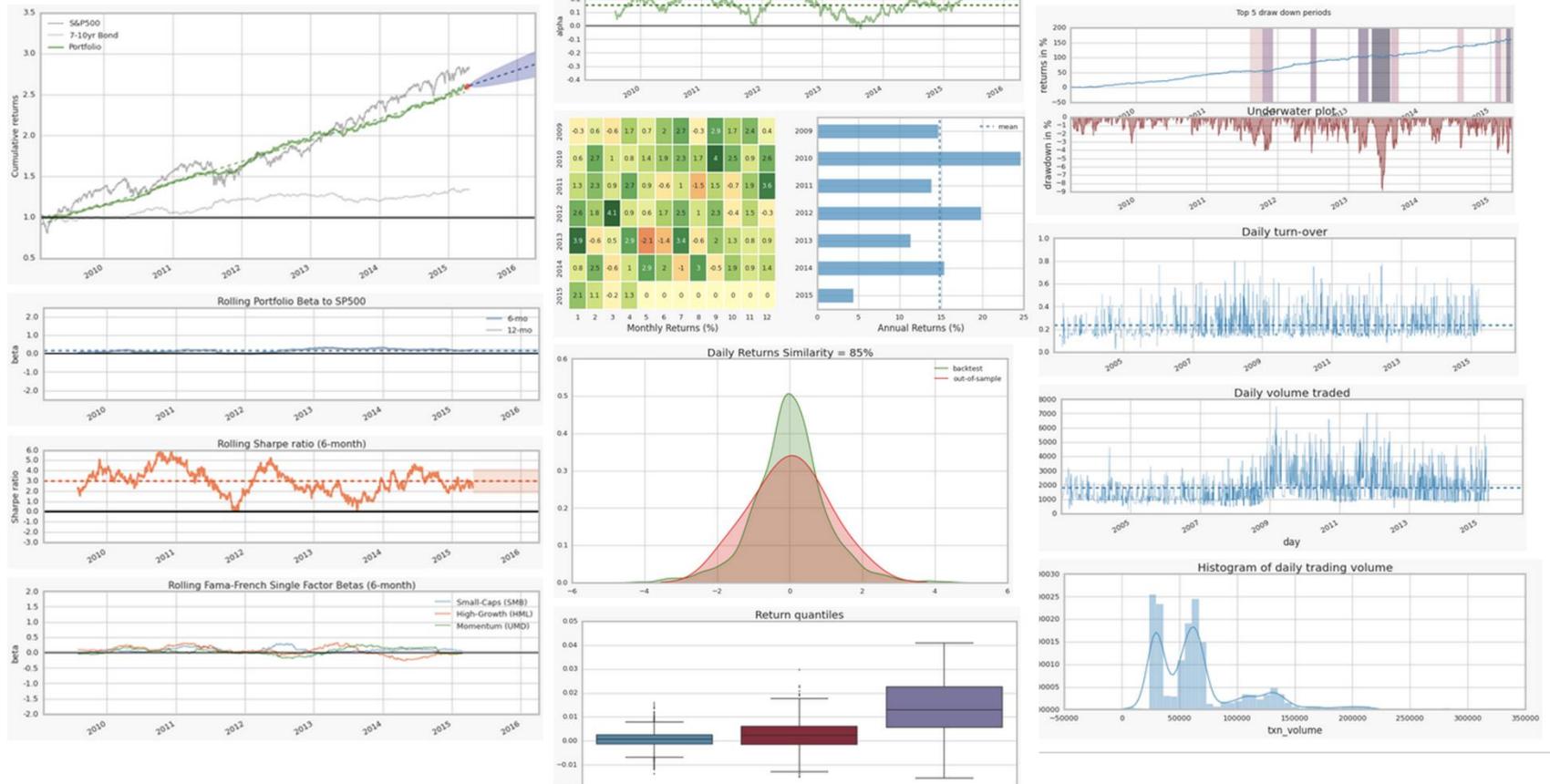


Python Ecosystem



<https://speakerdeck.com/jakevdp/the-state-of-the-stack-scipy-2015-keynote?slide=8>

Graphical Applications



Career Development

Equity Derivatives Quant

 Salary:	Competitive
 Location:	Hong Kong
 Job Type:	Permanent, Full time
 Company:	Webbe International
 Updated on:	12 Apr 17

74

people have
applied to this
job

Equity Derivatives Quant

Equity Derivatives Quant role within a tier 1 US bank.

This role is looking for expert quantitative analysts to support the equity derivatives traders across both index and single stocks with the main aim of quantitative optimization of trading and supporting the portfolio analytics tools.

A PHD or Master's degree in a quantitative discipline from a top tier institution is required as well as good expertise in statistical modelling and optimization.

A strong coding background with proficiency in C++ and Python is deemed essential.

Source: [efinancialcareers](#)

Career Development

JPMORGAN CHASE & CO.

Divisions

Locations

Programs

Why Us

Apply



Linear Quantitative Research

Linear Quantitative Research is an expert quantitative modeling group in J.P. Morgan, an unchallenged leader in financial engineering, statistical modeling and portfolio management. With more than 20 analysts worldwide, Linear Quantitative Research partners with traders, marketers and risk managers across all products and regions.

Quantitative skills are a core capability of J.P. Morgan, contributing critically to product innovation, effective risk management and appropriate financial and risk controls. The team's mission is to develop and maintain sophisticated mathematical models, cutting-edge methodologies and infrastructure to improve the performance of algorithmic trading strategies and promote the advanced electronic solutions to our clients worldwide. We also work closely with trading desks to develop statistical arbitrage strategies and other quantitative trading models.

Opportunities

New hires will benefit from both on the job training as well as intensive formal classroom training. Through the diversity of the businesses it supports and the variety of functions that it is responsible for, the Linear Quantitative Research group provides unique growth opportunities for its new interns to develop their abilities and their careers.

Roles and responsibilities include the following:

- Developing mathematical models for systematic quantitative trading strategies (for example, Index Arbitrage, Statistical Arbitrage and Market Making, etc), and for Risk Management (risk optimization, hedging strategies, optimal unwind, etc).
- Evaluating quantitative methodologies, back-testing and simulating quantitative models
- Supporting trading activities by explaining model behavior, carrying out scenario analyses, developing and delivering quantitative tools, and supporting analytics to the trading desk
- Designing and developing software frameworks for analytics and their delivery to systems and applications
- Position is located in Hong Kong

Qualifications

The ideal candidate will have :

- Earned a PhD or equivalent degree program in math, statistics, econometrics, financial engineering or computer science
- Exceptional analytical, quantitative and problem-solving skills
- Great communication and interpersonal skills
- Mastered advanced mathematics arising in financial modeling (i.e., probability theory, time series, econometrics)
- Strong software design and development skills using Python, Java and/or Matlab
- Good understanding of the various flows on an equity floor is a plus
- Knowledge in Asian market microstructure is a plus

Career Development

jobs > Accounting & Finance > Vice President - Senior Finance Analyst - Valuations

[Back to Search Results](#)

Vice President - Senior Finance Analyst - Valuations

Salary:	Competitive
Location:	Hong Kong
Job Type:	Permanent, Full time
Company:	Bank Of America / Merrill Lynch
Updated on:	22 Apr 17



Last application
19 Apr 17

Apply

Save

Share

Vice President - Senior Finance Analyst - Valuations

Job Description:

About Bank of America Merrill Lynch:

Our purpose as a firm is to make financial lives better, through the power of every connection. Across the world, we partner with leading corporate and institutional investors through our offices in more than 40 countries. In the U.S. alone, we serve almost all Fortune 500 companies and approximately 50 million consumers and small-business customers. We provide a full slate of financial products and services, from banking and investments to asset and risk management. We cover a broad range of asset classes, making us a global leader in corporate and investment banking, sales and trading.

Connecting Asia Pacific to the World

Our Asia Pacific team is spread across 23 offices in 12 markets. We are focused on connecting Asia to the world and the world to Asia, using our global expertise to ensure success is shared between us, our clients and our communities. Our regional footprint covers 12 currencies, more than a dozen languages and five time zones, placing us firmly among the region's leading financial services companies.

Bank of America Merrill Lynch is committed to attracting, recruiting and retaining top diverse talent from across the globe. Our diversity and inclusion mission is to actively promote an inclusive work environment where all employees have the opportunity to achieve personal success and contribute to the growth of our business. Each of our global Employee Networks bring together employees, create dialogue and awareness to support of our Diversity and Inclusion.

Position Description

The APAC Global Valuation Group (GVG) based at Bank of America Merrill Lynch provides valuation and independent price verification (IPV) support to the APAC Global Markets Businesses, covering Equities, Rates, FX, and Credit lines of business. Given we are a cross asset class group we are open to those with backgrounds in any of these products, multiple product experience is an asset.

Bank Of America / Merrill Lynch



[View All Jobs](#)

[Contact Details](#)



Specialist Insurance
Headhunter

[More jobs like this](#)

Vice President - Capital Management Finance Analyst
Associate Vice President, Senior Programming Analyst
Senior Vice President, Business Development - Global Trade and Receivables Finance
Assistant Vice President - Corporate Banking Finance
Vice President, Trade

Position Description

The APAC Global Valuation Group (GVG) team at Bank of America Merrill Lynch provides valuation and independent price verification (IPV) support to the APAC Global Markets Businesses, covering Equities, Rates, FX, and Credit lines of business. Given we are a cross asset class group we are open to those with backgrounds in any of these products, multiple product experience is an asset.

Key Responsibilities

- Define the methodology for IPV and associated Liquidity/Model/Deal Specific Valuation Adjustments (VA) for any given product or risk (including regular methodology reviews and advice on re-calibrations) - knowledge of XVA's are a plus.
- Create the standard process via system or spreadsheet on how to execute the methodology in line with compliance requirements
- Perform reviews and approval of all market data constructions, functions and models used in production to value transactions
- Review and Challenge the IPV processes conducted by Business Finance and Control (BFC)
- Review of material new deals and changes to models and market data configurations
- Assist with implementation of strategic projects, such as key risk management and system enhancements
- Working closely with Traders, Market Risk Management, Model Risk Management, FO Quants, BFC, senior managers on new product, model development, model sign-off etc.
- Define Fair Value Hierarchy categorization and justification and create the framework of a given product, risk, or portfolio as appropriate
- Define Liquidity (covered/non-covered) and create the framework of a given product, risk, or portfolio as appropriate

Key Requirements

- 5 years of Financial Modeling experience or equivalent
- 5+ years of derivatives valuation and independent price verification experience
- Working knowledge of MS Office applications including VBA, MarkIT, Reuters, Bloomberg
- Demonstrate ability to interact with personnel at all levels of a Top tier Financial Organization
- Desired University degree in Finance, Engineering, Mathematics or related degree. Post-graduate university degree (MFE or PhD preferred)
- Computer Programming Skills - Python
- FRM/CFA/ACA

Philosophy

➤ Python is a “*high-level programming language and its core design philosophy is all about code readability and a syntax which allows programmers to express concepts in a few lines of code*” by Guido van Rossum.

Features

- Ease of learning and maintaining
- A broad standard library
- Developer productivity
- Program portability
- Interactive Mode
- Software quality
- Readable, beautiful and brevity
- Support Structured / OOP Styles
- Easy integration with other languages C++, Fortran



Put Into Practice

- Open source license
- User created modules for ready use
- Run on many platform: Windows, Linux, Mac
- Fewer syntactical constructions than other languages
- Designed to be highly readable which uses English keywords
- Very clear syntax + large and comprehensive standard library
- Python Package Index (PyPI) – modules repository 30,000+
- A high-level, interpreted, interactive and object-oriented scripting language.
- Lots and lots of reference materials available including books, video lectures and tutorials

Applications

- GUIs
- Internet Scripting
- Systems Programming
- Component Integration
- Database Programming
- Numeric and Scientific Programming
- Gaming, Images, Data Mining, Robots, Excel...

Disadvantage

- Not always be as fast as that of compiled languages such as C and C++

Where we can find



Integrated Development Environment

- Includes all modules and integrated Development Environment (IDE):
- Install each module e.g. pip or conda (eg. pip install numpy):

Python

- Python.org: <https://www.python.org>
- Numpy: <https://sourceforge.net/projects/numpy>
- Scipy: <https://www.scipy.org/scipylib/download.html>
- Matplotlib: <http://matplotlib.org/users/installing.html>

Anaconda

- <https://www.continuum.io/downloads>
- Anaconda(Jupyter Notebook,Spyder,Ipython,R):
- Jupyter Notebook in a web browser

PyCharm

- PyCharm: <https://www.jetbrains.com/pycharm/download>

Python Installation

➤ How to Install Python

There are two common ways to download and install Python

1. Download Python from its official website. You have to manually install libraries.

- 2. Download Anaconda. It comes with Python software along with pre-installed popular libraries.

Python Environment

- ✓ Python Official Website: <http://www.python.org>
- ✓ Python is available on a wide variety of platforms
(Windows / Linux / Mac)
- ✓ Install Python
- ✓ Setting up PATH

Python.org

The screenshot shows the Python.org website with a dark blue header. The header includes a navigation bar with links for Python, PSF, Docs, PyPI, Jobs, and Community. Below the header is the Python logo. To the right of the logo is a search bar and user account links for Socialize and Sign In. A yellow button labeled 'Menu' is also present. The main content area features a code editor window displaying Python code for arithmetic operations. To the right of the code is a sidebar with the heading 'Intuitive Interpretation' and text explaining Python's calculation rules. At the bottom, there is a promotional message about Python's capabilities.

Python

PSF

Docs

PyPI

Jobs

Community

python™

Menu

Search

AA

Socialize

Sign In

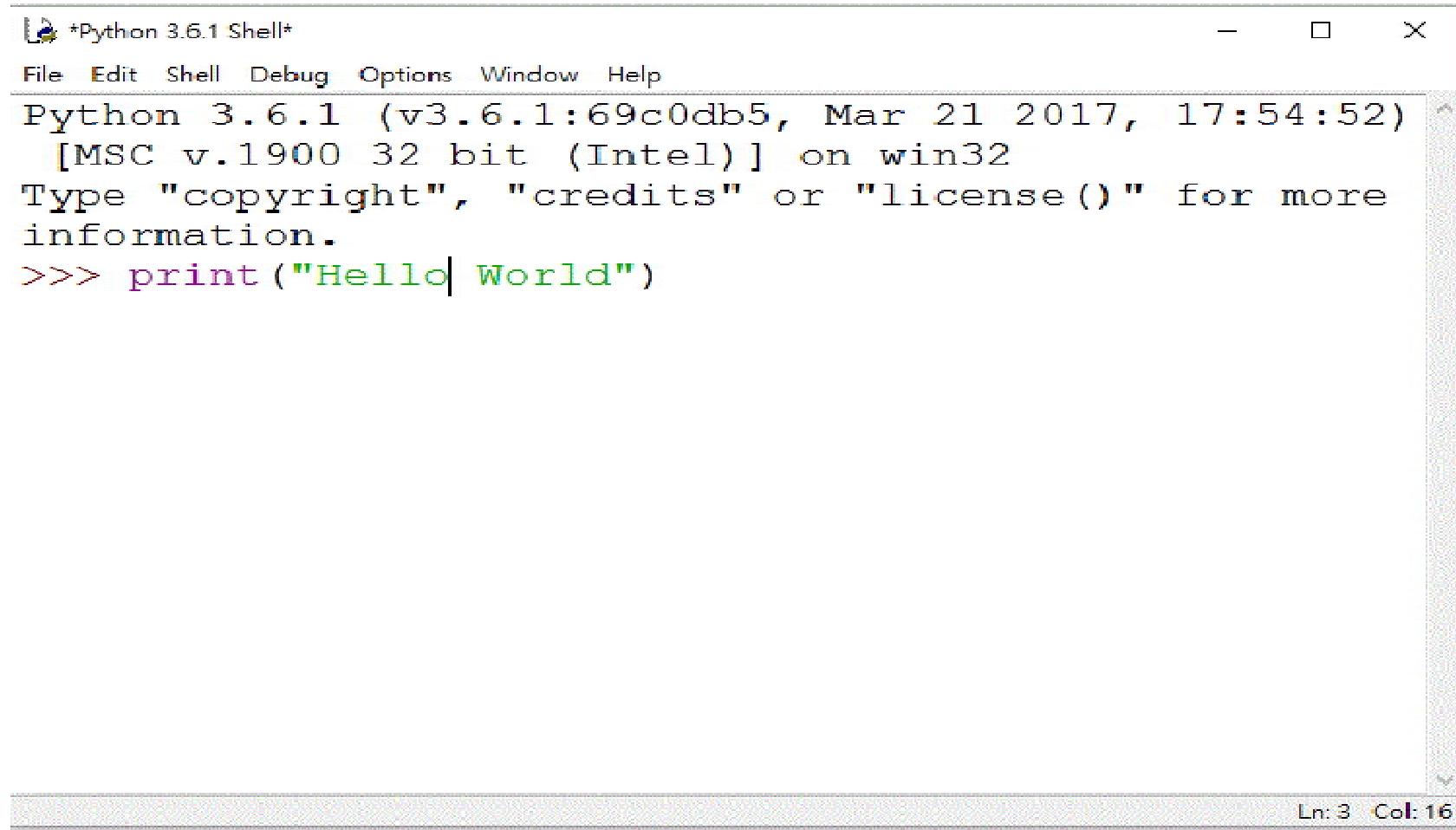
```
# Python 3: Simple arithmetic
>>> 1 / 2
0.5
>>> 2 ** 3
8
>>> 17 / 3 # classic division returns a float
5.666666666666667
>>> 17 // 3 # floor division
5
```

Intuitive Interpretation

Calculations are simple with Python, and expression syntax is straightforward: the operators `+`, `*` and `/` work as expected; parentheses `()` can be used for grouping. [More about simple math functions in Python 3.](#)

Python is a programming language that lets you work quickly and integrate systems more effectively. [»» Learn More](#)

Python Shell



The screenshot shows a window titled "Python 3.6.1 Shell". The window has a standard title bar with icons for minimize, maximize, and close. Below the title bar is a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main area of the window displays the Python interpreter's startup message:

```
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52)
[MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more
information.
```

Below this, a command prompt shows the execution of a print statement:

```
>>> print("Hello| World")
```

In the bottom right corner of the window, there is a status bar with "Ln: 3 Col: 16".

Interactive and Script modes

The screenshot shows two windows of the Python 3.6.1 IDE. On the left is the 'First-class Object.py' editor window, which contains the following code:

```
import math
trig = math.sin, math.cos, math.tan
for fn in trig:
    print(fn, fn(math.pi/3))
```

On the right is the 'Python 3.6.1 Shell' window, which displays the output of running the script:

```
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54 :52) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Python36-32\Doc\First-class Object.py =====
<built-in function sin> 0.8660254037844386
<built-in function cos> 0.5000000000000001
<built-in function tan> 1.7320508075688767
>>> |
```

The status bar at the bottom indicates 'Ln: 5 Col: 0' for the editor and 'Ln: 8 Col: 4' for the shell.

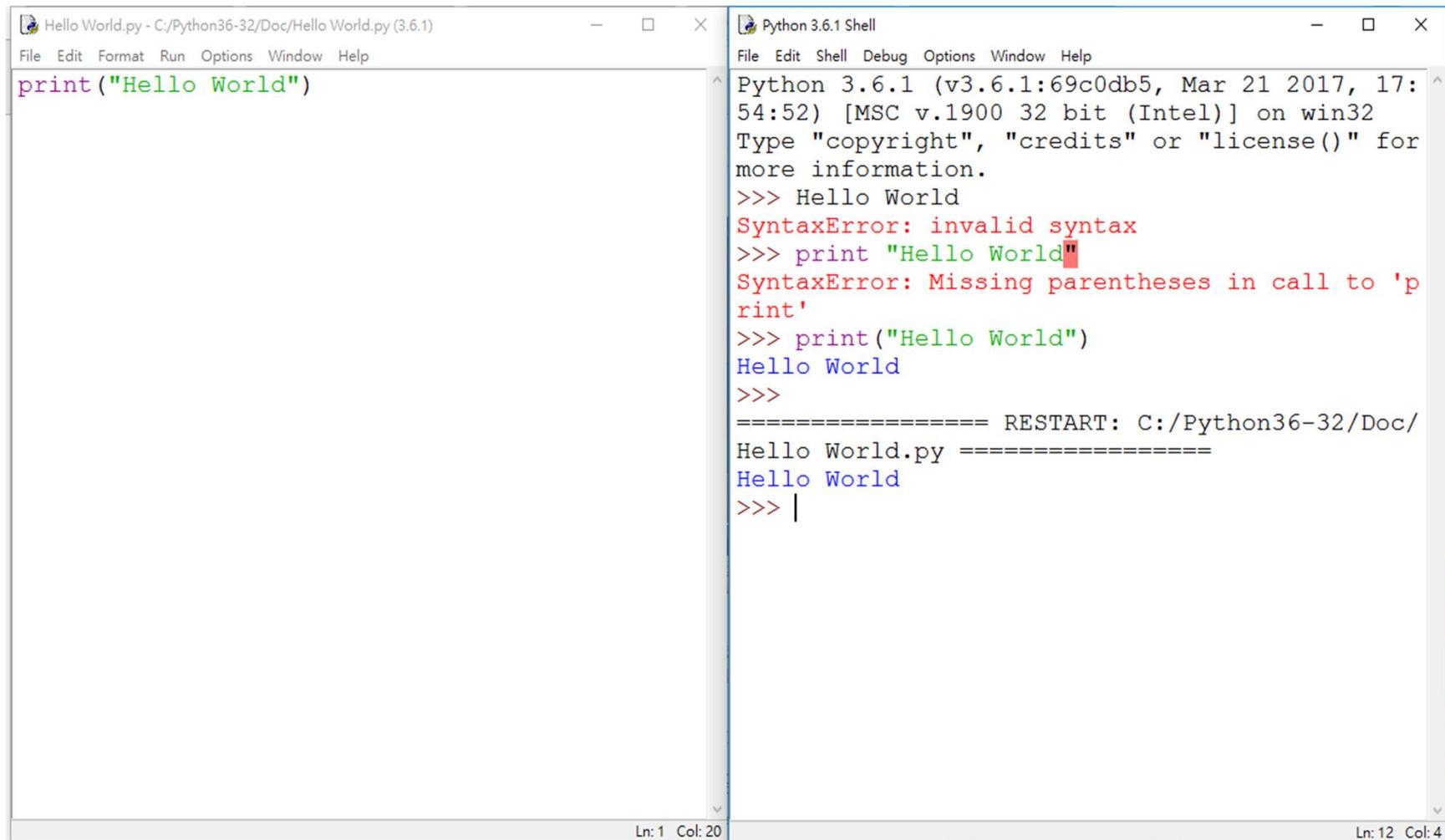
First Python Program

- In interactive mode programming
- Run script from the Command line
- Type and enter in Python prompt:
 - `print ("Hello, World!")`
- Press enter
- "Hello, World!"

First Python Program

- In script mode programming
- `print ("Hello, World!")`
- Save a Python script file and include code:
- File extension : py
- In command shell run Hello.py file

Python Outlook



The image shows a screenshot of a Windows desktop with two windows side-by-side. On the left is a code editor window titled "Hello World.py - C:/Python36-32/Doc/Hello World.py (3.6.1)". It contains a single line of Python code: `print("Hello World")`. On the right is a Python shell window titled "Python 3.6.1 Shell". The shell displays the following interaction:

```
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> Hello World
SyntaxError: invalid syntax
>>> print "Hello World"
SyntaxError: Missing parentheses in call to 'print'
>>> print("Hello World")
Hello World
>>>
===== RESTART: C:/Python36-32/Doc/
Hello World.py =====
Hello World
>>> |
```

The shell shows several attempts to run the code. The first attempt fails with a SyntaxError because it uses the old-style string concatenation operator (""). The second attempt fails with a SyntaxError because it lacks parentheses around the argument. The third attempt succeeds, printing "Hello World". The final attempt is a blank line starting with '>>> |'.

Basic Knowledge

- Everything in Python is an object that has:
 - An Identifier (id)
 - A value(mutable or immutable)
 - Mutable: List and Dictionary. When you alter the items, the id is still same.
 - Immutable: Tuple

Python Syntax

- ✓ Python Identifiers - case sensitive
- ✓ Reserved Words - 33 key words
- ✓ Lines and Indentation - tab
- ✓ Quotation in Python – ‘object_name’
- ✓ Comments in Python - # or “ ” or ” ”
- ✓ Multiple Line Statements - instructions

Reserve Word

```
and      del      from     not      while
as       elif     global    or       with
assert   else     if       pass     yield
break   except   import   print
class   exec     in       raise
continue finally  is       return
def    for      lambda  try
```

Lines and Indentation

- Blocks of code are denoted by line indentation
- if True:
 - print("Right")
- else:
 - print("False")

Quotation

- Uses quotes to denote string literals

word = 'Finance'

sentence = "I am a Quant."

paragraph = """This is a small world. It is consisted of
different peoples."""

Comments

A hash sign **#** begins a comment.

```
# first comment
```

```
print ("Hello, How are you") # second comment
```

Multiple line comments

```
"""This is a python workshop.
```

```
It will provide you the basic concept"""
```

Python Operators

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

Object Types

Type	Ordered	Mutable	Examples
Number	N/A	No	3.14159, 137, 99L, 4+-5j, 0x0a
String	Yes	No	'A string', "Peter and Paul"
List	Yes	Yes	[3, [4, 'two'], [6E-2, 10e4], -4L]
Dictionary	No	Yes	{'HSBC':'volatility', 'Nomura':'strike'}
Tuple	Yes	No	(2, 'three', -5j, 04, 0x55, 7L)

Assigning Values to Variables

```
people = 200          # An integer assignment
```

```
pi   = 3.14159        # A floating point
```

```
surname  = "Michael"  # A string
```

```
print (people)
```

```
200
```

```
print (pi)
```

```
3.14159
```

```
print (surname)
```

```
Michael
```

Number

- Number objects are created when you assign a value to them.
- staff = 200 # An integer assignment
- pi = 3.14159 # A floating point
- print (staff)
- 200
- print (pi)
- 3.14159

String

- Uses quotes to denote string literals
- word = 'Financial Engineer'
- sentence = "We are the world."
- paragraph = .""""This is a small world. It is made up of people.""""

String

- mystring = 'How are you?'
- print (mystring)
- How are you?
- print (mystring[0])
- H
- print (mystring[3:6])
- ar
- #5th an 6th characters

Data Structure

- List
- Dictionary
- Tuple

List

- Items belonging to a list can be of different data types and are separated by commas and enclosed within square brackets [].

```
mylist = [ 'HSBC', 68.60 , '700', 2.2356 , 240.5 ]
```

```
print (mylist)
```

```
['HSBC', 68.6, '700', 2.2356, 240.5]
```

```
print (mylist[0])
```

```
HSBC
```

```
print (mylist[2:4])
```

```
['700', 2.2356]
```

List

- Ordered sequence of objects
- Accessed by offset
- Variable-length, heterogeneous, and arbitrarily nestable
- Of the category “mutable sequence”
- Contents may be changed during program execution
- Arrays of object references

Dictionary

```
mydict = {}
```

```
mydict = {'name':'HSBC','code':'0005','business':'Bank'}
```

```
print (mydict)
```

```
{'name': 'HSBC', 'code': '0005', 'business': 'Bank'}
```

```
# Prints all the keys
```

```
print (mydict.keys())
```

```
dict_keys(['name', 'code', 'business'])
```

```
# Prints all the values
```

```
print (mydict.values())
```

```
dict_values(['HSBC', '0005', 'Bank'])
```

Dictionary

- An ordered sequence of key-value pairs
- Accessed by key, not offset
- Variable-length, heterogeneous, and arbitrarily nestable
- Of the category “mutable mapping”
- Tables of object references (hash tables)

Tuple

- A tuple is a sequence of immutable Python objects.
- Data are read only, write-protected and cannot be modified
- Tuples use parentheses ()

```
tup1 = ('finance', 'maths', 2016, 2015)
```

```
tup2 = (1, 2, 3, 4, 5 )
```

```
tup3 =( "a", "b", "c", "d")
```

```
print ("tup1[0]: ", tup1[0])
```

```
tup1[0]: finance
```

```
print ("tup2[2:4]: ", tup2[2:5])
```

```
tup2[2:4]: (3, 4, 5)
```

Tuple

- Ordered sequence of arbitrary objects
- Accessed by offset
- Of the category “immutable sequence”
- Contents may not changed once created
- Fixed-length, heterogeneous, and arbitrarily nestable
- Arrays of object references

Operators

Operators	Description
<code>a or b</code>	Logical OR (b evaluated only if a false), anonymous function
<code>a and b</code>	Logical AND (b evaluated only if a is true)
<code>not a</code>	Logical negation
<code><, <=, >, >=, ==, <>, !=, is, is not, in, not in</code>	Comparison operators, identity tests, sequence membership
<code>a b</code>	Bitwise OR
<code>a ^ b</code>	Bitwise EXCLUSIVE OR
<code>a & b</code>	Bitwise AND
<code>a << n, a >> n</code>	Shift a left or right by n bits
<code>a + b, a - b</code>	Numeric addition or sequence concatenation, subtraction
<code>a * b, a / b, a % b</code>	Multiplication or sequence repetition, division, modulus
<code>-a, +a, ~a</code>	Unary negation, identity, bitwise negation
<code>a[i], a[i:j], a.b, a(...)</code>	Indexing and slicing sequences, qualification, function call

Python Function

```
def function_name( parameters ):  
    "function_docstring" - optional  
    statements  
    return [expression]
```

Python Modules

- A module allows you to logically organize your Python code
- Break down large programs into small manageable and organized files
- Grouping related codes into a module makes the codes easier to understand and use
- A file containing Python statements and definitions

Creating Python Modules

- A file containing Python code, for e.g.: sample.py, is called a module and its module name would be called e.g. sample.add() and sample.sub().
- **sample.py file**

```
def add(a,b):  
    """This program adds two numbers  
    and return the result"""  
    result = a + b  
    return result  
  
def sub(a,b):  
    """This program subtracts two numbers  
    and return the result"""  
    result = a - b  
    return result
```

The import Statement

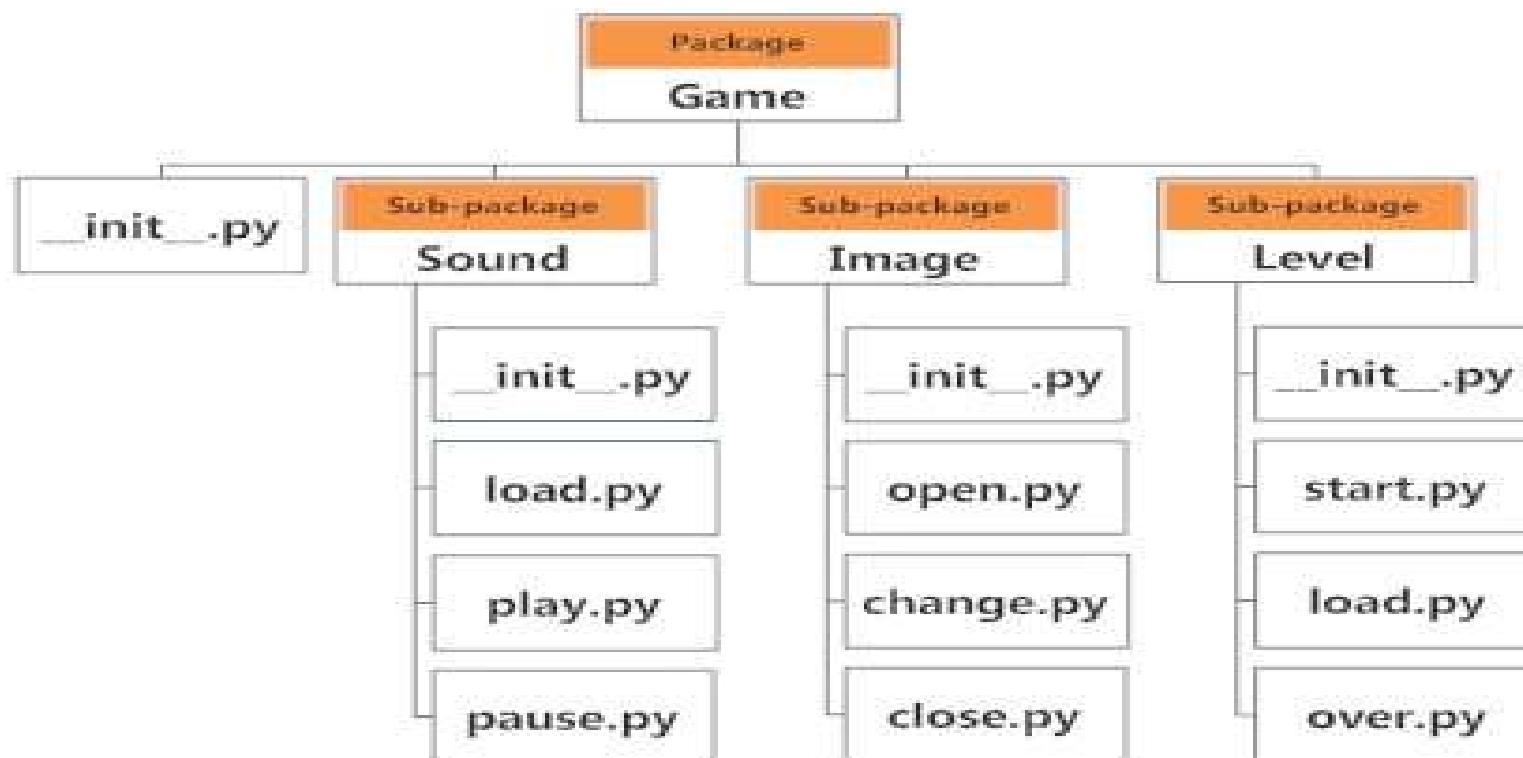
- Using a Python file as a module by executing an **import statement**.
- SYNTAX:
- `import module1, module2,... moduleN`
- e.g. `import sample`

From Module Import Statement

- Import specific attributes from a module into the current namespace.
- SYNTAX:
 - `from module import func1, func2, ... funcN`
 - Import all names from a module into the current namespace
- SYNTAX:
 - `from module import *`

Modules and Packages

➤ Import Game.Level.start / from Game.Level import start



<https://www.programiz.com/python-programming/package>

PyPI - Python Package Index

➤ <https://pypi.python.org/pypi>

Updated	Package	Description
2017-04-27	appier_extras 0.10.22	Appier Framework Extra Elements
2017-04-27	ipywidgets 7.0.0a0	IPython HTML widgets for Jupyter
2017-04-27	SimpleSoapy 1.5.1	Simple pythonic wrapper for SoapySDR library
2017-04-27	vsc-install 0.10.26	vsc-install provides shared setuptools functions and classes for python libraries developed by UGent's HPC group
2017-04-27	creh-logs 0.1.2	Log application
2017-04-27	widgetsnbextension 3.0.0a1	IPython HTML widgets for Jupyter
2017-04-27	bltest 1.0.0	BodyLabs unittest extensions
2017-04-27	hologram-python 0.5.4	Library for accessing Hologram Cloud at https://hologram.io
2017-04-27	chellow 1838	Web Application for checking UK utility bills.
2017-04-27	djangocms-spa-vue-js 0.1.13	This package prepares your django CMS and vue.js project to create a single-page application (SPA).
2017-04-27	pySOT 0.1.35	Surrogate Optimization Toolbox
2017-04-27	djangocms-spa 0.1.13	Run your django CMS project as a single-page application (SPA)
2017-04-27	awss 0.9.6	AWS Shortcuts for Command-Line Instance Control
2017-04-27	smartystreets_python_sdk 2.1.1	An official library to help Python developers easily access the Smarty Streets APIs
2017-04-27	clickhouse-cli 0.1.9.7	A third-party client for the Clickhouse DBMS server.
2017-04-27	yakumo 0.10.2	Pythonic Unified OpenStack Client Library
2017-04-27	os-win 0.5.0	Windows / Hyper-V library for OpenStack projects
2017-04-27	blmath 1.0.2	A collection of math related utilities used by many bits of BodyLabs code
2017-04-27	dila 0.0.0	Dila is a open source web-based translation platform for translators, content creators and developers.
2017-04-27	gbdxtools 0.11.5	Additional s3 functionality.
2017-04-27	pyalgs 0.0.3	Python implementation of Robert Sedgwick's Algorithm (Part I and Part II) Coursera course
2017-04-27	sqlalchemy-postgres-autocommit 0.2.0	A library to use SQLAlchemy with PostgreSQL in an autocommit mode.
2017-04-27	desmod 0.5.0	Discrete Event Simulation Modeling using SimPy
2017-04-27	lastcast 0.1.8	Scrobble music to last.fm from Chromecast.
2017-04-27	datary 0.0.22	Datary Python sdk lib

Anaconda

- Anaconda is a free open source distribution of the Python and R programming languages for large-scale data processing, predictive analytics, and scientific computing
- Simplify package management and deployment
- Enterprise-ready Python for big data
- Bundles together over 453 of the most widely used Python (version 3.6) packages

Anaconda Popular Packages

1. NumPy | numpy.org

N-dimensional array for numerical computation

2. SciPy | scipy.org

Collection of numerical algorithms and toolboxes, including signal processing and optimization

3. Matplotlib | matplotlib.org

Plotting library for Python

4. Pandas | pandas.pydata.org

Powerful Python data analysis toolkit

5. Seaborn | stanford.edu/~mwaskom/software/seaborn/

Statistical data visualization

6. Bokeh | bokeh.pydata.org

Interactive web visualization library

7. SciKit-Learn | scikit-learn.org/stable

Python modules for machine learning and data mining

8. NLTK | nltk.org

Natural language toolkit

9. Notebook | jupyter.org

Web-based interactive computational environment combines code execution, rich text, mathematics, plots and rich media

10. R essentials | conda.pydata.org/docs/r-with-conda.html

R with 80+ of the most used R packages for data science
"conda install -c r r-essentials"

Anaconda



The image shows the sign-up page for Anaconda Cloud. The header features the Continuum Analytics logo and navigation links for Gallery, About, Pricing, Anaconda (which is highlighted in blue), Help, Download Anaconda, and Sign In. A search bar with a magnifying glass icon is positioned above the main content area. The background has a green-to-black gradient with a geometric polygonal pattern. On the left, the Anaconda Cloud logo is displayed, consisting of a stylized green 'A' icon followed by the text "ANACONDA CLOUD". Below the logo, three bullet points describe the service: "Where packages, notebooks, and environments are shared.", "Powerful collaboration and package management for open source and private projects.", and "Public projects and notebooks are always free." To the right of these points, a sign-up form is shown with fields for username, email, password, and confirmation. A note specifies that the password must contain at least one lowercase letter, one numeral, and seven characters. A checkbox for accepting terms and conditions is present, along with a "Sign up!" button. A small disclaimer at the bottom states that by clicking "Sign up!", the user agrees to privacy policy and terms of service.

CONTINUUM[®]
ANALYTICS

Gallery About Pricing **Anaconda** Help Download Anaconda Sign In

Search Anaconda Cloud

Sign Up Sign In

New to Anaconda Cloud? Sign up!

Pick a username

Your email

Use at least one lowercase letter, one numeral, and seven characters.

Create a password

Confirm password

I accept the [Terms & Conditions](#)

Sign up!

By clicking "Sign up!" you agree to our [privacy policy](#) and [terms of service](#). We will send you account related emails occasionally.

Anaconda

[Download for Windows](#)[Download for macOS](#)[Download for Linux](#)

Anaconda 4.4.0

For Windows

Anaconda is BSD licensed which gives you permission to use Anaconda commercially and for redistribution.

[Changelog](#)

1. Download the installer
2. Optional: Verify data integrity with [MD5 or SHA-256](#) [More info](#)
3. Double-click the `.exe` file to install Anaconda and follow the instructions on the screen

Behind a firewall? Use these [zipped Windows installers](#)

Python 3.6 version

[64-BIT INSTALLER \(437M\)](#)[32-BIT INSTALLER \(362M\)](#)

Python 2.7 version

[64-BIT INSTALLER \(430M\)](#)[32-BIT INSTALLER \(354M\)](#)

Ipython Notebook

- One of the hottest Python projects out there
- Tab completion, introspection, interactive debugger, command history
- Designed to enhance the productivity in every way
- IPython HTML notebook is #winning
- File extension : ipynb

Jupyter

- Jupyter is useful for presenting your work to others or creating step by step project report as it can combine code, output, words, graphics and etc

Anaconda

The screenshot shows the Anaconda Navigator application window. The left sidebar contains links for Home, Environments, Projects (beta), Learning, Community, Documentation, Developer Blog, and Feedback, along with social media icons for Twitter, YouTube, and GitHub. The main area displays a grid of application cards:

- Jupyter notebook**: Version 5.0.0. Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis. [Launch](#)
- qtconsole**: Version 4.2.1. PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more. [Launch](#)
- spyder**: Version 3.1.2. Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features. [Launch](#)
- anaconda-fusion**: Version 1.0.2. Integration between Excel® and Anaconda via Notebooks. Run data science functions, interact with results and create advanced visualizations in a code-free app inside Excel. [Install](#)
- glueviz**: Version 0.10.4. Multidimensional data visualization across files. Explore relationships within and among related datasets.
- orange3**: Version 3.4.1.
- rstudio**: Version 1.0.136. A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

At the top right, there are buttons for Upgrade Now, Sign in to Anaconda Cloud, Refresh, and standard window controls.

Jupyter

jupyter

Logout

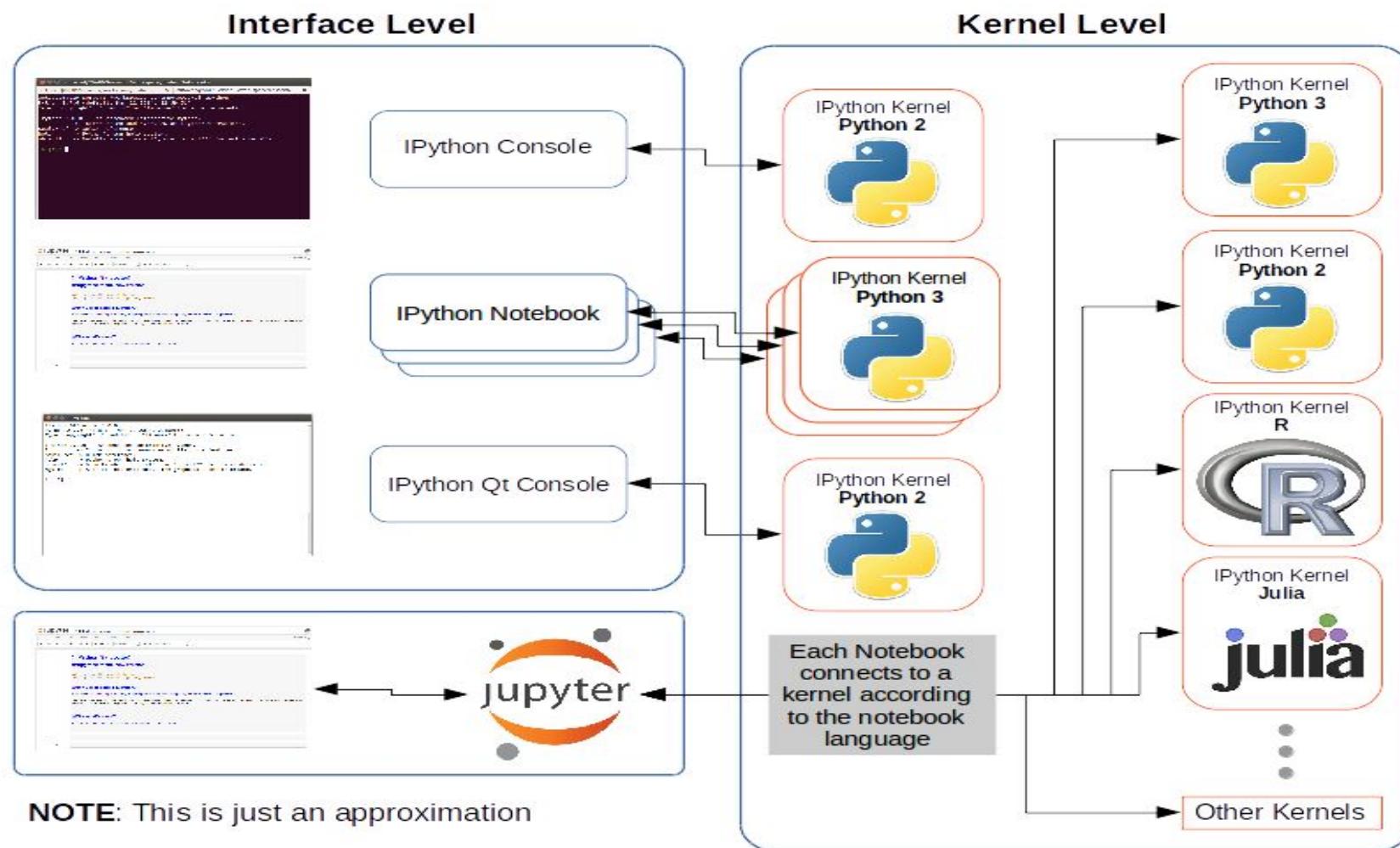
Files Running Clusters

Select items to perform actions on them.

Upload New

	Name ↑	Last Modified ↑
	3D Objects	a year ago
	Anaconda3	a month ago
	AnacondaProjects	2 months ago
	Contacts	22 days ago
	Corel	5 years ago
	Desktop	2 days ago
	Documents	13 days ago
	Downloads	2 days ago
	dwhelper	12 days ago
	Favorites	22 days ago
	Links	22 days ago
	Music	22 days ago
	OneDrive	21 days ago
	Pictures	22 days ago
	PycharmProjects	5 months ago
	Roaming	6 years ago
	Saved Games	22 days ago

Architecture



Python for Finance

- Technology costs for new business development and innovation in finance industry
- Technology and talent as barriers to entry in the finance industry
- Increasing high speeds, high frequencies, large data volumes
- The rise of real-time and big data analysis

Reference Book

