

Wykład 3

- Typy danych w języku Python

3

Typy danych w języku Python

- Typy proste
 - Logiczny (bool)
 - Całkowity (int)
 - Zmiennopozycyjny (float)
 - Zespółony (complex)
 - Napisowy (str)
- Typy strukturalne
 - Lista (list)
 - Krotka (tuple)
 - Zbiór (set)
 - Słownik, tabela (dict)
 - Plik (file)

4

Działania na typach prostych

- typ logiczny (bool)
 - True False
 - not or and
- typ całkowity (int)
 - 12 -21 0b1101 0o77 0xff
 - + - * // % **
 - | & ^ ~ < > <= >=
- typ zmiennopozycyjny (float)
 - 12.3 2e-23
 - + - * / **
- typ zespolony (complex)
 - 3+4j 3.0+4.0j 2j
 - + - * / **

Handwritten notes and examples:

- $13 : 5 = 2 \text{ r } 3$
- | | | | |
|-----|-----|-----|--|
| 200 | div | and | |
| 144 | // | % | |
| 100 | / | % | |
- $81/4/2 = 1$
- $13/5 = 2.6$
- $2 * 3 + 2 = 8$
- | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
- $16 : 14 = 1$
- | | | |
|----|-----|-----|
| 0b | 111 | = 7 |
| 0x | b | = 4 |
| 0x | b | = 3 |
- $x > 2$
- $x < 1$

3

Funkcje wbudowane

- abs(n) - wartość bezwzględna
- chr(n) - znak o kodzie n
- ord(zn) - kod znaku
- min(a,b,...) - najmniejsza z liczb
- max(a,b,...) - największa z liczb
- round(x) - zaokrąglenie wartości
- len(s) - długość napisu, listy, krotki
- type(x) - typ zmiennej
- eval(w) - ewaluacja wyrażenia

*np. input() "2+3*4"*
eval(w)

4

Biblioteki funkcji

from math import sqrt, sin
from math import *
import math
import math as m

niezalecane
math.sqrt(x)
m.sqrt(x)

$\cos \approx 7$

Funkcja	Znaczenie
ceil(<i>value</i>)	rounds up
cos(<i>value</i>)	cosine, in radians
floor(<i>value</i>)	rounds down
log(<i>value</i>)	logarithm, base <i>e</i>
log10(<i>value</i>)	logarithm, base 10
max(<i>value1</i> , <i>value2</i>)	larger of two values
min(<i>value1</i> , <i>value2</i>)	smaller of two values
sin(<i>value</i>)	sine, in radians
sqrt(<i>value</i>)	square root

5

Liczby całkowite

Java	C/C++	zakres	rozmiar
byte	char	-128..127	1B
short	short int	-32768..32767	2B
int	int	-2147483648..214748647	4B
long	long int	-2 ⁶³ ..2 ⁶³ -1	8B

W języku Python 3.x zakres typu int jest ograniczony jedynie dostępną pamięcią.

6

Liczby zmiennopozycyjne

Java, C/C++

typ	zakres	dokładność	rozmiar
float	1.5E-45 .. 3.4E38	7-8	4B
double	5.0E-324 .. 1.7E308	15-16	8B

W języku Python typ float jest równoważny z typem double w języku C/C++.

Kod ASCII

	0	1	2	3	4	5	6	7
0	NUH	DCI	space	@	P		p	
1	STX	DC2	"	1	A	Q	a	q
2	ETX	DC3	#	2	B	R	b	r
3	EOF	XOFF	\$	3	C	S	c	s
4	ETD		%	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(8	H	X	h	x
9	HT	EM)	9	I	Y	i	y
A	LF	SUB	*		J	Z	j	z
B	V	ESC	+		K	[k	{
C	FF	FS	.		L	\	l	
D	CR	GS	=		M]	m	}
E	SO	RS	>		N	^	n	~
F	SI	US	/		O	_	o	del

Rozszerzenie kodu ASCII

128	Ç	144	É	160	á	176	⌘	192	Ł	208	⌚	224	α	240	≡
129	ú	145		161	í	177		193	209	225	ß	241			
130	é	146	Æ	162	ó	178	■	194	┘	210	226	Γ	242	≥	≤
131	â	147	ó	163	ú	179		195	┘	211	227	π	243	≤	≤
132	ä	148	ó	164	ñ	180	┘	196	—	212	228	Σ	244		
133	ä	149	ó	165	Ñ	181	┘	197	┘	213	229	σ	245	┘	┘
134	ä	150	ó	166		182		198	┘	214	230	μ	246		
135	ç	151	ú	167	ó	183	┘	199	┘	215	231	τ	247	≡	≡
136	é	152	ý	168	ó	184	┘	200	┘	216	232	Φ	248	ó	ó
137	é	153	Ö	169	185	┘	201	┘	217	233	Ö	249	ó	ó	ó
138	é	154	Ü	170	┘	186	┘	202	218	234	Ü	250	ó	ó	ó
139	i	155	ó	171	½	187	┘	203	219	235	ó	251	ó	ó	ó
140	i	156	ⓔ	172	¼	188	┘	204	220	236	00	252	ó	ó	ó
141	i	157	✱	173	189	205	221	237	ó	253	ó	ó	ó	ó	ó
142	Ä	158	174	«	190	206	222	238	é	254	ó	ó	ó	ó	ó
143	Ä	159	f	175	191	207	223	239	ó	255	ó	ó	ó	ó	ó

Source: www.LookupTables.com

Unicode

UTF-8

Znaki ASCII kodujemy za pomocą 1 bajta.

Alfabet: łaciński, grecki, armeński, hebrajski, arabski, koptyjski i cyrylica
kodujemy za pomocą 2 bajtów.

Kolejne znaki (m.in. alfabety chiński i japoński) kodowanych jest na 3 i 4 bajtach.

```
00000000 – 0000007F:      0xxxxxxx
00000080 – 000000FF:      110xxxxx 10xxxxxx
00000800 – 0000FFFF:      1110xxxx 10xxxxxx 10xxxxxx
00010000 – 001FFFFF:      11110xxx 10xxxxxx 10xxxxxx 10xxxxxx
```

UCS-2

Wszystkie znaki zapisywane są za pomocą 2 bajtów. Kodowanie to pozwala na zapisanie tylko 65536 początkowych znaków Unikodu.

UCS-4

Wszystkie znaki zapisywane są za pomocą 4 bajtów.

Kodowanie „polskich” znaków

Znak	ISO 8859-2	Unicode	UTF-8
a	161	261	196 133
ć	198	263	196 135
e	202	281	196 153
í	163	322	197 130
ñ	209	324	197 132
ó	211	211	195 179
s	166	347	197 155
z	172	378	197 186
ž	175	380	197 188
Ą	177	260	196 132
Ć	230	262	196 134
Ę	234	280	196 152
Ł	179	321	197 129
Ń	241	323	197 131
Ō	243	243	195 147
Ś	182	346	197 154
Ž	188	377	197 185
Ż	191	379	197 187

Operacje na napisach

- "hello"+"world" "helloworld" # concatenation
- "hello"*3 "hellohellohello" # repetition
- "hello"[0] "h" # indexing
- "hello"[-1] "o" # (from end)
- "hello"[1:4] "ell" # slicing
- len("hello") 5 # size
- "hello" < "jello" 1 # comparison
- "e" in "hello" 1 # search
- escapes: \n, \033, \t
- 'single quotes', """triple quotes"""

'also' "also"
 "also" ... "also" ...
 "also" ... "also" ...

Operacje na napisach

H	e	l	l	o	-	B	o	b
0	1	2	3	4	5	6	7	8

```
>>> s[0:3]
'Hel'
>>> s[5:9]
'-Bob'
>>> s[:5]
'Hello'
>>> s[5:]
'-Bob'
>>> s[:]
'Hello-Bob'
```

s = s[0:3]
p = s
t = s[5:]
s.strip()

13

Funkcje przetwarzające napisy

- `s.lower()` – małe litery
- `s.upper()` – duże litery
- `s.capitalize()` – wielka pierwsza litera
- `s.title()` – wielkie pierwsze litery
- `s.center(szer)` – formatowanie
- `s.ljust(szer)`, `s.rjust(szer)` – j.w.
- `s.count(sub)` – liczba wystąpień sub
- `s.find(sub)` – pierwsza pozycja na której występuje sub
- `s.rfind(sub)` – ostatnia pozycja na której występuje sub
- `s.split()` – lista napisów składowych

t = lower(s)
s = "kole i dla idę do domu"
s.split() → *["kole", "i", "dla", "idę", "do", "domu"]*
s.strip()

14

Typy danych w języku Python

Typy proste

- Logiczny (bool)
- Całkowity (int)
- Zmiennopozycyjny (float)
- Zespólny (complex)
- Napisowy (str)

Typy strukturalne

- Lista (list)
- Krotka (tuple)
- Zbiór (set)
- Słownik, tabela (dict)
- Plik (file)

15

Listy

- Uporządkowane kolekcja elementów
- Elementy listy mogą być różnych typów
- Elementy listy mogą być zmieniane
- Rozmiar listy może się zmieniać
- Elementy dostępne poprzez indeksowanie
- Operacje na listach analogiczne do operacji na napisach $a+b$, $a*3$, $a[0]$, $a[-1]$, $a[1:]$, $len(a)$

Przykład:

```
lista1 = [ ]
lista2 = [2,3,5,7]
lista3 = [23,"ala",[1,2,3]]
```

L = [2,3,5,7]
L[-2] = 5

16

Podstawowe operacje na listach

- `lista.append(element)` – dołącz element do listy
- `lista.pop()` – pobierz element z listy
- `lista.extend(lista)` – dołącz kilka elementów
- `lista.sort()` – posortuj listę w miejscu
- `lista2 = sorted(lista)` – stwórz posortowaną listę
- `lista.reverse()` – odwróć kolejność elementów

pop() *append()*

17

Wyrażenia listowe

```
[ expr for x in collection ]
[ expr for x in collection if warunek ]

kwadraty = [ i**2 for i in range(100) ]
kwadraty_zakonczone_6 = [ i**2 for i in range(100) if i**2%10==6 ]

osoby = [ ('ola',18), ('ula',16), ('ala',19), ... ]
peploletni = [ os for (os,wiek) in osoby if wiek>=18 ]

lx = [ x for x in range(-10,11) ]
ly = [ f(x) for x in lx ]

la = [ 2,3,5 ]
lb = [ 7,11,13 ]

iloczyn_kartezjanski = [ (a,b) for a in la for b in lb ]

LL = [ [3,5,7], [2,4,8] ]
[ x**2 for L in LL for x in L ]
```

18

Tablice w Pythonie

```
t = [ 0 for _ in range(100) ]
```

```
t[23] = 6
```

```
t = {}
for i in range(100): t[i] = 0
t[6] = 23
```

```
from numpy import *
t = zeros(100, float)
```

```
t[6] = 0.1428
```

```
t = [[0 for _ in range(8)] for _ in range(8) ]
```

```
t[6][5] = 23
```

```
t = {}
for i in range(8):
    for j in range(8): t[i,j] = 0
```

```
t[5,6] = 29
```

```
from numpy import *
t = zeros((8,8), float)
```

```
t[6][6] = 0.1428
```

Pytania i zadania

1. Jak wyznaczyć maksimum z 4 liczb nie używając operatorów relacyjnych?
2. Dany jest program:

```
n = int(input())
while n!=rewers(n):
    print(n)
    n = n+rewers(n) # dodawanie arytmetyczne
print("koniec")
```

rewers(n) to liczba n zapisana od końca
Czy powyższy program zakończy się dla każdej liczby naturalnej?
Proszę sprawdzić to dla wszystkich liczb $n < 200$.

< > <= >=
==
!
- 23 + 321
444

20

19

20