## basics

REPL

read, eval, print, loop

.py (scripts)

.ipynb (python notebooks)

### IDEs (environments)

idle

pycharm

spyder

jupyter

eclipse

vscode (+ visual studio)

intelliJ

### names:

variable function names

can have alphabets numbers underscores

can start with alphabets underscores

### comments:

# single line

""" """ used as multiline comment

## Data Types:

### basic data types:

int integers

float floating point

complex (6+7i)

6+7j

strings

bool True

False

### data structs:

list

tuple

set

dict

# keywords:

del

True False None

if elif else

for

while

break continue

def return

global nonlocal

class import

with as

try except finally

## Functions:

### generic:

print()

type()

len()

### conversion (cast):

int()

float()

str()

complex()

list()

set()

tuple()

### iterative

range()

enumerate()

### for iterables

min()

max()

sum()

sorted()

any()

all()

filter()

map()

### lambda:

anonymous functions

## Operators:

### arithmetic :

+ add

- subtract

\* multiply

/ division

% modulo

(remainder)

= assignment

// floor division

\*\* power

not available:

++ --

### multiple assignment:

a = b = c = 33

### relational (conditional):

<

<=

>

>=

==

!=

### logical:

and

or

not

### bitwise:

^

<<

>>

&

|

### identity:

is

is not

### membership:

in

not in

## strings:

double or single quotes

indexed

negative index

slice

no IndexError

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| a | i | r | b | u | s |
| 0 | 1 | 2 | 3 | 4 | 5 |
| -6 | -5 | -4 | -3 | -2 | -1 |

immutable

functions:

capitalize, upper, lower, isupper, islower

isalpha, isalnum, isdigit

count, index, rindex

replace, endswith

split , splitlines

…….

### escape sequences:

\n new line

\b backspace

\t tab

\v vertical tab

\r carriage return

\\ \

### string format:

%

.format()

f strings

r strings (raw)

## list:

nay kind of data (heterogenous)

indexed

negative index

slice

no IndexError

step

mutable

nested

references

functions: count, index

sort, sort(reverse=True)

reverse

pop, remove

insert, append, extend

copy

## tuple:

any kind of data (heterogenous)

indexed

negative index

slice

no IndexError

step

immutable

nested

functions: count, index

## set:

no duplicates (only unique)

unordered

any kind of immutable data (heterogenous)

~~index~~

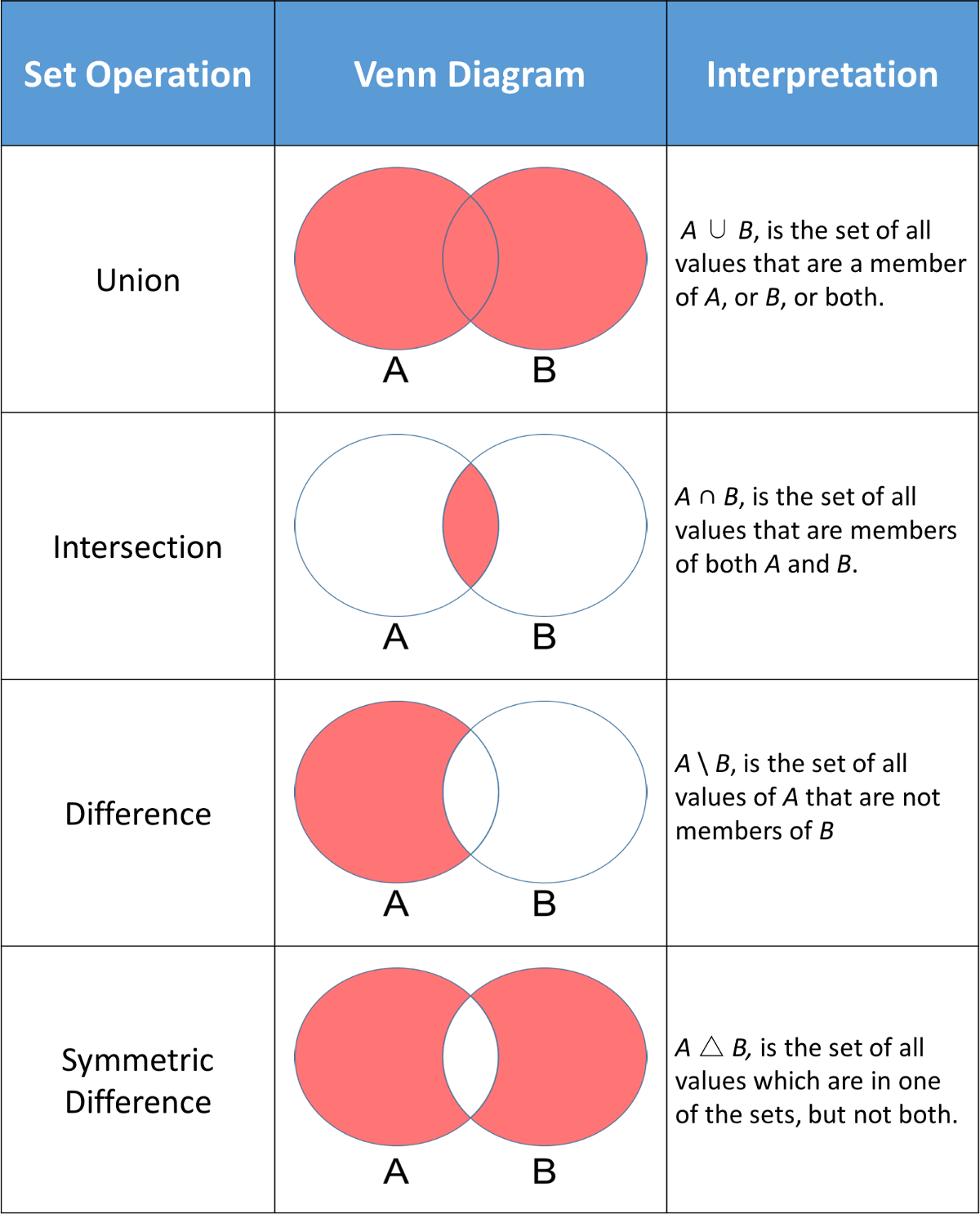
sets by themselves mutable

members should be immutable

functions: add, clear,

……

…..



## dict:

key-value pairs

indexed using keys

unordered

keys:

immutable

unique

values:

anything

dict by themselves are mutable

nesting

### comprehensions:

list

dictionary

## flow control:

if elif else

for else

while else

break

continue

range()

### false:

condition being false

False

None

0

0.0

'' [] {}

## custom functions:

def

any number of arguments can be passed

(indefinite)

any number of arguments can be returned

default : None

default values

name arguments

variable number of arguments(args, kwargs)

nesting of functions

## Memory:

# never use id in programming

|  |  |  |  |
| --- | --- | --- | --- |
| 205680 | 20 | varx, varz |  |
| 205676 | 25 | vary |  |
| 205672 | (23, 56, 78) | tupa |  |
| 200002 | [&205680, &205676, | lista, listb | [20, 25, ...] |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

varz = 20

## modules:

1. import syed

syed.blue()

syed.yellow()

syed.green()

1. from syed import blue, yellow

blue()

yellow()

~~green()~~

1. from syed import \*

blue()

yellow()

green()

1. import syed as s

s.blue()

s.green()

s.yellow()

1. from syed import blue as b

b()

os

sys

time

## Object Oriented Programming

class

\_\_init\_\_

\_\_del\_\_

\_\_private encapsulation

child (parent) inheritance

operator overloading polymorphism

## files:

fa = open()

fa.read()

fa.readline()

fa.readlines()

fa.readable()

fa.writable()

fa.close()

fa.closed

fa.tell()

fa.seek()

with open() as

r read

w write (deletes all old data)

(reats like a new file)

a append

r+ read & write

b binary

## Regular Expressions:

### Meta Characters:

. single character

(except \n)

^ start of string

$ end of string (line)

\* zero or more occurrences

+ one or more occurrences

? zero or one occurrences

[ ] set of characters

[a-z] all lowercase chars

[0-9] any numeric string

[0-39] = [01239]

[^ ] invert of pattern

{x,y} at least x and amx y repetitions

| or

( ) grouping

### Special Sequence :

\s white space

space \n \t \v \r

\S invert of \s

\d any numeric

\D any non numeric

\w alphanumeric

alphabets numbers underscore

[a-zA-Z0-9\_]

### re functions:

match

findall

split

sub

subn

## Exception Handling

try except

finally else

exception occurs, is handled:

except

finally

code continues

exception occurs, is NOT handled:

finally

code breaks

exception does not occurs

else

finally

code continues

## NumPY:

NUMerical PYthon

mathematical & scientific computations

linalg (linear algebra)

direct C language based

faster

designed to work with numbers

can use other kinds of data too

## pandas:

parked topics:

security

timestamp of a file

sample use of classmethods etc in classes & modules

migration issues (unicode etc.)

memory libraries

Data types:

text

gsheets

csv

json

xml

doc

SQL