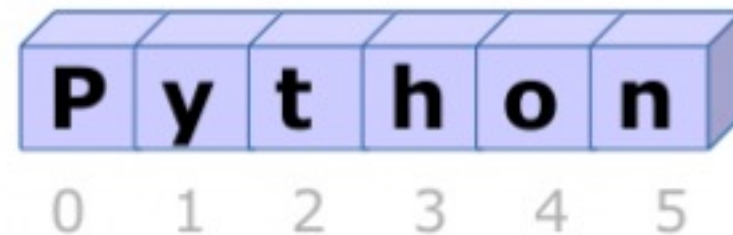


łańcuchy znaków

Podstawy programowania w języku Python



ASCII (American Standard Code for Information Interchange)

Kod dziesiętny	Znak	Kod dziesiętny	Znak	Kod dziesiętny	Znak	Kod dziesiętny	Znak
0	NUL	32	Space	64	@	96	`
1	SOH	33	!	65	A	97	a
2	STX	34	„	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	\$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEI	39	'	71	G	103	g

```
1 print(ord("A")) # 65
2 print(chr(65)) # A
```

Unicode

U+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0100	Ā	ā	Ă	ă	Ą	ą	Ć	ć	Ĉ	ĉ	Č	č	Č	č	Ď	ď
0110	Đ	đ	Ě	ě	Ě	ě	È	è	Ě	ě	Ě	ě	Ĝ	ĝ	Ğ	ğ
0120	Ĝ	ĝ	Ĝ	ĝ	Ĥ	ĥ	Ĥ	ĥ	İ	ı	İ	ı	İ	ı	İ	ı
0130	İ	ı	İ	ı	Ĵ	ĵ	Ķ	ķ	κ	Ł	ł	Ł	ł	Ł	ł	Ł

- jest standardem kodowania znaków, który obejmuje znaki z różnych alfabetów, symboli matematycznych i innych znaków specjalnych
- każdy znak jest reprezentowany przez unikalny numer zwany kodem Unicode

```
1 print(ord("ą")) # 261
2 print(chr(261)) # ą
3 print(hex(261)) # 0x105
4 print("\u0105") # ą
```

UTF-8 (8-bit Unicode Transformation Format)

- system kodowania Unicode, wykorzystujący od 1 do 4 bajtów do zakodowania pojedynczego znaku
- w pełni kompatybilny z ASCII
- jest najczęściej wykorzystywany do przechowywania napisów w plikach i komunikacji sieciowej

Code point ↔ UTF-8 conversion

First code point	Last code point	Byte 1	Byte 2	Byte 3	Byte 4	Code points
U+0000	U+007F	0xxxxxxx				128
U+0080	U+07FF	110xxxxx	10xxxxxx			1920
U+0800	U+FFFF	1110xxxx	10xxxxxx	10xxxxxx		[a] 61440
U+10000	[b] U+10FFFF	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx	1048576

Kodowanie znaków w Pythonie

- znaki są reprezentowane przez typ danych str (string), który reprezentuje ciągi znaków
- każdy znak w ciągu jest kodowany za pomocą jego kodu Unicode
- domyślnym kodowaniem dla ciągów znaków jest UTF-8

Wieloliniowe łańcuchy znaków

- zastosowanie trzech cudzysłówów lub apostrofów z każdej strony ciągu
- pozwala zachowywać wcięcia w tekście

```
1 print("""To jest
2     wielolinijkowy ciąg
3     znaków""")
```

Operacje na łańcuchach znaków

- konkatencja (operator +)
- replikacja (operator *)

```
1 print("Ała " + "ma " + "kota") #Ała ma kota
2 print("kot" * 3) #kotkotkot
```

Punkt kodowy

Punkt kodowy to liczba tworząca znak. Na przykład, 32 to punkt kodowy, który tworzy spację w kodowaniu ASCII.

Funkcje:

- **ord()** - zwraca wartość punktu kodowego ASCII/UNICODE określonego znaku
- **chr()** - przyjmuje punkt kodowy i zwraca jego znak

```
1 print("Ała " + "ma " + "kota") #Ała ma kota
2 print("kot" * 3) #kotkotkot
```


Łańcuchy znaków jako sekwencje

- indeksowanie

```
1 s = "Ała ma kota."  
2 for i in range(len(s)):  
3     print(s[i])
```

- iterowanie

```
1 s = "Ała ma kota."  
2 for c in s:  
3     print(c)
```

Wycinki

Podobnie jak na listach i krotkach, na łańcuchach znaków możemy stosować wycinki.

```
1 string = "Ała ma kota."  
2 print(string[7:11]) #kota  
3 print(string[:3]) #Ała
```

Operatory in, not in




- operator **in** - sprawdza, czy jego lewy argument (łańcuch) można znaleźć gdzieś w obrębie prawego argumentu (inny łańcuch)
- operator **not in** - to po prostu zanegowany wynik operatora in
- w wyniku sprawdzenia otrzymujemy True lub False

```
1 string = "Ała ma kota."  
2 print("Ała" in string) #True  
3 print("k" not in string) #False
```

Niezmiennność łańcuchów znaków

Łańcuchy tekstów są stałe (immutable), dlatego:

- instrukcją **del** nie można usunąć znaku z łańcucha
- instrukcją **del** można usunąć cały łańcuch
- nie posiadają instrukcji **append()** oraz **insert()**

```
1 string = "Ała ma kota."  
2  #del[0] #błąd  
3  #string.append("A kot ma Alę.") #błąd  
4  #string.insert(0, "!") #błąd  
5 del string
```

Pytanie

Jakie są zalety stosowania standardu Unicode?

- a) możliwość kodowania znaków z wielu języków i alfabetów
- b) szybsza praca systemu operacyjnego
- c) ograniczenie liczby kodów znaków do minimum

Odpowiedź: a)

Pytanie

Która z podanych instrukcji na przykładowym fragmencie kodu jest poprawna?

- a) instrukcja w linii 1
- b) instrukcja w linii 2
- c) instrukcja w linii 3
- d) instrukcja w linii 4
- e) wszystkie instrukcje są poprawne
- f) każda z instrukcji wygeneruje błąd

```
1 "Ała".append(" ma kota.")
2 print("Ała"[2] * 99)
3 del "Ała"[1]
4 text = "Ała ma" - " ma"
```

Odpowiedź: b)

Pytanie

Jakie jest domyślne kodowanie używane w Pythonie do przechowywania ciągów znaków typu str?

- a) ASCII
- b) UTF-8
- c) UTF-16

Odpowiedź: b)