## Basics

### installation

python.org

idle

linux command line (ubuntu)

sudo apt install python3

sudo apt install python3-pip

### editors & IDEs

idle

vscode

pycharm

visual studio

jupyter

spyder

vi & python command

### REPL

read

eval

print

loop

### features

ease of learning

most domains

libraries

resources

python 3

### everything is an object

### keywords

del None

True False

and or not

is in

if elif else

while for

break continue

pass

### comments

# single line comment

docstring used for multiline comments

## Data Types

int

float

str

complex

6 + 7j

### bool

True

False

### what is False

0

0.0

{ }

[ ]

''

None

False

## 

## Functions

### general

print()

type()

help()

dir()

### sequence (iterable)

len()

sum()

min()

max()

sorted()

reversed()

any()

all()

enumerate()

zip()

map()

filter()

### anonymous functions (lambda)

lambda

### conversion (cast)

int()

float()

str()

list()

set()

tuple()

## operators

### arithmetic

+

-

/

\*

% modulo

// floor division

\*\* power of

= assignment

no increment/decrement

### comparison (relational)

<

>

<=

>=

==

!=

### logical

and

or

not

### membership

in

not in

### identity

is

is not

## str

text data

double quotes or single quotes

ordered

index

out of bounds (Index Error)

negative index

slice

no out of bound error

step

| n | o | o | b | s |
| --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 |
| -5 | -4 | -3 | -2 | -1 |

immutable

### str functions

| capitalize() | Returns a copy of the string with its first character capitalized and the rest lowercased. |
| --- | --- |
| casefold() | Returns a casefolded copy of the string. Casefolded strings may be used for caseless matching. |
| center(width[, fillchar]) | Returns the string centered in a string of length width. Padding can be done using the specified fillchar (the default padding uses an ASCII space). The original string is returned if width is less than or equal to len(s) |
| count(sub[, start[, end]]) | Returns the number of non-overlapping occurrences of substring (sub) in the range [start, end]. Optional arguments start and end are interpreted as in slice notation. |
| endswith(suffix[, start[, end]]) | Returns True if the string ends with the specified suffix, otherwise it returns False. suffix can also be a tuple of suffixes. When the (optional) start argument is provided, the test begins at that position. With optional end, the test stops comparing at that position. |
| expandtabs(tabsize=8) | Returns a copy of the string where all tab characters are replaced by one or more spaces, depending on the current column and the given tab size. T |
| find(sub[, start[, end]]) | Returns the lowest index in the string where substring sub is found within the slice s[start:end]. Optional arguments start and end are interpreted as in slice notation. Returns -1 if sub is not found. |
| format(\*args, \*\*kwargs) | Performs a string formatting operation. The string on which this method is called can contain literal text or replacement fields delimited by braces {}. |
| index(sub[, start[, end]]) | Like find() but raises a ValueError when the substring is not found (find() returns -1 when the substring isn't found). |
| isalnum() | Returns True if all characters in the string are alphanumeric and there is at least one character. Returns False otherwise. |
| isalpha() | Returns True if all characters in the string are alphabetic and there is at least one character. Returns False otherwise. |
| isdigit() | Returns True if all characters in the string are digits and there is at least one character. Returns False otherwise. |
| islower() | Returns True if all cased characters in the string are lowercase and there is at least one cased character. Returns False otherwise. |
| isnumeric() | Returns True if all characters in the string are numeric characters, and there is at least one character. Returns False otherwise. |
| isspace() | Returns True if there are only whitespace characters in the string and there is at least one character. Returns False otherwise.  Whitespace characters are space, tab, new line, vertical tab etc. |
| istitle() | Returns True if the string is a titlecased string and there is at least one character (for example uppercase characters may only follow uncased characters and lowercase characters only cased ones). Returns False otherwise. |
| isupper() | Returns True if all cased characters in the string are uppercase and there is at least one cased character. Returns False otherwise. |
| join(iterable) | will be discussed during the session |
| lower() | Returns a copy of the string with all the cased characters converted to lowercase. |
| partition(sep) | Splits the string at the first occurrence of sep, and returns a 3-tuple containing the part before the separator, the separator itself, and the part after the separator. If the separator is not found, it returns a 3-tuple containing the string itself, followed by two empty strings. |
| replace(old, new[, count]) | Returns a copy of the string with all occurrences of substring old replaced by new. If the optional argument count is provided, only the first count occurrences are replaced. For example, if count is 3, only the first 3 occurrences are replaced. |
| rfind(sub[, start[, end]]) | Returns the highest index in the string where substring sub is found, such that sub is contained within s[start:end]. Optional arguments start and end are interpreted as in slice notation. This method returns -1 on failure. |
| rindex(sub[, start[, end]]) | Like rfind() but raises ValueError when the substring sub is not found. |
| rpartition(sep) | Splits the string at the last occurrence of sep, and returns a 3-tuple containing the part before the separator, the separator itself, and the part after the separator. If the separator is not found, it returns a 3-tuple containing the string itself, followed by two empty strings. |
| rsplit(sep=None, maxsplit=-1) | Returns a list of the words in the string, using sep as the delimiter string. If maxsplit is given, at most maxsplit splits are done, the rightmost ones. If sep is not specified or is set to None, any whitespace string is a separator. |
| rstrip([chars]) | Return a copy of the string with trailing characters removed. The chars argument is a string specifying the set of characters to be removed. If omitted or set to None, the chars argument defaults to removing whitespace. |
| split(sep) | Returns a list of the words in the string, using sep as the delimiter string. |
| splitlines() | Returns a list of the lines in the string, breaking at line boundaries. |
| startswith(prefix[, start[, end]]) | Returns True if the string starts with the specified prefix, otherwise it returns False. prefix can also be a tuple of prefixes. When the (optional) start argument is provided, the test begins at that position. With optional end, the test stops comparing at that position. |
| strip([chars]) | Returns a copy of the string with leading and trailing characters removed. The chars argument is a string specifying the set of characters to be removed. If omitted or set to None, the chars argument defaults to removing whitespace. |
| swapcase() | Returns a copy of the string with uppercase characters converted to lowercase and vice versa. |
| title() | Returns a title-cased version of the string. Title case is where words start with an uppercase character and the remaining characters are lowercase. |
| translate(table) | will be discussed in the intermediate session |
| upper() | Returns a copy of the string with all the cased characters converted to uppercase. |

### escape sequences

\n new line

\t tab

\b backspace

\\ \

\r, \v

### docstrings

triple quotes

multiple lines

used as multiline comment

## list

any & all kinds of data

allows duplicates

ordered

index

out of bounds (Index Error)

negative index

slice

no out of bound error

step

mutable

references

nesting

### list functions

insert, append

remove, pop, clear

extend, append,

reverse, sort

copy

count, index

## tuple

any & all kinds of data

allows duplicates

ordered

index

out of bounds (Index Error)

negative index

slice

no out of bound error

step

immutable

nesting

### tuple functions

count, index

## set

members:

immutable

(strings, numbers, tuples)

no duplicates

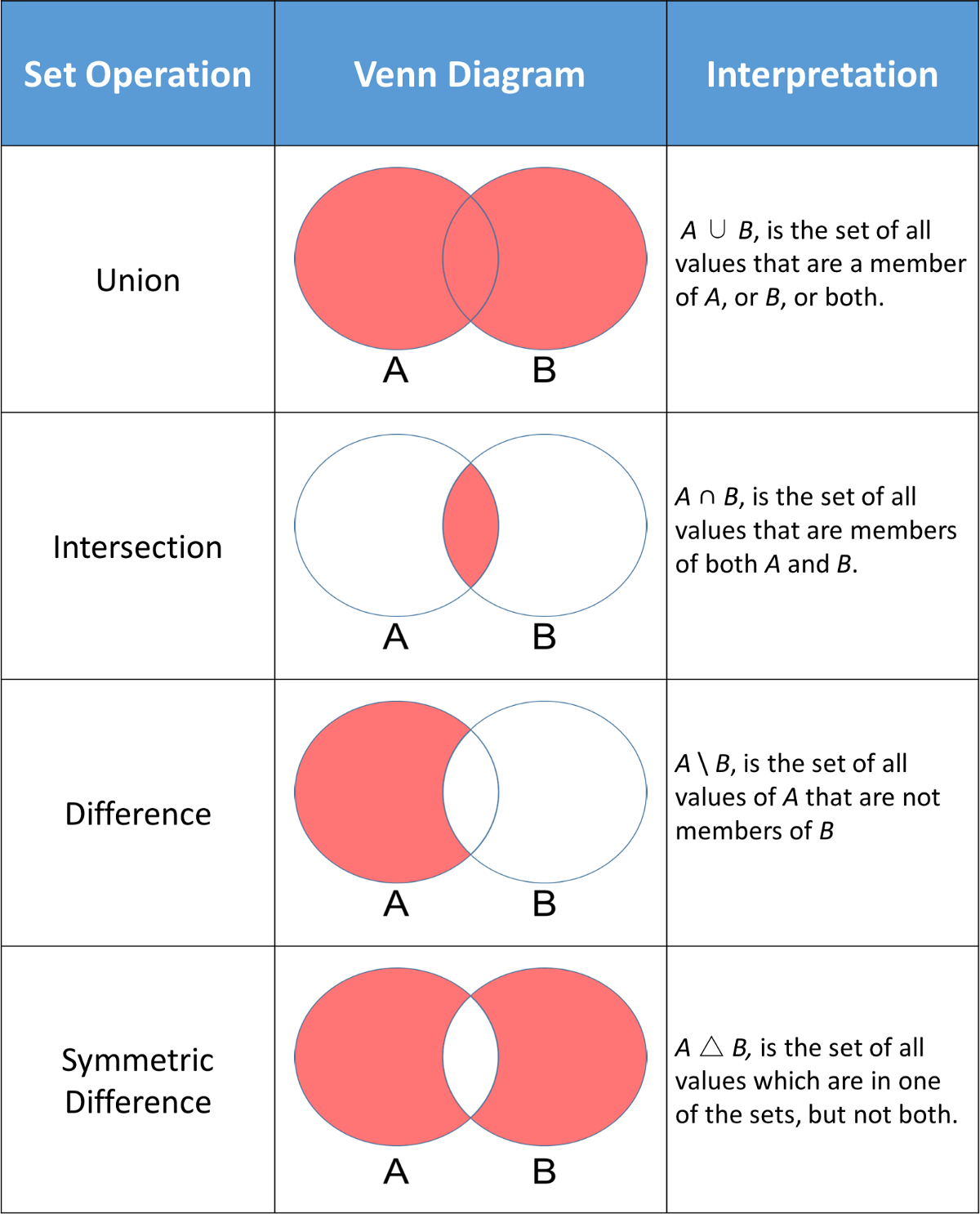
unordered

no indexing, no slicing

set by itself is mutable

(can add/remove members to set)

### set operations



| union

& intersection

- difference

^ symmetric difference

### set functions

… ….

## dict

key : value

ordered (3.7 +)

keys:

immutable,

(strings, numbers, tuples)

only unique

values:

anything

dict by itself is mutable

(can add/remove key:value pairs to dict)

nesting

another dictionary as value

### dict functions

clear() Removes all items from the dictionary.

copy() Returns a copy of the dictionary.

fromkeys(seq[, v]) Returns a new dictionary with keys from seq and value

equal to v (defaults to None).

get(key[,d]) Returns the value of the key.

If the key does not exist, returns d (defaults to None).

items() Return a new object of the dictionary's items in (key, value) format.

keys() Returns a new object of the dictionary's keys.

pop(key[,d]) Removes the item with the key and returns its value

or d if key is not found.

If d is not provided and the key is not found, it raises KeyError.

popitem() Removes and returns an arbitrary item (key, value).

setdefault(key[,d]) Returns the corresponding value if the key is in the dictionary.

If not, inserts the key with a value of d and returns d

(defaults to None).

update([other]) Updates the dictionary with the key/value pairs from other,

overwriting existing keys.

values() Returns a new object of the dictionary's values

## flow control

### if else

if

elif

else

### match case

3.10+

### while

no do while

break, continue

while-else

### for

for with sequences

for with range

break, continue

for-else

## functions (user defined)

### arguments or parameters

any number

default values (arguments)

right most

named arguments

### arbitrary arguments

\*args

\*\*kwargs

### return

default is None

any number of values

## modules

### ways of import

1. import colour

colour.blue()

colour.yellow()

colour.green()

colour.red

1. import colour as c

c.blue()

c.yellow()

c.green()

c.red

1. from colour import blue, yellow

blue()

yellow()

~~green()~~

~~red()~~

1. from colour import \*

blue()

yellow()

green()

red()

1. from colour import blue as b

b()

## files

fa = open()

fa.close()

fa.read(n)

fa.readline()

fa.readlines()

fa.readable()

fa.writable()

fa.write()

fa.tell()

fa.seek()

fa.closed

extras:

json

python docs

excel

pandas