Variables:

<variable name> = <value>

Variable assignment works from left to right. So the following will give you an syntax error.

Rules for variable naming:

1. Variables names must start with a letter or an underscore.
2. The remainder of your variable name may consist of letters, numbers and underscores
3. Names are case sensitive.
4. No Keywords

Overview of Data Types

Friday, April 10, 2020 12:22 AM

Data types are nothing but variables you use to reserve some space in memory. Python variables do not need an explicit declaration to reserve memory space. The declaration happens automatically when you assign a value to a variable.

1. String Data Type
2. Set Data Types
3. Numbers data type
4. List Data Type
5. Dictionary Data Type
6. Tuple Data Type

Comment and User Input

Thursday, April 9, 2020 11:53 PM

- Comments are used to explain code when the basic code itself isn't clear

1. Single line comments
2. Multi-Line comments

- User Input

To get input from the user, use the input function (note: in Python 2.x, the function is called raw_input instead, although Python 2.x has its own version of input that is completely different):

Operators and Operations

Thursday, April 9, 2020 11:58 PM

Python does common mathematical operators on its own, including integer and float division, multiplication, exponentiation, addition, and subtraction. The math module (included in all standard Python versions) offers expanded functionality like trigonometric functions, root operations, logarithms, and many more.

Boolean Operator:
AND, OR, NOT

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

Operator Precedence:

Python follows PEMDAS rule. PEMDAS stands for Parentheses, Exponents, Multiplication and Division, and Addition and Subtraction

Built-in Modules

Friday, April 10, 2020 12:06 AM

A module is a file containing Python definitions and statements. Function is a piece of code which execute some logic.

To check the built in function in python we can use dir(). If called without an argument, return the names in the current scope. Else, return an alphabetized list of names comprising (some of) the attribute of the given object, and of attributes reachable from it.

Creating own modules:

A module can be created by creating a .py file.

List:

The Python List is a general data structure widely used in Python programs. They are found in other languages, often referred to as dynamic arrays. They are both mutable and a sequence data type that allows them to be indexed and sliced. The list can contain different types of objects, including other list objects.

Examples in Pycharm Editor

Dictionary:

Parameter	Details
key	The desired key to lookup
value	The value to set or return

A dictionary is an example of a key value store also known as Mapping in Python. It allows you to store and retrieve elements by referencing a key. As dictionaries are referenced by key, they have very fast lookups. As they are primarily used for referencing items by key, they are not sorted.

Examples in Pycharm Editor

Block Indentation

Friday, April 10, 2020 10:16 AM

Python uses indentation to define control and loop constructs. This contributes to Python's readability, however, it requires the programmer to pay close attention to the use of whitespace. Thus, editor miscalibration could result in code that behaves in unexpected ways.

Python uses the colon symbol (:) and indentation for showing where blocks of code begin and end (If you come from another language, do not confuse this with somehow being related to the ternary operator). That is, blocks in Python, such as functions, loops, if clauses and other constructs, have no ending identifiers.

Conditionals

Friday, April 10, 2020 12:22 AM

Conditional expressions, involving keywords such as if, elif, and else, provide Python programs with the ability to perform different actions depending on a boolean condition: True or False. This section covers the use of Python conditionals, boolean logic, and ternary statements.

Example of Conditional Expression (or "The Ternary Operator") in Pycharm

String Methods

Thursday, April 16, 2020 12:28 PM

Python's string type provides many functions that act on the capitalization of a string. These include:
str.casefold
str.upper
str.lower
str.capitalize
str.title
str.swapcase

Functions

Thursday, April 16, 2020 1:20 PM

Functions in Python provide organized, reusable and modular code to perform a set of specific actions. Functions
simplify the coding process, prevent redundant logic, and make the code easier to follow.

Python has many built-in functions like print(), input(), len(). Besides built-ins you can also create your own functions to do more specific jobs—these are called user-defined functions.
Using the def statement is the most common way to define a function in python. This statement is a so called single clause compound statement with the following syntax:

```
def function_name(parameters):
    statement(s)
```

Functional Programming

Thursday, April 16, 2020 4:41 PM

Functional programming decomposes a problem into a set of functions. Ideally, functions only take inputs and
produce outputs, and don't have any internal state that affects the output produced for a given input.
functional techniques common to many languages: such as lambda, map, reduce.

Decorators

Friday, April 17, 2020 1:30 PM

A decorator takes in a function, adds some functionality and returns it. In this article, you will learn how you can create a decorator and why you should use it.

What are decorators in Python?

Python has an interesting feature called **decorators** to add functionality to an existing code.
This is also called **metaprogramming** as a part of the program tries to modify another part of the program at compile time.

Comprehensions

Friday, April 17, 2020 2:22 PM

List Comprehension:

A **list comprehension** creates a new list by applying an expression to each element of an iterable. The most basic form is:

[<expression> for <element> in <iterable>]

There's also an optional 'if' condition:

[<expression> for <element> in <iterable> if <condition>]

Dictionary Comprehension:

A dictionary comprehension is similar to a list comprehension except that it produces a dictionary object instead of a list.

Iterators

Friday, April 17, 2020 3:15 PM

Iterators are everywhere in Python. They are elegantly implemented within for loops, comprehensions, generators etc. but hidden in plain sight.

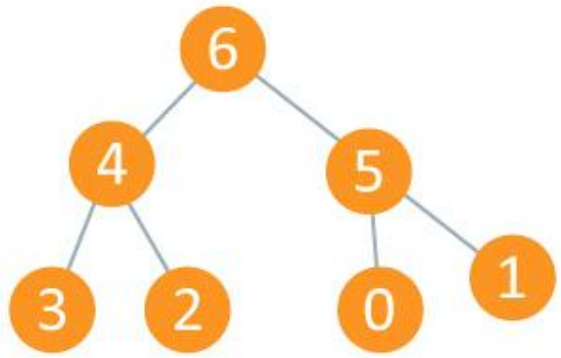
Iterator in Python is simply an object that can be iterated upon. An object which will return data, one element at a time.

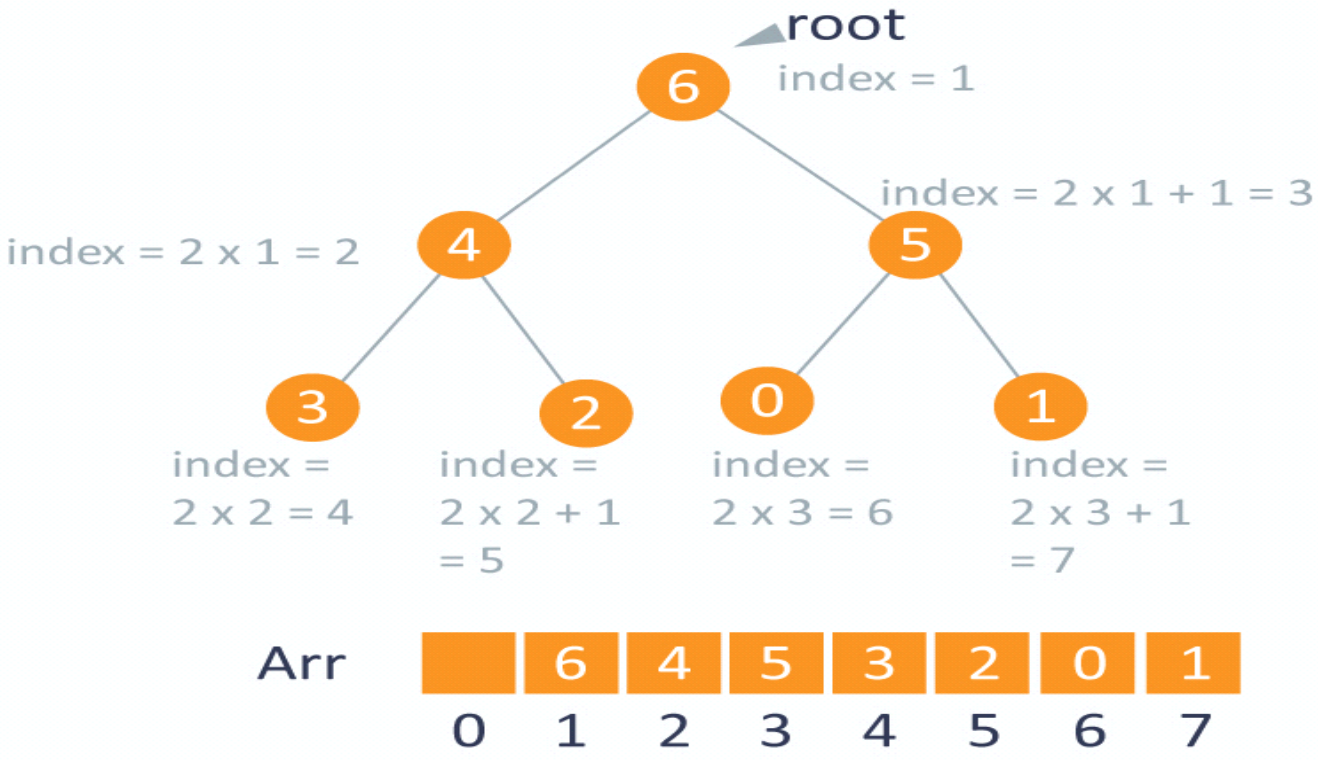
Technically speaking, Python **iterator object** must implement two special methods, __iter__() and __next__(), collectively called the **iterator protocol**.

Heap Queue

Friday, April 17, 2020 4:09 PM

A heap is a tree-based data structure in which all the nodes of the tree are in a specific order.





OS module

Friday, April 17, 2020 4:53 PM

Given a local directory with the following contents:

└── dir1
 |── subdir1
 └── subdir2

We want to create the same subdir1, subdir2 under a new directory dir2, which does not exist yet.

```
import os
```

Collections

Friday, April 17, 2020 9:57 PM

The built-in collections package provides several specialized, flexible collection types that are both high performance and provide alternatives to the general collection types of dict, list, tuple and set.

The module also defines abstract base classes describing different types of collection functionality (such as MutableSet and ItemsView).

Date and Time

Saturday, April 18, 2020 12:41 AM

Date and time objects may be categorized as “aware” or “naive.”

By default all datetime objects are naive. To make them timezone-aware, you must attach a tzinfo object, which provides the UTC offset and timezone abbreviation as a function of date and time.

Example of Computing Time differences in Pycharm

Date Formatting examples

Exception Handling

Saturday, April 18, 2020 10:35 AM

Errors detected during execution are called exceptions and are not unconditionally fatal. Most exceptions are not handled by programs; it is possible to write programs that handle selected exceptions.
There are specific features in Python to deal with exceptions and exception logic. Furthermore, exceptions have a rich type hierarchy, all inheriting from the BaseException type.

Use try, catch, finally and else to handle exceptions

File Handling

Saturday, April 18, 2020 11:52 AM

Syntax: with open(filename, 'r') as f

Parameter	Details
filename	the path your file or, if the file is in the working directory, the filename of your File
access_mode	string value that determines how the file is opened **

There are different modes you can open a file with, specified by the mode parameter. These include:

'r' - reading mode. The default. It allows you only to read the file, not to modify it. When using this mode the file must exist.
'w' - writing mode. It will create a new file if it does not exist, otherwise will erase the file and allow you to write to it.
'a' - append mode. It will write data to the end of the file. It does not erase the file, and the file must exist for this mode.
'rb' - reading mode in binary. This is similar to r except that the reading is forced in binary mode. This is also a default choice.
'r+' - reading mode plus writing mode at the same time. This allows you to read and write into files at the same time without having to use r and w.
'rb+' - reading and writing mode in binary. The same as r+ except the data is in binary
'wb' - writing mode in binary. The same as w except the data is in binary.
'w+' - writing and reading mode. The exact same as r+ but if the file does not exist, a new one is made. Otherwise, the file is overwritten.
'wb+' - writing and reading mode in binary mode. The same as w+ but the data is in binary.
'ab' - appending in binary mode. Similar to a except that the data is in binary.
'a+' - appending and reading mode. Similar to w+ as it will create a new file if the file does not exist. Otherwise, the file pointer is at the end of the file if it exists.
'ab+' - appending and reading mode in binary. The same as a+ except that the data is in binary.

Classes and Objects

Saturday, April 18, 2020 12:35 PM

Python offers itself not only as a popular scripting language, but also supports the object-oriented programming paradigm. Classes describe data and provide methods to manipulate that data, all encompassed under a single object. Furthermore, classes allow for abstraction by separating concrete implementation details from abstract representations of data.

Species: Homo Sapiens

What he/she does?

1. Walk
2. Talk
3. Run

How two objects/person be uniquely identified?

Attributes/Properties:

1. Name
2. Hair color
3. Eyes color
4. Height
5. Weight and so on

Methods/behavior:

1. Run
2. Walk
3. Sleep

Manipulation on the Attributes

1. Rename : manipulating object attributes
2. Sorting with height, weight etc.

Examples are covered in the pyCharm editor, refer lecture on Object Oriented Programming.

Work sample:

1. Class name: Bike
2. Attributes: ?
3. Method: ?
4. Objects: ?

Operator Overloading

Saturday, April 18, 2020 6:08 PM

Python operators work for built-in classes. But same operator behaves differently with different types. For example, the + operator will, perform arithmetic addition on two numbers, merge two lists and concatenate two strings.

1. 5 + 6
2. [1, 2, 3] + [4, 5, 6]
3. "hello " + "world"

This feature in Python, that allows same operator to have different meaning according to the context is called operator overloading.

Binary Operators

Operator	Magic Method
+	__add__(self, other)
–	__sub__(self, other)
*	__mul__(self, other)
/	__truediv__(self, other)
//	__floordiv__(self, other)
%	__mod__(self, other)
**	__pow__(self, other)

Comparison Operators

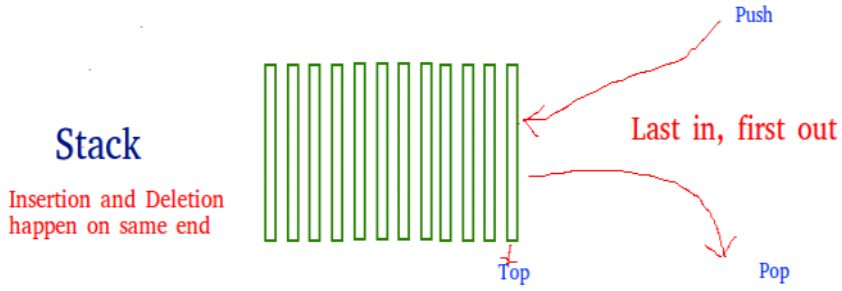
Operator	Magic Method
<	__lt__(self, other)
>	__gt__(self, other)
<=	__le__(self, other)
>=	__ge__(self, other)
==	__eq__(self, other)
!=	__ne__(self, other)

Assignment Operators

Operator	Magic Method
-=	__isub__(self, other)
+=	__iadd__(self, other)
***=**	__imul__(self, other)
/=	__idiv__(self, other)
//=	__ifloordiv__(self, other)
%=	__imod__(self, other)

Stack Implementation using Python

Sunday, April 19, 2020 9:01 AM



Consider an example of plates stacked over one another in the canteen. The plate which is at the top is the first one to be removed

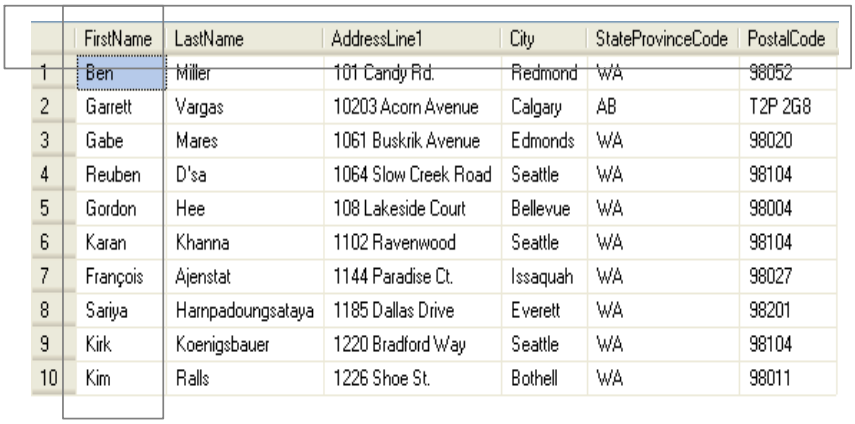
Creating a Stack class with a List Object in Pycharm

Most common usage of stack:
Parsing Parentheses

Stacks are often used for parsing. A simple parsing task is to check whether a string of parentheses are matching.

For example, the string ([]) is matching, because the outer and inner brackets form pairs. ()<>) is not matching, because the last) has no partner. ([)] is also not matching, because pairs must be either entirely inside or outside other pairs.

Consider COMPANY table with the following records –



	FirstName	LastName	AddressLine1	City	StateProvinceCode	PostalCode
1	Ben	Miller	101 Candy Rd.	Redmond	WA	98052
2	Garrett	Vargas	10203 Acorn Avenue	Calgary	AB	T2P 2G8
3	Gabe	Mares	1061 Buskrik Avenue	Edmonds	WA	98020
4	Reuben	D'sa	1064 Slow Creek Road	Seattle	WA	98104
5	Gordon	Hee	108 Lakeside Court	Bellevue	WA	98004
6	Karan	Khanna	1102 Ravenwood	Seattle	WA	98104
7	François	Ajenstat	1144 Paradise Ct.	Issaquah	WA	98027
8	Sariya	Harnpadoungsataya	1185 Dallas Drive	Everett	WA	98201
9	Kirk	Koenigsbauer	1220 Bradford Way	Seattle	WA	98104
10	Kim	Ralls	1226 Shoe St.	Bothell	WA	98011

SQLite **CREATE TABLE** statement is used to create a new table in any of the given database. Creating a basic table involves naming the table and defining its columns and each column's data type.

Syntax

Following is the basic syntax of CREATE TABLE statement.

```
CREATE TABLE database_name.table_name(
   column1 datatype PRIMARY KEY(one or more columns),
   column2 datatype,
   column3 datatype,
   .....
   columnN datatype
)
```

SQLite **INSERT INTO** Statement is used to add new rows of data into a table in the database.

Syntax

Following are the two basic syntaxes of INSERT INTO statement.

```
INSERT INTO TABLE_NAME [(column1, column2, column3,...columnN)]
VALUES (value1, value2, value3,...valueN);
```

SQLite **SELECT** statement is used to fetch the data from a SQLite database table which returns data in the form of a result table. These result tables are also called **result sets**.

Syntax

Following is the basic syntax of SQLite SELECT statement.

```
SELECT column1, column2, columnN FROM table_name;
```

Here, column1, column2 ... are the fields of a table, whose values you want to fetch. If you want to fetch all the fields available in the field, then you can use the following syntax −

```
SELECT * FROM table_name;
```

Multithreading

Sunday, April 19, 2020 12:46 PM

Threads allow Python programs to handle multiple functions at once as opposed to running a sequence of commands individually.

Queue

Monday, April 20, 2020 12:00 PM

Example of Multi-Threading: let's say you want to download several pages of a website and compile them into a single page.

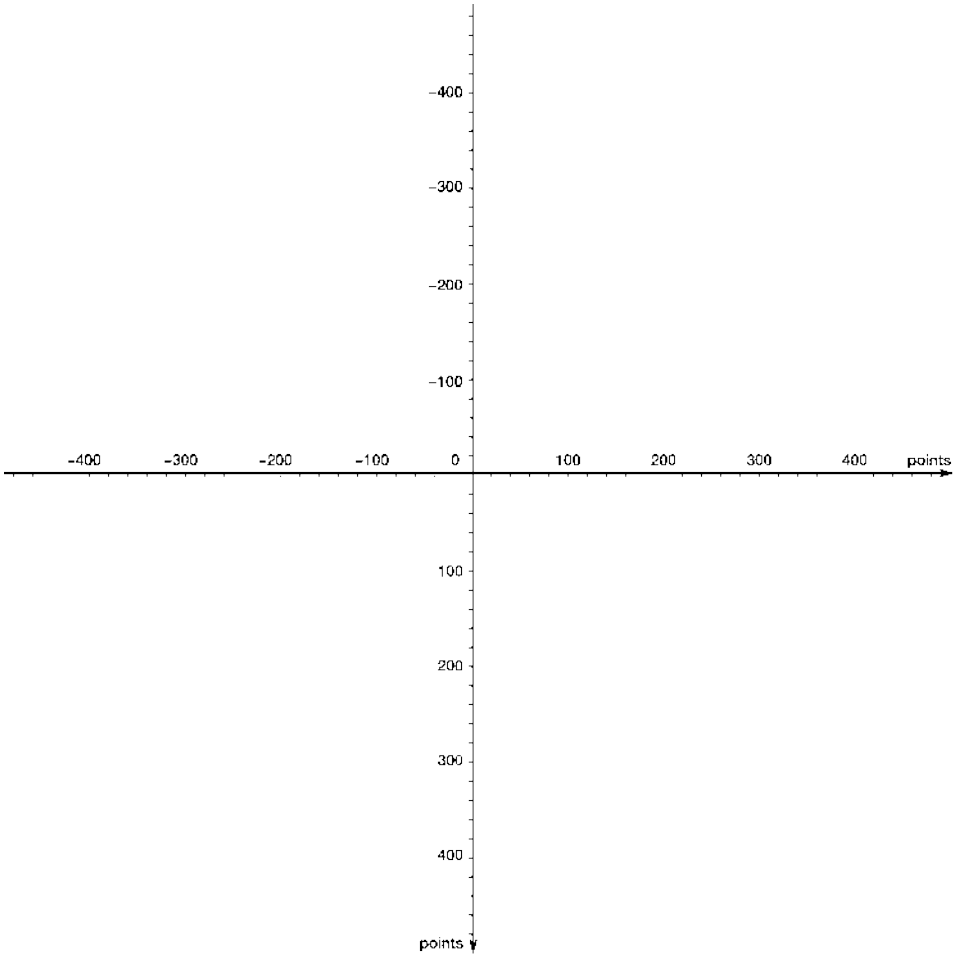
The Queue module implements multi-producer, multi-consumer queues. It is especially useful in threaded programming when information must be exchanged safely between multiple threads. There are three types of queues provides by queue module, Which are as following : 1. Queue 2. LifoQueue 3. PriorityQueue Exception which could be come: 1. Full (queue overflow) 2. Empty (queue underflow)

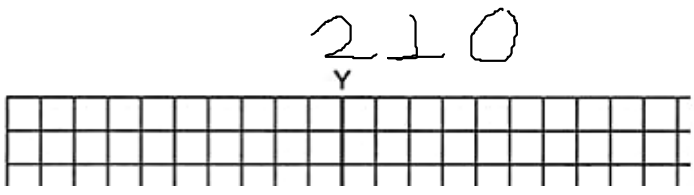
Turtle Module

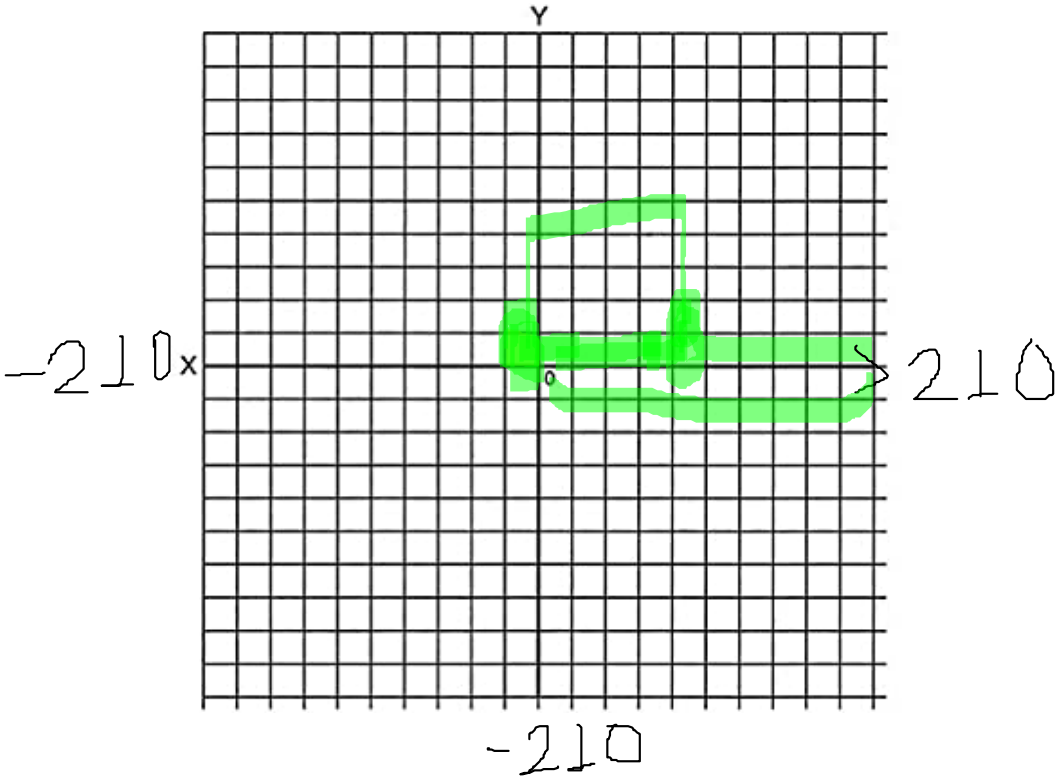
Saturday, April 25, 2020 7:20 PM

** Examples of some built-in functions **

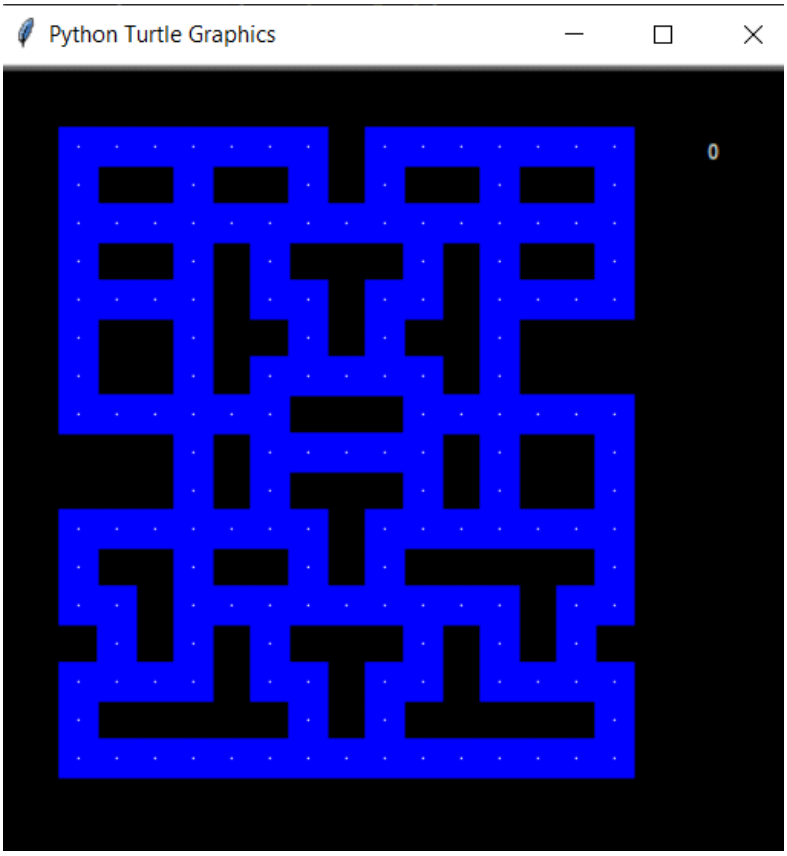
Screen Layout 2D





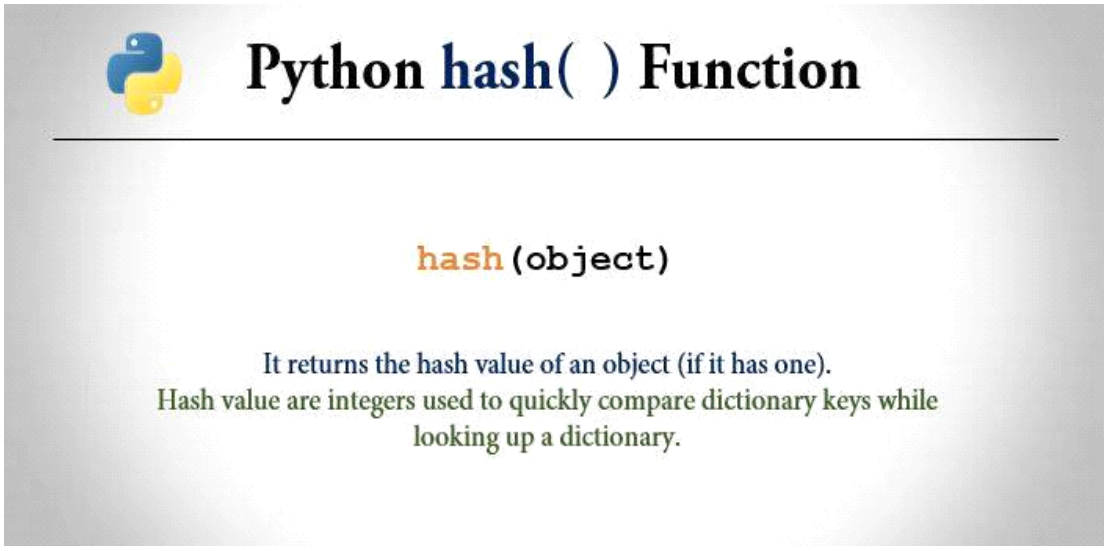


Pacman Game World


Python Turtle Graphics
0

Hash

Saturday, April 25, 2020 9:19 PM



Behind the scenes Python hash() function calls, __hash__() method internally to operate on different types of data types. __hash__() method is set by default for any object.

Framerate

Monday, April 27, 2020 5:29 PM

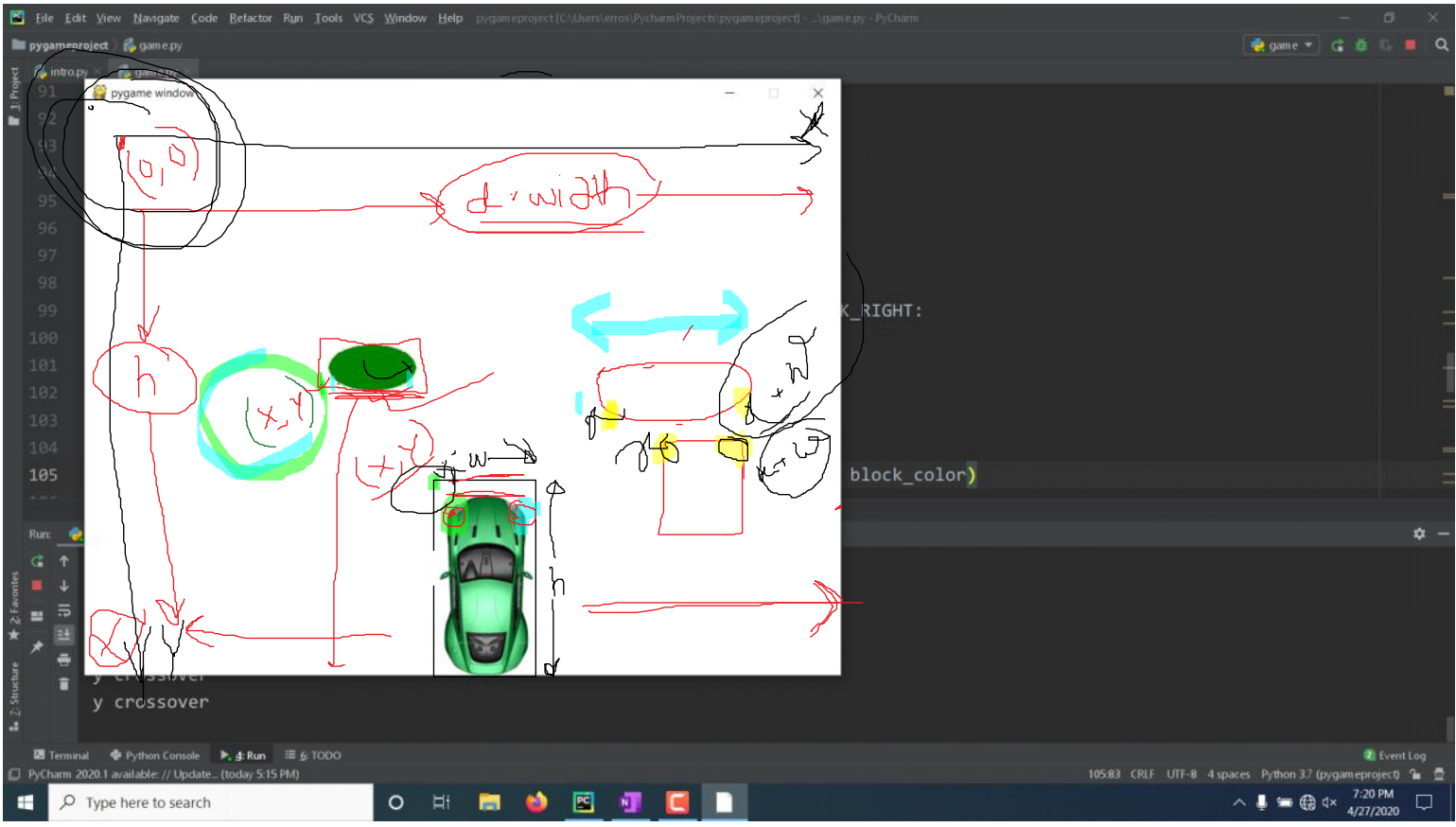
Frame rate is the speed at which those images are shown, or how fast you “flip” through the book and it’s usually expressed as “frames per second,” or FPS. Each image represents a frame, so if a video is captured and played back at 24fps, that means each second of video shows 24 distinct still images.

The speed at which they are shown tricks your brain into perceiving smooth motion.

Pygame Window

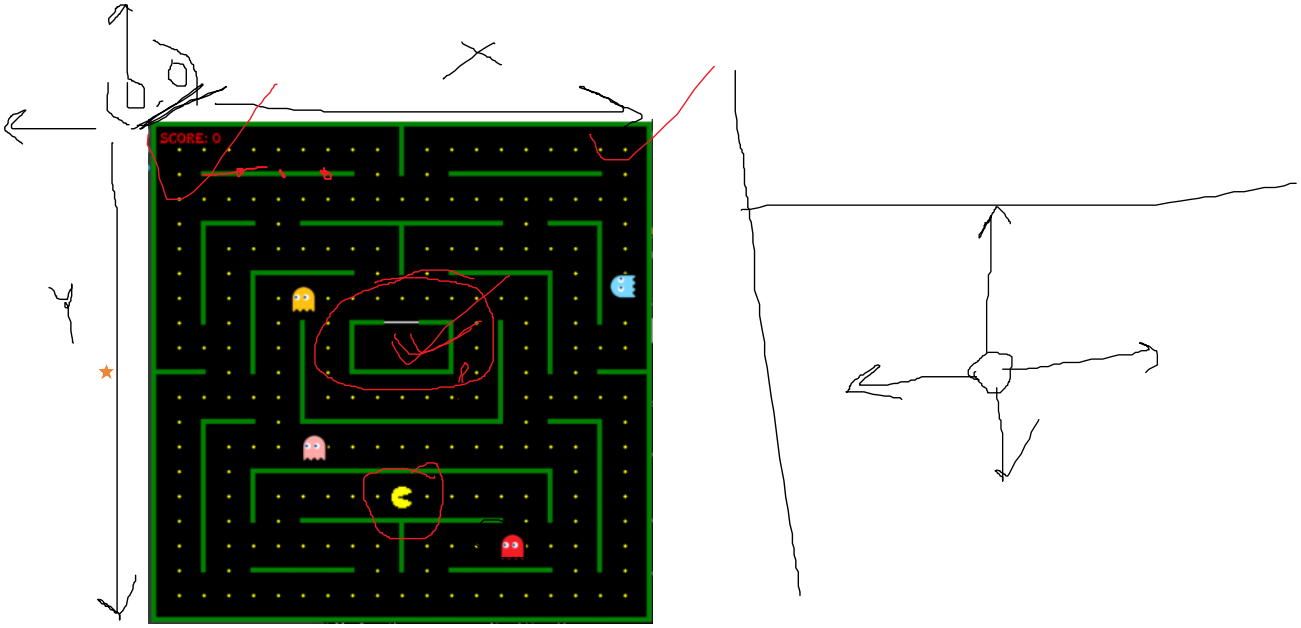
Car Race

Monday, April 27, 2020 7:29 PM



Pacman Using
Pygame

Tuesday, April 28, 2020 12:51 PM



1. Wall
2. Food
3. Player: Ghost, Pacman

Web Scraping

Wednesday, May 6, 2020 12:22 PM

Web scraping is an automated, programmatic process through which data can be constantly 'scraped' off webpages.
Also known as screen scraping or web harvesting, web scraping can provide instant data from any publicly accessible webpage. On some websites, web scraping may be illegal.

To extract data using web scraping with python, you need to follow these basic steps:

1. Find the URL that you want to scrape
2. Inspecting the Page
3. Find the data you want to extract
4. Write the code
5. Run the code and extract the data
6. Store the data in the required format

##Write a code##
we will be using the following libraries:

- **Selenium**: Selenium is a web testing library. It is used to automate browser activities.
- **BeautifulSoup**: Beautiful Soup is a Python package for parsing HTML and XML documents. It creates parse trees that is helpful to extract the data easily.

Selenium WebDriver is a web framework that permits you to execute cross-browser tests. This tool is used for automating web-ba

sed application testing to verify that it performs expectedly.
Selenium WebDriver allows you to choose a programming language of your choice to create test scripts. As discussed earlier, it is an advancement over Selenium RC to overcome a few limitations. Selenium WebDriver is not capable of handling window components, but this drawback can be overcome by using tools like Sikuli, Auto IT, etc.

Beautiful Soup is a Python library designed for quick turnaround projects like screen-scraping. Three features make it powerful:

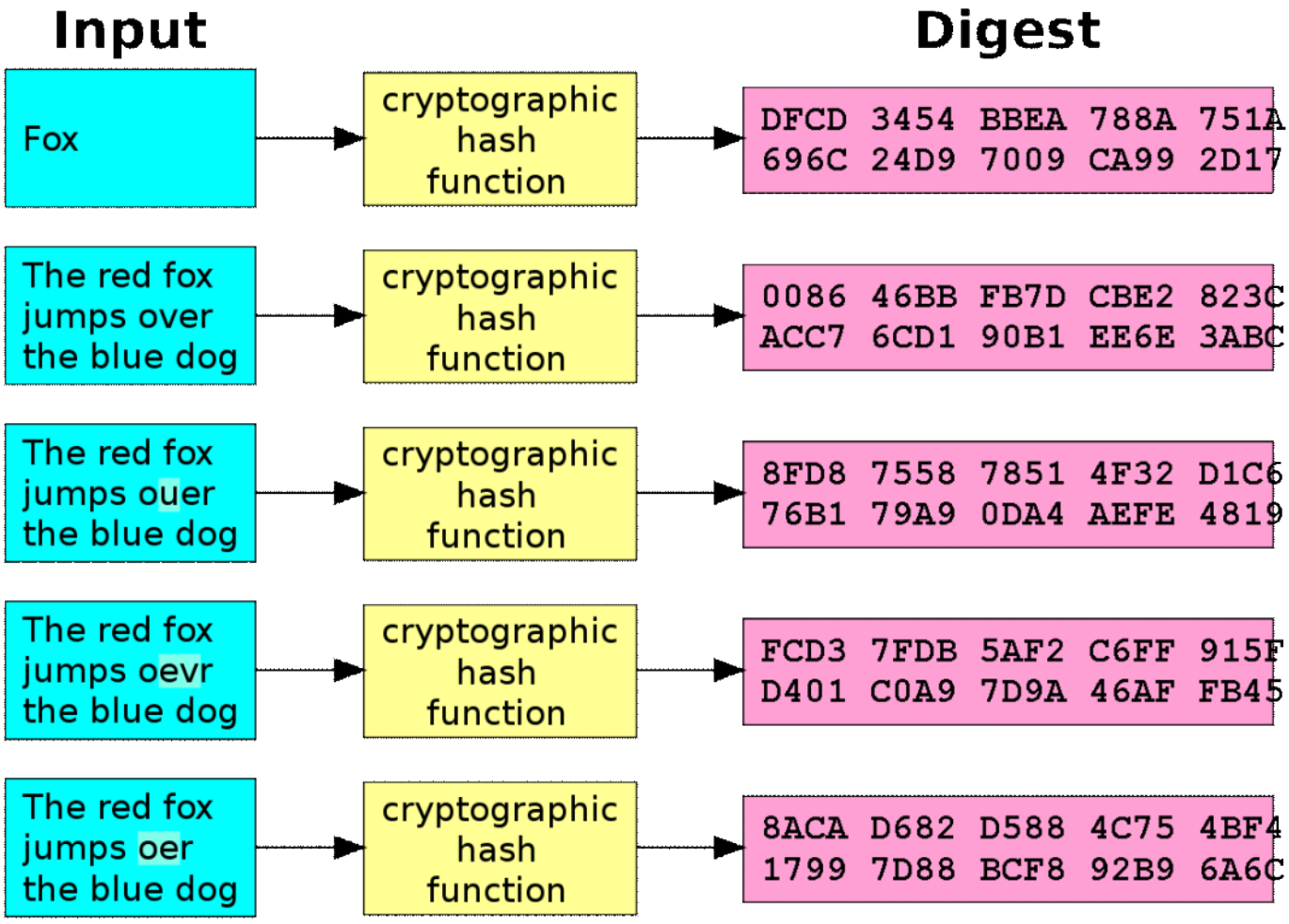
1. Beautiful Soup provides a few simple methods and Pythonic idioms for navigating, searching, and modifying a parse tree: a toolkit for dissecting a document and extracting what you need. It doesn't take much code to write an application

2. Beautiful Soup automatically converts incoming documents to Unicode and outgoing documents to UTF-8. You don't have to think about encodings, unless the document doesn't specify an encoding and Beautiful Soup can't detect one. Then you just have to specify the original encoding.
3. Beautiful Soup sits on top of popular Python parsers like lxml and html5lib, allowing you to try out different parsing strategies or trade speed for flexibility.

A *parser* is a software component that takes input data (frequently text) and builds a data structure – often some kind of parse tree, abstract syntax tree

Security and Cryptography

Wednesday, May 6, 2020 4:51 PM



Decoding:

1. Rainbow Table
2. Dictionary Attacks
3. Brute Force Attacks

SALT:

Output Hash ==> Hash (Password + SALT)

Password: Security

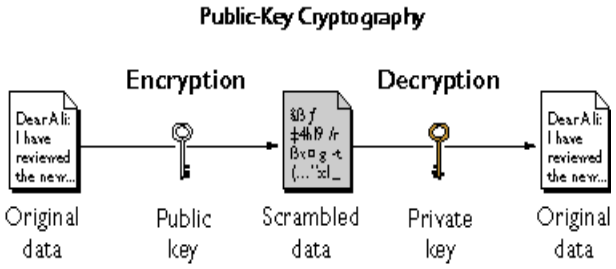
SALT:)#$*

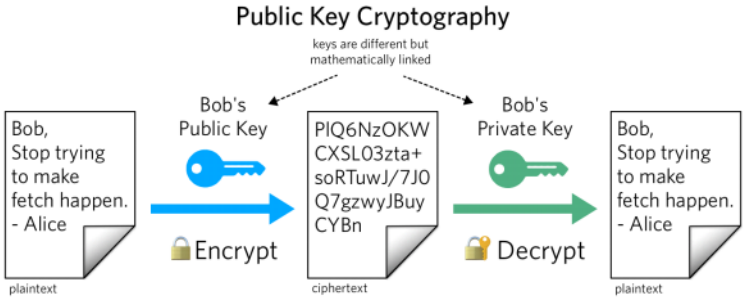
Output Hash: Security)#$*

Cryptography

Wednesday, May 6, 2020 9:52 PM

RSA can be used to create a message signature. A valid signature can only be generated with access to the private RSA key, validating on the other hand is possible with merely the corresponding public key. So as long as the other side knows your public key they can verify the message to be signed by you and unchanged - an approach used for email for example.
Currently, a third-party module like pycrypto or pycryptodome is required for this functionality.





Important libraries

```
from Crypto.Hash import SHA256
from Crypto.PublicKey import RSA
from Crypto.Signature import PKCS1_v1_5
```

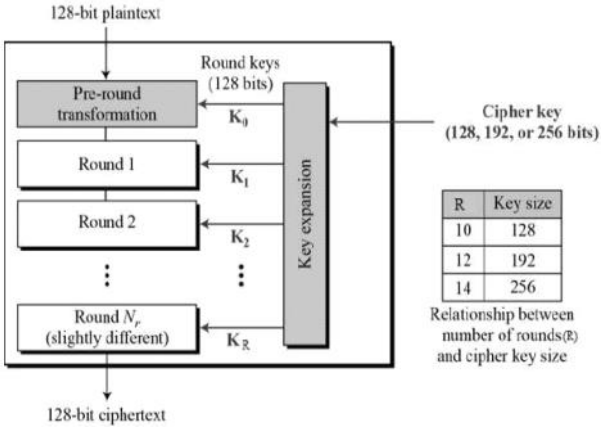
AES encryption

Wednesday, May 6, 2020 11:43 PM

A replacement for DES was needed as its key size was too small. With increasing computing power, it was considered vulnerable against exhaustive key search attack. Triple DES was designed to overcome this drawback but it was found slow.

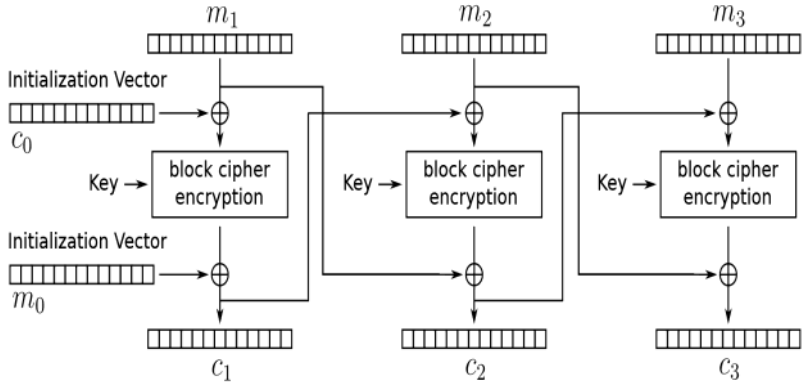
The features of AES are as follows –

- Symmetric key symmetric block cipher
- 128-bit data, 128/192/256-bit keys
- Stronger and faster than Triple-DES
- Provide full specification and design details



R	Key size
10	128
12	192
14	256

Relationship between number of rounds(R) and cipher key size



MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAuTivFqP7DiBJt7t1t7m5

Red => Initialization Vector (iv) (given size)

Black ==> Key

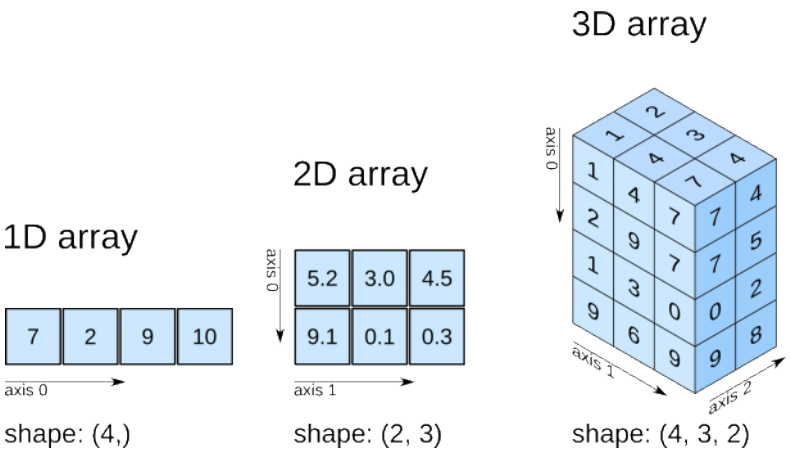
Salt =>)#&1

Encryption ==> Salt + (key , iv) ==> encrypt

NumPy is a Python package which stands for 'Numerical Python'. It is the core library for scientific computing, which contains a powerful n-dimensional array object, provide tools for integrating C, C++ etc. It is also useful in linear algebra, random number capability etc. NumPy array can also be used as an efficient multi-dimensional container for generic data. Now, let me tell you what exactly is a python numpy array.

Python NumPy Array v/s List

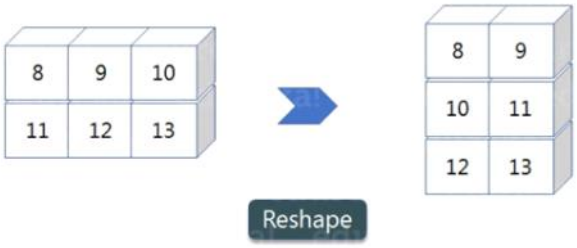
We use python numpy array instead of a list because of the below three reasons:

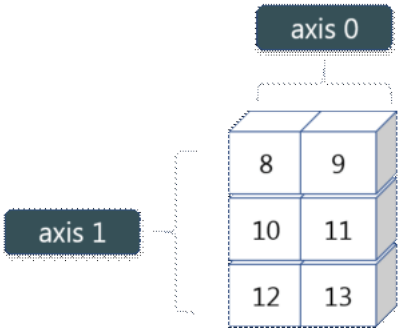


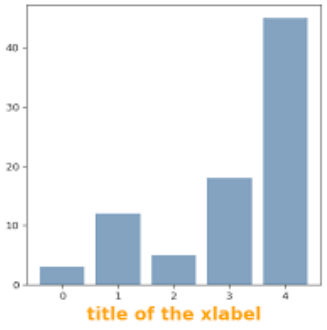
1. Less Memory
2. Fast
3. Convenient

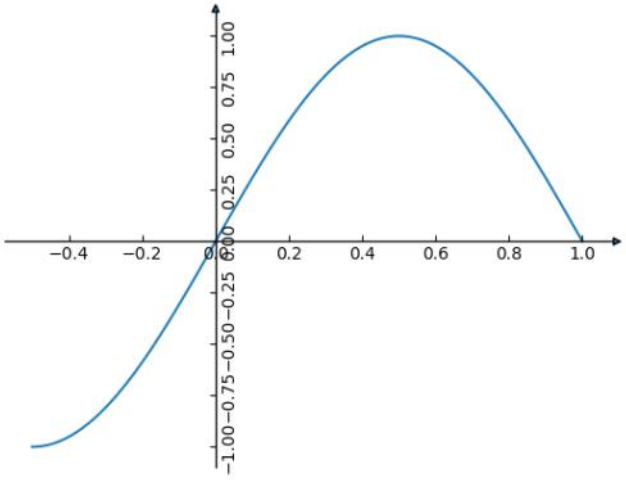
The very first reason to choose python numpy array is that it occupies less memory as compared to list. Then, it is pretty fast in terms of execution and at the same time it is very convenient to work with numpy. So these are the major advantages that python numpy array has over list. Don't worry, I am going to prove the above points one by one practically in PyCharm.

Reshaping:


8
9
10
11
12
13
Reshape
8
9
10
11
12
13


axis 0
axis 1
8
9
10
11
12
13


40
30
20
10
0
0
1
2
3
4
title of the xlabel


−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

Matplotlib.pyplot

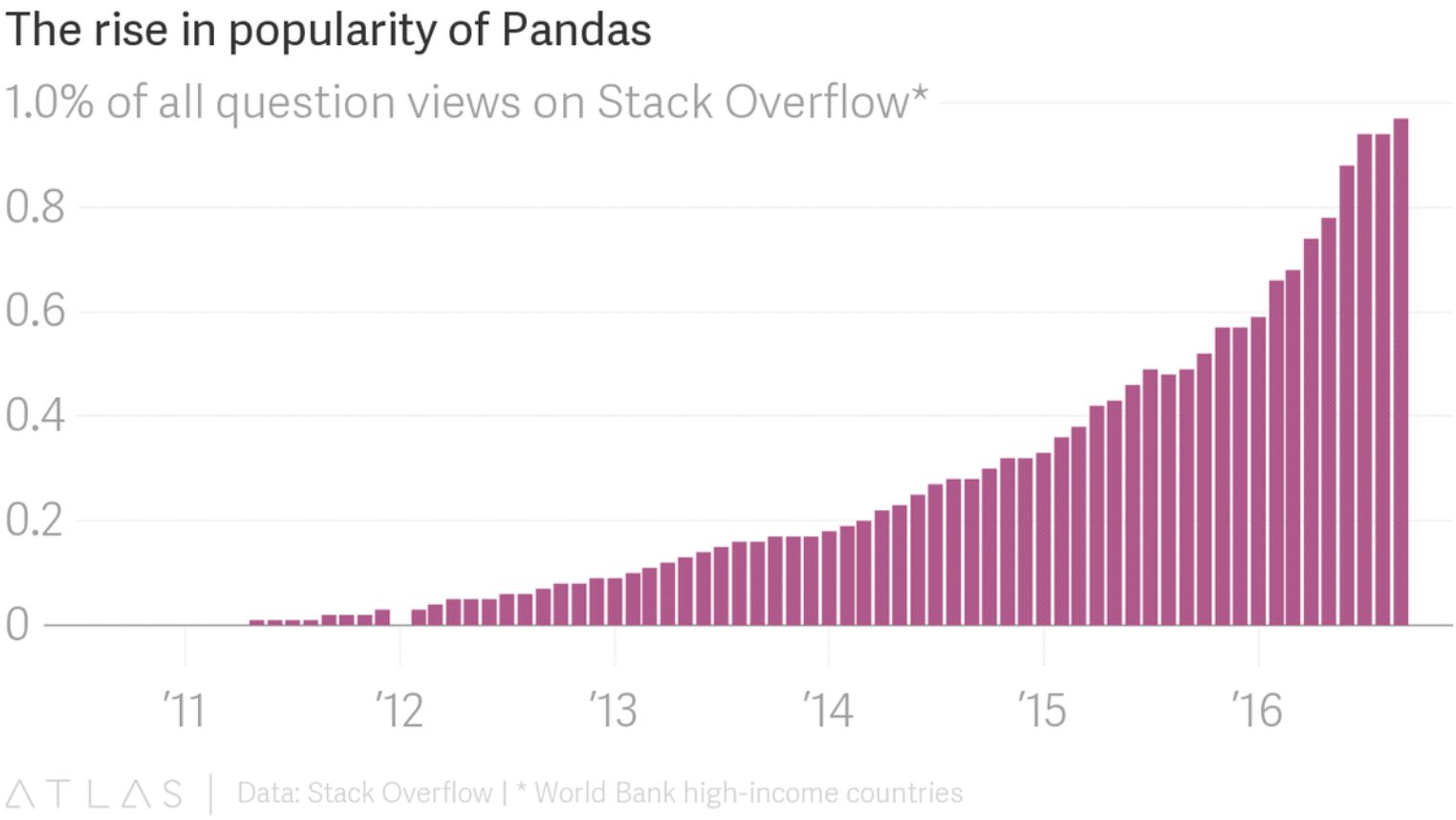
Thursday, May 7, 2020 5:08 PM

matplotlib.pyplot is a plotting library used for 2D graphics in python programming language. It can be used in python scripts, shell, web application servers and other graphical user interface toolkits.

Pandas

Thursday, May 7, 2020 10:00 PM

The *pandas* package is the most important tool at the disposal of Data Scientists and Analysts working in Python today. The powerful machine learning and glamorous visualization tools may get all the attention, but pandas is the backbone of most data projects.



What's Pandas for?

Pandas has so many uses that it might make sense to list the things it can't do instead of what it can do.

This tool is essentially your data's home. Through pandas, you get acquainted with your data by cleaning, transforming, and analyzing it. For example, say you want to explore a dataset stored in a CSV on your computer.

Pandas will extract the data from that CSV into a DataFrame — a table, basically — then let you do things like:

- Calculate statistics and answer questions about the data, like
  - What's the average, median, max, or min of each column?
  - Does column A correlate with column B?
  - What does the distribution of data in column C look like?
- Clean the data by doing things like removing missing values and filtering rows or columns by some criteria
- Visualize the data with help from Matplotlib. Plot bars, lines, histograms, bubbles, and more.
- Store the cleaned, transformed data back into a CSV, other file or database