Python

Leonardo Saurwein - lsaurwein@student.ethz.ch

Version: March 21, 2022

Python 3.10

General

Operators

x+y b: sum of x and y a - b: minus a * b: times

a ** n: a to the power of 7

a / b: division, return a float

a // b: floor division discards the fractional part

a **%** b: the operator % returns the remainder of the division

_ :console, the last printed expression is assigned to the variable _

abs(x): absolute value or magnitude of x

int(x): x converted to integer

float(x): x converted to floating point

complex(re, im): complex number, re: real, im: imaginary

c.conjugate(): conjugate of the complex number c bm

Bitwise Operations

TODO

Boolean operators

x or y: if x is false, then y, else x

x and y: if x is false, then x, else y

not x: if x is false, then True, else False

Comparisons

<: strictly less than

<=: less than or equal

>: strictly greater than

>=: greater than or equal

==: equal

! =: not equal

is: object identity

is not: negated object identity

Strings

s = 'a' or "a": single quotes strings

'doesn\'t': use \' to escape the single quote

"doesn't": or use double quotes instead

n: go to the next line **t**: tab space **r**: carriage return

r"...": raw strings

"""...""": string literals on multiple lines, use \ to prevent \n

+ and *: sum or multiply strings

a[n]: to access n index of a (start from 0)

a[-k]: to access n-k index of a (start from n+1)

a[i:j]: range from i to j, leave black to get first or last

strings are immutable (don't support index assignment)

Methods (all return a copy):

len(a): return the length of the string

s.capitalize(): first character capitalized and the rest lowercased

s.casefold(): casefolded copy of the string. **s.center(w, c)**: centered in a string of length w, fill with c

s.count(i): count ammount of substring i in str

s.encode(): encode the string, default: utf-8

s.endswith(s): true if s end with s, false otherwise

s.startswith(s): true if s start with s, false otherwis

s.expandtabs(n): expand the tab (\t)

s.find(i): return smallest index of i in s, -1 if not found

s.index(i): like find but raise an error

s.isalnum(): true if alphanumeric

s.isalpha(): true if all alphabetic

s.isascii(): true if ascii

s.isdecimal(): true if all decimal

s.isdigit(): true if all digit

s.islower(): true if all lowercase

s.lower(): all cased characters converted to lowercase

s.isupper(): true if all uppercase

s.upper(): all cased characters converted to uppercase

s.isnumeric(): true if all numeric character

s.isspace(): true if only whitespace characters

s.istitle(): true if title first upper rest lower

s.title(): convert s to a title

s.strip([c]): strip c from left and right, black by deafult

s.lstrip([c]): –¿ remove c from left, blank by default

s.rstrip([c]): ¡– remove c from right, blank by deafult

s.removeprefix(i): remove prefix i

s.removesuffix(i): remove sufflix i

s.replace(old, new, count): replace all old with new

s.split(sep): split on sep return a list

s.splitlines(): split on **n** return a list

s.swapcase(): convert upper to lower and viceversa

Tuples and Sets

TODO

Lists

l = [a,b,c]: list, support index like strings

l[:]: return a copy of l

l[n] = k: set n index to k, work also with a slice (l[i:j])

[i for i in a]: concise way to create lists

del l[n]: delete n index, work also with interval

Data structure:

len(l): return the length of the list

min(l): return smallest item of s

max(l): return largest item of s

l.append(i): add i to the end of the list (a[len(a):] = [x])

l.extend(it): appending all the items (a[len(a):] = it)

l.insert(i, x): insert at i, x all the value shift

l.remove(x): remove first item that equal x. error if don't exist

l.pop([i]): remove index i, if blank the last. return the value

l.clear(): remove all the elements (del a[:])

l.index(x): return index of x. error if don't exist

l.count(x): return the ammount of x

l.sort(): sort the list

l.reverse(): flip the list ([::-1])

l.copy(): return a copy of the list ([:])

Dictionaries

Flow Tools

if Statements

Use:

if (condition):

elif (condition):

else:

Classes Special Attributes

__**dict**__: A dictionary or other mapping

__**class**__: The class to which a class instance belongs

__**bases**__: The tuple of base classes of a class object.

__**name**__: The name of the class, function, method, descriptor, or generator instance.

____:

____:

____:

____:

____:

____:

____:

____:

____:

Libraries

Basics

Os

Math

Random

Statistics

Matplotlib

Numpy

Pandas

Datetime

Timeit

Pygame

Threading

Requests

Flask