

Pole

E 3011

Jan Böhm

RECETOX

April 3, 2024

Co nás dnes čeká

1 Pole

2 Vektory

Pole

Pole (anglicky list) je datová struktura v Pythonu. Je to posloupnost hodnot. Vytváří se pomocí hranatých závorek.

```
1 # list init
2 listNumeric = [3, 0, 1, 9]
3 listString = ["apple", "banana", "cherry"]
4 listBool = [True, True, False, True]
5 emptyList = []
```

Pole - indexování

Python indexuje prvky od 0. Tzn. že první hodnota má index 0, druhá má index 1, atd. Pokud chceme brát hodnoty od konce, poslední je -1, předposlední -2, ...

```
1 listString = ["apple", "banana", "cherry"]
2 print(listString[1])
3 print(listString[0])
4 print(listString[3])
5
6 print(listString[-2])
7
8 # change value at given index
9 listString[1] = "beer"
10 print(listString[1])
```

Pole - přidávání hodnot

Existuje více způsobů, jak přidávat hodnoty do pole. Ukážeme si dva z nich:

- `list.append(value)` – přidá `value` na konec pole
- `list.insert(index, value)` – přidá `value` na pozici před `index`

```
1 listString = ["apple", "banana", "cherry"]
2
3 listString.append("dragonfruit")
4 listString.insert(1, "avocado")
5
6 print(listString)
```

Pole - odstraňování hodnot

Existuje více způsobů, jak odstraňovat hodnoty z pole. Ukážeme si tři z nich:

- `del list[index]` – odstraní hodnotu na pozici `index`
- `list.remove(value)` – odstraní první výskyt `value` z pole
- `list.pop(index = -1)` – odstraní hodnotu na pozici `index` a vrátí ji

```
1 listString = ["apple", "banana", "cherry",  
2             "banana", "dragonfruit"]  
3 del listString[2]  
4 listString.remove("banana")  
5 fruit = listString.pop()  
6  
7 print(listString)  
8 print(fruit)
```

Pole - několik dalších užitečností.

- list1 + list2 – spojení (concat) dvou listů
- len(value) – počet prvků v poli (délka pole)
- list.sort(reverse = False) – seřadí pole (s reverse = True pozpátku)

```
1 fruits = ["apple", "banana", "cherry"]
2 veggies = ["pepper", "potato", "carrot", "onion"]
3
4 food = fruits + veggies
5 print(len(food))
6 food.sort()
7 print(food)
8 food.sort(reverse = True)
9 print(food)
```

Funkce `range(start, stop, step)` vytváří aritmetickou posloupnost od `start` (inclusive) po `stop` (exclusive) z krokem `step`. Výsledkem je objekt typu `range`, který ale můžeme pomocí `list()` převést na pole:

```
1 print(list(range(5)))  
2 print(list(range(1,5)))  
3 print(list(range(10,0,-1)))
```


Aplikování funkce na každý prvek v poli

```
1 L = [1,2,3]
```

```
2 L = L + 2
```

Aplikování funkce na každý prvek v poli

```
1 L = [1,2,3]
2 L = L + 2
```

Tento kus kódu ohlásí chybu, protože dělení je operace která potřebuje dvě čísla, ne list a číslo. Pokud chceme nějakou operaci provést s každým prvkem v poli, existuje na to typicky "pythonovská" konstrukce:

Aplikování funkce na každý prvek v poli

```
1 L = [1,2,3]
2 L = L + 2
```

Tento kus kódu ohlásí chybu, protože dělení je operace která potřebuje dvě čísla, ne list a číslo. Pokud chceme nějakou operaci provést s každým prvkem v poli, existuje na to typicky "pythonovská" konstrukce:

```
1 L = [1,2,3]
2 L = [l + 2 for l in L]
3 print(L)
```

Nebo pokud například si chceme připravit posloupnost od 0 do 1 s krokem 0.1, tak můžeme takto:

```
1 seq = [x/10 for x in list(range(11))]
```

Statistiky

Nad polem čísel L můžeme chtít spočítat statistiky. Např.

- minimum $\min(L)$ a maximum $\max(L)$
- součet $\text{sum}(L)$
- počet prvků $\text{len}(L)$
- ...

Statistiky

Nad polem čísel L můžeme chtít spočítat statistiky. Např.

- minimum $\min(L)$ a maximum $\max(L)$
- součet $\text{sum}(L)$
- počet prvků $\text{len}(L)$
- ...

Průměr a směrodatná odchylka

Napište funkci $\text{mean}(L)$, která spočítá průměr z hodnot v poli L .

Dále napište funkci $\text{sd}(L)$, která spočítá směrodatnou odchylku z hodnot v poli L .

Co nás dnes čeká

1 Pole

2 Vektory

Vektory v Pythonu

Pro práci s vektory existují specializované moduly do Pythonu, např. `numpy`. My si však vystačíme se základním Pythonem a potřebné funkce si naimplementujeme.

Co je vektor?

Vektorem budeme v Pythonu rozumět pole čísel $u = [1, -2, 1/3]$.

Co umíme dělat s vektory I

- Umím vytvořit nulový vektor $\mathbf{0} = (0, 0, \dots, 0)$ libovolné délky n .
- Pro libovolný vektor \mathbf{u} mohu spočítat jeho (euklidovskou) velikost $\|\mathbf{u}\| = \sqrt{u_1^2 + u_2^2 + \dots + u_n^2}$
- Pro libovolný vektor $\mathbf{u} \in \mathbb{R}^n$ a skalár $c \in \mathbb{R}$ mohu spočítat $c \cdot \mathbf{u} = (cu_1, cu_2, \dots, cu_n)$.
- Dva vektory \mathbf{u} a \mathbf{v} stejné délky můžeme sečíst (nebo odečíst): $\mathbf{u} \pm \mathbf{v} = (u_1 \pm v_1, u_2 \pm v_2, \dots, u_n \pm v_n)$.
- Pro libovolnou dvojici stejně dlouhých vektorů umím spočítat jejich euklidovskou vzdálenost $\|\mathbf{u} - \mathbf{v}\|$
- Lineární kombinace $c\mathbf{u} + d\mathbf{v} + \dots$

Co umíme dělat s vektory II: násobení

Je více způsobů, jak násobit vektory stejné délky.

- Skalární součin $\mathbf{u} \cdot \mathbf{v} = u_1 v_1 + u_2 v_2 + \dots + u_n v_n$.
- Odchylka vektorů $\angle \mathbf{u}\mathbf{v} = \theta$, kde $\cos \theta = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$
- Vektorový součin (v \mathbb{R}^3) $\mathbf{u} \times \mathbf{v} = \mathbf{n} \|\mathbf{u}\| \|\mathbf{v}\| \sin \theta$. kde θ je odchylka $\angle \mathbf{u}\mathbf{v}$ a \mathbf{n} je vektor kolmý na \mathbf{u} a \mathbf{v} .
- Smíšený součin $\mathbf{u} \cdot (\mathbf{v} \times \mathbf{w})$.

Skalární součin

Skalární součin $\mathbf{u} \cdot \mathbf{v}$ nám umožňuje poznat, zda jsou 2 vektory na sebe kolmé (když je 0).

Vektorový součin

Velikost vektorového součinu v \mathbb{R}^3 $\mathbf{u} \times \mathbf{v}$ je obsah rovnoběžníku tvořeného vektory \mathbf{u} a \mathbf{v}

Smíšený součin

Smíšený součin v \mathbb{R}^3 $\mathbf{u} \cdot (\mathbf{v} \times \mathbf{w})$ je objem rovnoběžnostěnu tvořeného vektory \mathbf{u} , \mathbf{v} a \mathbf{w} .