Python 3 Ściąga

Najnowsza wersja dostępna na stronie: https://perso.limsi.fr/pointal/python:memento

```
Rodzaje kontenerów
integer, float, boolean, string, bytes
                                          Typy proste

    uporządkowane sekwencje, szybki dostęp do indeksów, powtarzalne wartości

                                                                        list [1,5,9]
                                                                                                 ["x",11,8.9]
                                                                                                                              ["mot"]
                                                                                                                                                    []
 int 783 0 -192 l. calkowita zero
                             0b010 0o642 0xF3
                                     ósemkowy
heksadecymalny
                  zero
                                                                     *tuple (1,5,9)
                                                                                                   11,"y",7.4
                                                                                                                              ("mot",)
                                                                                                                                                    ()
float 9.23 0.0 - L. zpiennopr_ecinkowa bool True False
                          -1.7e-6
                                                             Wartości niemodyfikowalne (niezmienne)

    wyrażenie z tylko przecinkami → tuple

                                  ×10<sup>-6</sup>
                                                                     str bytes (uporządkowane sekwencje znaków / bajtów)
                                                                                                                                                    min.
typ logiczny
Str "One\nTwo"
lańcuch znaków
                                  wielowierszowy string:
                                                                                                                                                  b""
                                                             • kontenery na klucze, brak kolejności a priori, szybki dostęp do kluczy, każdy klucz jest unikalny
                                     """X\tY\tZ
           Przejście do nowej linii
                                                            dictionary dict {"key":"value"}
                                                                                                               dict(a=3,b=4,k="v")
                                                                                                                                                    { }
           'I<u>\</u>m'
                                     1\t2\t3"""
           Wyświetlenie znaku 1
                                                            (pary klucz/wartość) {1:"one", 3:"three", 2:"two", 3.14:"π"}
                                     Wyświetlenie tabulacji
bytes b"toto\xfe\775"
                                                                          set {"key1","key2"}
                                                            collection
                                                                                                                {1,9,3,0}
                                                                                                                                               set()
            szesnastkowy ósemkowy
                                           niezmienne
                                                             klucze = wartości, które można mieszać (typy podstawowe, niezmienne, itd) frozenset niezmienny zestaw
                                 Identyfikatory
                                                  int("15") \rightarrow 15
                                                                                                        type (expression)
```

```
pusty
dla nazw zmiennych.
                                                                                                                                      Konwersje typów
funkcii, modułów, klas ...
                                                 int("3f",16) \rightarrow 63
                                                                                         określenie podstawy liczby całkowitej w drugim parametrze
a...zA...Z_ po którym mogą być a...zA...Z_0...9
                                                 int(15.56) \rightarrow 15
                                                                                         obcina część dziesiętną

    znaki diakrytyczne dozwolone, ale należy ich unikać

                                                 float("-11.24e8") \rightarrow -1124000000.0

    zabronione słowa kluczowe w języku

                                                 round (15.56,1) → 15.6 zaokraglenie do 1 miejsca po przecinku (0 liczba dziesiętna → liczba całkowita)

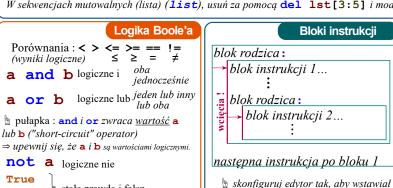
    rozróżnianie małych/DUŻYCH liter

                                                 bool (x) False dla zera x, pusty kontener x, None lub False x; True dla innych x
       © a toto x7 y_max BigOne

⊗ 8y and for

                                                 \operatorname{\mathtt{str}}(\mathbf{x}) \to \text{"..."} reprezentuje łańcuch \mathbf{x} do wyświetlenia (por. formatowanie z tylu)
                                                 chr(64) \rightarrow '@' ord('@') \rightarrow 64
                                                                                                    kod \leftrightarrow znak
                Przypisania do zmiennych
                                                 repr(x) → "..." dosłowny ciąg reprezentujący x
 ½ zadanie ⇔ wiązanie nazwy z wartością
                                                 bytes([72,9,64]) \rightarrow b'H\t@'
 1) ocena wartości wyrażenia po prawej stronie
 2) przypisanie w kolejności z nazwami po lewej stronie
                                                 list("abc") \rightarrow ['a','b','c']
x=1.2+8+\sin(y)
                                                 dict([(3,"three"),(1,"one")]) \rightarrow \{1:'one',3:'three'\}
a=b=c=0 przypisanie do tej samej wartości
                                                 set(["one","two"]) -> {'one','two'}
y,z,r=9.2,-7.6,0 równoczesne przypisania
                                                 separator \ \mathtt{str} \ i \ sekwencja \ \mathtt{str} \ 	o złożony \ \mathtt{str}
a,b=b,a zamiana wartości
                                                      ":".join(['toto','12','pswd']) \rightarrow 'toto:12:pswd'"
a, *b=seq \ \  sekwencja rozpakowywania w
                                                  str wycięcie białych znaków → list utworzona z str
*a,b=seq ∫ pozycji i w listach
                                                      "words with spaces".split() \rightarrow ['words','with','spaces']
                                            and
x+=3
           inkrementacja \Leftrightarrow x=x+3
                                                 \mathtt{str} rozdzielenie na separatorze \mathtt{str} \to \mathtt{list} utworzona z \mathtt{str}
x=2
           dekrementacja \Leftrightarrow x=x-2
                                            /=
                                                      "1,4,8,2".split(",") \rightarrow ['1','4','8','2']
x=None «niezdefiniowany» stała wartość
                                            용=
                                                 sekwencja\ jednego\ typu \rightarrow \texttt{list}\ innego\ typu
del x
           usuwa nazwe x
                                                      [int(x) for x in ('1','29','-3')] \rightarrow [1,29,-3]
```

```
dla list, krotek, łańcuchów znaków, bajtów...
                                                                                                         Indeksowanie kontenerów sekwencji
                                                  -1
                                                            Liczba elementów
                                                                                    Indywidualny dostęp do elementów przez lst[index]
                    -5
                           -4
                                   -3
                                          -2
  indeks uiemny
                    0
                                           3
 indeks dodatni
                            1
                                    2
                                                   4
                                                                                     lst[0]→10
                                                            len (1st) \rightarrow 5
                                                                                                        ⇒ pierwszy
                                                                                                                         lst[1] \rightarrow 20
          lst=[10]
                          20,
                                   30,
                                          40
                                                  501
                                                                                     lst[-1] \rightarrow 50 \Rightarrow ostatni
                                                                                                                         lst[-2] \rightarrow 40
                                                               index od 0
 wycinek dodatni 0
                        1
                                       3
                                               4
                                                                                     Na modyfikowalnych sekwencjach (list),
                                                              (tutaj od 0 do 4)
 wycinek ujemny -5
                                      -2
                                                                                     usuwanie za pomocą del lst[3] i
                                                                                     według przypisania lst[4]=25
Dostęp do sekwencji podrzędnych za pośrednictwem <code>lst[start slice:end slice:step]</code>
                                                                                                               lst[:3] \rightarrow [10,20,30]
lst[:-1] \rightarrow [10,20,30,40] lst[::-1] \rightarrow [50,40,30,20,10] lst[1:3] \rightarrow [20,30]
                                                                                lst[-3:-1] \rightarrow [30,40] lst[3:] \rightarrow [40,50]
lst[1:-1] \rightarrow [20,30,40]
                                    lst[::-2] \rightarrow [50,30,10]
lst[::2] \rightarrow [10,30,50]
                                     lst[:] \rightarrow [10,20,30,40,50] płytka kopia sekwencji
Brak wskazania wycinka → od początku / do końca.
W sekwencjach mutowalnych (lista) (list), usuń za pomocą del lst[3:5] i modyfikuj z przydziałem lst[1:4]=[15,25]
```





moduly math, statistics, random.

decimal, fractions, numpy, itp. (por. doc)

 $pow(4,3) \rightarrow 64.0$

zwykła kolejność operacji

if warunek logiczny: → blok instrukcji age<=18: Może iść z kilkoma elif, elif... i tylko jednym state="Dziecko" końcowym else. Wykonywany jest tylko blok elif age>65: pierwszego prawdziwego warunku. state="Emeryt" if bool(x)==True: ⇔ if x: state="Pracownik" if bool(x)==False: \Leftrightarrow if not x: Sygnalizacja błędu: Wyjątki raise ExcClass(...) Przetwarzanie błędów: normalne przetwarzania try:

from monmod import nom1, nom2 as fct

moduły i pakiety przeszukane w ścieżce python path (por sys.path)

import monmod → dostęp przez monmod.nom1 ...

Importy modułów / nazw

Instrukcje warunkowe

tak

→bezpośredni dostęp do nazw, zmiana nazwy za pomocą as

moduł truc⇔ plik truc.py

blok instrukcji wykonywany tylko

jeśli warunek jest prawdziwy



```
Instrukcja pętli iteracyjnej
                                                 Instrukcja pętli warunkowej
   blok instrukcji wykonywany tak długo,
                                                                                    blok instrukcji wykonywany dla każdego
pętl
                                                                                    element kontenera lub iteratora
   dopóki warunek jest prawdziwy
na nieskończone
                                                                                                 for var in sekwencja:
      while warunek logiczny:
                                                                          Kontrola pętli
             blok instrukcji
                                                                                                       sekwencja blok
                                                            break natychmiastowe wyjście
                                                            continue następna iteracja
                                                                                             Przejrzyj wartości sekwencji
   s = 0 } inicjalizacje przed pętlą
                                                                 belse alternatywne
                                                                                                                                                       pętli
                                                                                             s = "tekst" } inicjalizacje przed pętlą
   \mathbf{i} = \mathbf{1} warunęk z co najmniej jedną wartością zmiennej (tutaj \mathbf{i})
                                                                 wyjście z pętli
                                                                  Algo:
                                                                                                                                                        nawyk: nie modyfikuj zmiennej
   while i <= 100:
                                                                                                  zmienna pętli, przypisanie zarządzane przez instrukcję for
        s = s + i**2
i = i + 1
                                                                                             for c in s:
                         🖢 zmień zmienną warunku!
                                                                                                                                 Algo: zlicza liczbę e
                                                                                                  if c == "e":
   print("sum:",s)
                                                                                                                                 w ciągu znaków.
                                                                                                        cnt = cnt + 1
                                                                                             print("znaleziony",cnt,"'e'")
                                                                      Wyświetlanie
 print("v=",3,"cm :",x,"
                                                                                      pętla na dict / set ⇔ pętla na sekwencjach klawiszy
                                                                                       użyj wycięć do zapętlenia podzbioru sekwencji
                                                                                    Przejrzyj indeks sekwencji
 elementy do wyświetlenia: wartości literalne, zmienne, wyrażenia

    modvfikui element w indeksie

 print opcje:

    dostęp do elementów wokół indeksu (przed / po)

 • sep=" "
                            separator elementów, domyślna spacja
                                                                                    lst = [11,18,9,12,23,4,17]
 end="\n"
                            koniec funkcji print, domyślna nowa linia
                                                                                    lost = []
                                                                                                                          Algo: wartości graniczne większe
 • file=sys.stdout drukuj do pliku, domyślne standardowe wyjście
                                                                                    for idx in range(len(lst)): niż 15, zapamiętywanie
                                                                                                                                                        dobry
                                                                         Wejście
                                                                                         val = lst[idx]
                                                                                                                          utraconych wartości.
 s = input("Instrukcje:")
                                                                                         if val > 15:
 input zawsze zwraca string, przekonwertuj go na wymagany typ
                                                                                               lost.append(val)
                                                                                                                                                       Æ
     (por. ramka Konwersje typów po drugiej stronie).
                                                                                               lst[idx] = 15
                                                                                    print("zmieniony:",lst,"-zagubiony:",lost)
len (c) \rightarrow l. elementów
                                        Ogólne operacje na kontenerach
                                                                                    Przejdź jednocześnie przez indeks i wartości sekwencji:
min(c) max(c) sum(c)
                                            Uwaga: w przypadku słowników i zbiorów,
                                                                                    for idx,val in enumerate(lst):
sorted(c) → list posortowana kopia te operacje używają kluczy.
                                                                                                                                 Sekwencje całkowite
val in c → boolean, operator członkostwa in (brak not in)
                                                                                      range ([start,] koniec [,krok])
enumerate (c) → iterator na (index, wartość)
                                                                                     🖢 start domyślnie 0, koniec nie zawarte w sekwencji, krok podpisany, domyślny 1
zip (c1, c2...) → iterator krotek zawierających elementy w tym samym indeksie
                                                                                    range (5) \rightarrow 0 1 2 3 4
                                                                                                                   range (2,12,3) \rightarrow 25811
all (c) → True jeśli wszystkie elementy C mają wartość true, w przeciwnym razie False
                                                                                    range (3,8) \rightarrow 34567
                                                                                                                   range (20,5,-5) \rightarrow 20 15 10
any (c) → True jeśli przynajmniej jedna pozycja C została uznana za prawdziwą, w przeciwnym razie Fa.

Specyficzne dla kontenerów uporządkowanych sekwencji (listy, krotki, ciągi znaków, bajty...)
                                                                                    range (len (seq)) \rightarrow sekwencja indeksów wartości w seq
                                                                                     🖢 range zapewnia niezmienną sekwencję liczb całkowitych konstruowanych w razie potrzeby
reversed (c) \rightarrow odwrócony iterator c*5 \rightarrow duplikat
                                                                                                                                  Definiowanie funkcji
c.index (val) \rightarrow pozycja
                                      c.count(val) \rightarrow liczba wydarzeń
                                                                                    nazwa funkcji (identyfikator)
                                                                                                  nazwane parametry
import copy
copy.copy(c) → płytka kopia kontenera
                                                                                     def fct(x,y,z):
                                                                                                                                                fct
copy.deepcopy(c) → głęboka kopia kontenera
                                                                                            """dokumentacja"""
                                                         Operacje na listach
 h modyfikuj oryginalną listę
                                                                                            # blok instrukcji, obliczanie res itp.
                                                                                            return res - wartość wynikowa połączenia, jeśli nie została obliczona, wynik do zwrócenia: return None
lst.append(val)
                                dodaj element na końcu
lst.extend(seq)
                                dodaj sekwencję elementów na końcu
                                                                                       b parametry i wszystkie zmienne tego bloku istnieją tylko w bloku i podczas
lst.insert(idx, val)
                                wstaw element w indeksie
                                                                                       wywołania funkcji (myśl o "czarnej skrzynce")
                                usuń pierwszą pozycję z wartością val
lst.remove(val)
                                                                                     Zaawansowane: def fct(x,y,z,*args,a=3,b=5,**kwargs):
lst.pop ([idx]) \rightarrow wartość usuń i zwróć element w indeksie idx (domyślnie ostatni)
                                                                                       *args zmienne argumenty pozycyjne (→tuple), wartości domyślne,
                  lst.reverse() sortowanie / odwracanie listy na miejscu
                                                                                       **kwargs zmienna o nazwie argumenty (\rightarrowdict)
                                                        Operacje na zbiorach
           Operacje na słownikach
                                           Operatory:
                                                                                      r = fct(3,i+2,2*i)
                                                                                                                                       Wywołanie funkcji
d[klucz]=wartość
d.clear()
                                             I → suma (znak pionowej kreski)
                                                                                      przechowywanie / użytkowanie jeden argument na
d[klucz] \rightarrow wartość del d[klucz]
                                                                                      zwrócona wartość
                                             & → część wspólna
                                                                                                                parametr
d. update (d2) { zaktualizuj/dodaj pary

    - ^ różnica/różnica symetryczna

                                                                                                                                                   fct
                                                                                                                Zaawansowane:
                                                                                      iest to użycie nazwy
                                                                                                                *sekwencja
**słownik
d.keys()
                                             < <= > >= → relacje zawieranie
                                                                                     funkcji z nawiasami, która
d.values() → iterowalne poglądy na klucze / wartości / skojarzenia
                                           Operatory istnieją również jako metody.
d.items()
d.pop(klucz[,domyślny]) → wartość
                                           s.update(s2) s.copy()
                                                                                    s.startswith (prefiks[,start[,koniec]]) Operacje na łańcuchach
                                           s.add(klucz) s.remove(klucz)
d.popitem() \rightarrow (klucz, wartość)
                                                                                    s.endswith(sufiks[,start[,koniec]]) s.strip([znaki])
d.get(klucz[,domyślny]) \rightarrow wartość
                                           s.discard(klucz) s.clear()
d.setdefault(klucz[,domyślny]) →wartość
                                                                                    s.count(sub[,start[,koniec]]) s.partition(sep) → (przed,sep,po)
                                                                                    s.index(sub[,start[,koniec]]) s.find(sub[,start[,koniec]])
 przechowywanie danych na dysku i ich odczytywanie
                                                                                    s.is...() testy na kategoriach znaków (ex. s.isalpha())
     f = open("plik.txt","w",encoding="utf8")
                                                                                    s.upper()
                                                                                                   s.lower()
                                                                                                                    s.title()
                                                                                                                                    s.swapcase()
                                                                                    s.casefold() s.capitalize() s.center([szerokość,wypelnienie])
                nazwa pliku
                                   tryb otwierania
plik zmiennej
                                                              kodowanie
                                                                                    s.ljust([szerokość,wypełnienie]) s.rjust([szerokość,wypełnienie])
                                                              znaki dla tekstu
                                   "r' czytaj
                na dvsku
do operacji
                                                                                    s.zfill([szerokość]) s.encode(kodowanie) s.split([sep]) s.join(seq)
                                                             pliki:
                                   " 'w' zapisz
                (+ ścieżka ...)
                                                                                                                  wartości do sformatowania Formatowanie
                                                                                       dyrektywy formatujące
                                                              utf8
                                                                      ascii
                                    □ 'a' dodaj
zobacz moduły os, oś. path and pathlib ...'+' 'x' 'b' 't' latin1
                                                                                     "modele{} {} {}".format(x,y,r)-
 zapis
                                   🖢 czytaj pusty ciąg znaków jeśli koniec pliku 🛮 wczytanie
                                                                                      "{wybór: formatowanie! konwersja}"
 f.write("test")
                                  f.read([n])
                                                         → nastepne znaki
                                                                                     Wybór:
                                       jeśli n nie określono, czytaj do końca!
                                                                                                                  "{:+2.3f}".format(45.72793)
 f.writelines(lista linii)
                                                         → lista następnych wierszy
→ następna linia
                                                                                       2
                                  f.readlines([n])
                                                                                                                 \rightarrow '+45.728 '
                                                                                       nom
                                  f.readline()
                                                                                                                 "{1:>10s}".format(8,"toto")
                                                                                       0.nom
   tryb tekstowy t domyślne (odczyt/zapis str), możliwy tryb binarny
                                                                                                                             toto'
                                                                                       4[klucz]
   b (odczyt/zapis bytes). Konwertuj z / na wymagany typ!
                                                                                                                  "{x!r}".format(x="I'm")
                                                                                       0[2]
f.close () h nie zapomnij zamknąć pliku po użyciu!
                                                                                                                <sup>_</sup>→'"I\'m"'
                                                                                     • Formatowanie :
f.flush() zapisz pamięć podręczną f.truncate([rozmiar]) Zmień rozmiar
                                                                                     wypełnij char wyrównanie znak min.szerokość.precyzja~maks.szerokość rodzaj
postęp\ odczytu\ /\ zapisu\ sekwencyjnie\ w\ pliku,\ modyfikowalny\ za\ pomocą:
                                                                                     <>^= + - spacja
                                                                                                             0 na początku do wypełniania 0
f.tell()\rightarrow pozycja
                                     f.seek (pozycja[,źródło])
                                                                                    l. całkowita: b dwójkowy, c znak, d dziesiętny (domyślnie), o ósemkowy, x lub X hex
                                                                                    zmiennoprzecinkowa: e lub E wykładniczy, f lub F punkt stały, g lub G odpowiednie (domyślne),
                                                  with open(...) as f:
  Bardzo częste: otwieranie za pomocą chronionego
 bloku (automatyczne zamykanie) oraz petla
                                                                                    łańcuch znaków: s ...
                                                     for line in f :
  odczytu w wierszach pliku tekstowego:

    Konwersja: s (czytelny tekst) lub r (reprezentacja dosłowna)

                                                         # przetwarzanie line
```