

Liste

Kako da pod jednim imenom čuvamo celu paletu, inventar ili stotinu pahuljica

Danas pravimo: paletu sortiranu po svetlini i inventar lika u igri

```
boje = [ ... ]
```

0**crvena****-5****1****žuta****-4****2****zelena****-3****3****plava****-2****4****ljubičasta****-1**

Tri pitanja za zagrevanje

1. prošla tema

```
for i in range(1, 10, 4):  
    print(i, end=" ")
```

Ispis: 1 5 9

2. stringovi

```
s = "PIKSEL"  
print(s[1:4])
```

Ispis: IKS

3. mešano

```
for z in "abc":  
    if z != "b":  
        print(z, end="")
```

Ispis: ac

Pet novih reči za danas

lista	niz vrednosti u uglastim zagradama, pod jednim imenom
element	jedna vrednost u listi
indeks	redni broj mesta u listi; prvi element ima indeks 0
isecanje	uzimanje dela liste: lista[početak:kraj]
metoda	funkcija koja se poziva nad objektom: lista.append(5)

Lista čuva više vrednosti pod jednim imenom

Uglaste zagrade i zarezi

```
prazna_lista = []  
print(prazna_lista)  
  
lista_brojeva = [1, 2, 3]  
print(lista_brojeva)  
  
mesovita_lista = ["Programiranje", 1]  
print(mesovita_lista)
```

ISPIS

```
[]  
[1, 2, 3]  
['Programiranje', 1]
```

list() i množenje liste

```
nova_lista = list("piksel")  
print(nova_lista)  
  
nova_lista = [5] * 10  
print(nova_lista)
```

ISPIS

```
['p', 'i', 'k', 's', 'e', 'l']  
[5, 5, 5, 5, 5, 5, 5, 5, 5, 5]
```

Indeksi i isecanje

Svaki element ima svoje mesto

Indeks je redni broj mesta, a brojanje počinje od 0

```
boje = ["crvena", "žuta", "zelena", "plava", "ljubičasta"]  
print(boje[0])  
print(boje[2])  
print(boje[-1])  
print(boje[-2])
```

ISPIS

```
crvena  
zelena  
ljubičasta  
plava
```

0

"crvena"**-5**

1

"žuta"**-4**

2

"zelena"**-3**

3

"plava"**-2**

4

"ljubičasta"**-1**

negativni indeksi broje od kraja

Šta ispisuje ovaj program?

```
slojevi = ["pozadina", "drveće", "lik", "oblaci"]  
print(slojevi[1], slojevi[-1])
```

Pažljivo: od kog broja počinju indeksi?

A

pozadina oblaci

B

drveće oblaci

tačno

C

drveće lik

D

IndexError

Čudovište: IndexError



indeksi su 0, 1, 2

Kako izgleda

```
IndexError: list index out of range
```

Zašto se pojavljuje

Tražimo indeks koji ne postoji: lista od tri elementa ima indekse 0, 1 i 2 (ili -1, -2 i -3).

Kako se pobeđuje

Poslednji indeks je `len(lista) - 1`, a još jednostavnije `lista[-1]`. Za prolazak koristimo `for element in lista`.

Isecanje: lista[početak:kraj:korak], sa dvotačkom

```
a = ["A", "B", "C", "D", "E", "F"]
```

[početak:kraj]

kraj se ne uključuje

```
print(a[1:4])
```

```
['B', 'C', 'D']
```

[:kraj] i [početak:]

od početka, do kraja

```
print(a[:2], a[4:])
```

```
['A', 'B'] ['E', 'F']
```

:::korak]

svaki drugi element

```
print(a[::2])
```

```
['A', 'C', 'E']
```

:::-1]

ista lista, unazad

```
print(a[::-1])
```

```
['F', 'E', 'D', 'C', 'B', 'A']
```

Šta ispisuje ovaj program?

```
kadrovi = ["k1", "k2", "k3", "k4", "k5", "k6"]  
print(kadrovi[1:5:2])
```

Pomoć: početak 1, kraj 5 (ne uključuje se), korak 2.

A

tačno

```
['k2', 'k4']
```

B

```
['k1', 'k3', 'k5']
```

C

```
['k2', 'k4', 'k6']
```

D

```
['k2', 'k3', 'k4', 'k5']
```

Menjanje liste

Dodajemo, menjamo i uklanjamo elemente

Element menjamo dodelom preko indeksa

```
lista_brojeva = [1, 2, 3, 4, 5]
print(3 in lista_brojeva)
print(lista_brojeva.index(2))
lista_brojeva[0] = "novo"
print(lista_brojeva)
lista_brojeva[2:5] = [4, 6, "kraj"]
print(lista_brojeva)
```

ISPIS

True

1

['novo', 2, 3, 4, 5]

['novo', 2, 4, 6, 'kraj']

in pita da li je vrednost u listi.

index kaže na kom mestu je vrednost. Lista se menja na licu mesta, bez pravljenja nove.

append i insert dodaju jedan element, extend više

append i insert: jedan element

```
lista_brojeva = [1, 2, 3, 4, 5]
lista_brojeva.append(6)
lista_brojeva.insert(0, 0)
print(lista_brojeva)
```

ISPIS

```
[0, 1, 2, 3, 4, 5, 6]
```

+= i extend: više elemenata

```
lista_brojeva = [1, 2, 3]
lista_brojeva += [4, 5]
lista_brojeva.extend([6, 7])
print(lista_brojeva)
```

ISPIS

```
[1, 2, 3, 4, 5, 6, 7]
```

insert(0, x) ubacuje x na mesto sa indeksom 0, a ostali elementi se pomeraju udesno.

Šta ispisuje ovaj program?

```
a = [1, 2]
a.append([3, 4])
print(len(a))
```

Pažljivo: append dodaje tačno jedan element.

A
2

B tačno
3

C
4

D
TypeError

del, pop i remove uklanjaju na tri načina

```
brojevi = [1, 2, 3, 3, 3, 4, 5, 6]
del brojevi[2]
print(brojevi)
del brojevi[5:]
print(brojevi)
print(brojevi.pop())
print(brojevi)
print(brojevi.pop(1))
print(brojevi)
brojevi.remove(3)
print(brojevi)
```

ISPIS

```
[1, 2, 3, 3, 4, 5, 6]
[1, 2, 3, 3, 4]
4
[1, 2, 3, 3]
2
[1, 3, 3]
[1, 3]
```

- del lista[i] briše po indeksu, a del lista[a:b] ceo opseg.
- pop() uklanja i vraća element; bez argumenta poslednji.
- remove(x) briše prvo pojavljivanje vrednosti x.

Šta ispisuje ovaj program?

```
boje = ["crna", "bela", "crna", "siva"]  
boje.remove("crna")  
print(boje)
```

Pažljivo: "crna" se u listi pojavljuje dva puta.

A

```
['bela', 'siva']
```

B

tačno

```
['bela', 'crna', 'siva']
```

C

```
['crna', 'bela', 'siva']
```

D

```
ValueError
```

Prolazak, funkcije i sortiranje

Šta sve možemo da uradimo sa celom listom

Za izmenu elemenata u petlji treba nam indeks

Promenljiva petlje ne menja listu

```
brojevi = [1, 2, 3]
for broj in brojevi:
    broj *= 100
print(brojevi)
```

ISPIS

```
[1, 2, 3]
```

Dodela preko indeksa menja element

```
brojevi = [1, 2, 3]
for i in range(len(brojevi)):
    brojevi[i] *= 100
print(brojevi)
```

ISPIS

```
[100, 200, 300]
```

broj je samo kopija vrednosti iz liste: kad mu dodelimo novu vrednost, element u listi ostaje isti.

len, sum, min i max rade sa celom listom

```
brojevi = [1, 7, 10, 5, 3, 4, 2, 6, 10, 8, 9, 10]
print("Elementa:", len(brojevi))
print("Zbir:", sum(brojevi))
print("Najmanji:", min(brojevi))
print("Najveći:", max(brojevi))
print("Broj desetki:", brojevi.count(10))
```

ISPIS

```
Elementa: 12
Zbir: 75
Najmanji: 1
Najveći: 10
Broj desetki: 3
```

sum radi isto što i ova petlja:

```
brojevi = [1, 2, 3, 4, 5]
zbir = 0
for broj in brojevi:
    zbir += broj
print("Zbir je", zbir)
```

ZBIR PETLJOM

```
Zbir je 15
```

sorted pravi novu listu, a sort menja postojeću

```
brojevi = [3, 1, 4, 1, 5, 9, 2]
nova = sorted(brojevi)
print(nova)
print(brojevi)
brojevi.sort(reverse=True)
print(brojevi)
```

ISPIS

```
[1, 1, 2, 3, 4, 5, 9]
[3, 1, 4, 1, 5, 9, 2]
[9, 5, 4, 3, 2, 1, 1]
```

- `sorted(lista)` vraća novu sortiranu listu.
- `lista.sort()` menja samu listu.
- `reverse=True` daje opadajući poredak.

Šta ispisuje ovaj program?

```
brojevi = [3, 1, 2]  
x = brojevi.sort()  
print(x)
```

Pažljivo: šta vraća metoda sort?

A

[1, 2, 3]

B

tačno

None

C

[3, 1, 2]

D

TypeError

Argument `key` kaže po čemu se elementi porede

```
oblici = ["kvadrat", "trougao", "Pravougaonik"]  
print(sorted(oblici))  
print(sorted(oblici, key=str.lower))
```

```
reci = ["četskica", "boja", "platno"]  
print(sorted(reci, key=len))  
print(max(reci, key=len))
```

ISPIS

```
['Pravougaonik', 'kvadrat', 'trougao']  
['kvadrat', 'Pravougaonik', 'trougao']  
['boja', 'platno', 'četskica']  
četskica
```

`key=str.lower`

poredi bez obzira na velika i mala slova.

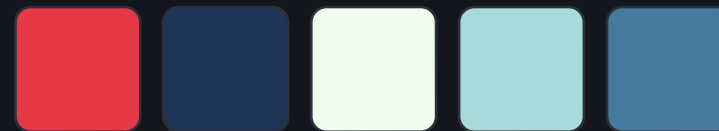
`key=len`

poredi po dužini.

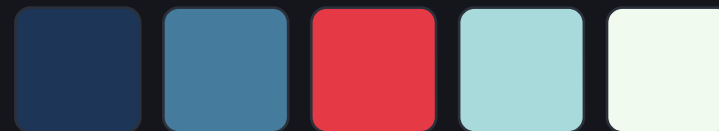
Paleta sortirana po svetlini

```
paleta = [  
    (230, 57, 70),  
    (29, 53, 87),  
    (241, 250, 238),  
    (168, 218, 220),  
    (69, 123, 157),  
]  
po_svetlini = sorted(paleta, key=sum)  
for boja in po_svetlini:  
    print(boja, sum(boja))
```

pre sortiranja



po svetlini (zbir r + g + b)



169 349 357 606 729

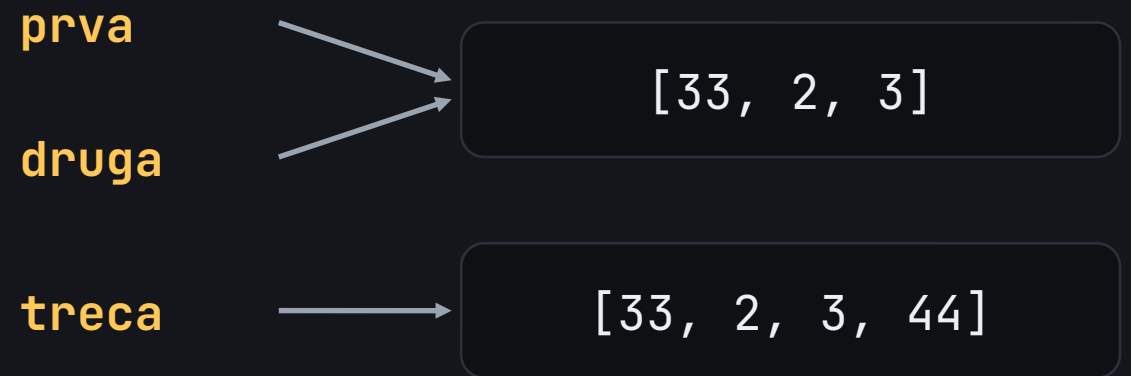
(230, 57, 70) je torka: kao lista, ali se ne menja. Torke radimo u sledećoj temi.

key=sum: za svaku boju Python izračuna sum(boja) i poredi te zbirove.

b = a ne pravi kopiju, nego drugo ime za istu listu

```
prva = [1, 2, 3]
druga = prva
druga[0] = 33
print(prva)

treca = prva.copy()
treca.append(44)
print(prva)
print(treca)
```



ISPIS

```
[33, 2, 3]
[33, 2, 3]
[33, 2, 3, 44]
```

Šta ispisuje ovaj program?

```
a = [1, 2, 3]
b = a
c = a[:]
a.append(4)
print(len(b), len(c))
```

Pažljivo: `a[:]` je isecanje cele liste.

A

3 3

B

tačno

4 3

C

4 4

D

3 4

List comprehension pravi novu listu u jednom redu

Petljom

```
kvadrati = []  
for x in range(5):  
    kvadrati.append(x * x)  
print(kvadrati)
```

U jednom redu

```
kvadrati = [x * x for x in range(5)]  
print(kvadrati)
```

ISTI ISPIS U OBA SLUČAJA

```
[0, 1, 4, 9, 16]
```

Sa uslovom: samo parni brojevi

```
parni = [x for x in range(10) if x % 2 == 0]  
print(parni)
```

ISPIS

```
[0, 2, 4, 6, 8]
```

Pahuljice padaju: nova lista sa novim pozicijama

```
pahuljice = [(40, 10), (120, 60), (200, 30)]  
print(pahuljice)  
pahuljice = [(x, y + 30) for (x, y) in pahuljice]  
print(pahuljice)
```

ISPIS

```
[(40, 10), (120, 60), (200, 30)]  
[(40, 40), (120, 90), (200, 60)]
```

(x, y) je jedna pahuljica; for (x, y) in pahuljice u svakom prolazu raspakuje oba broja.

y raste naniže: y + 30 znači 30 piksela niže



Ne brišemo iz liste kroz koju petlja prolazi

Ne ovako: petlja preskače elemente

```
a = [1, 1, 2, 1, 1]
for element in a:
    if element == 1:
        a.remove(element)
print(a)
```

ISPIS

[2, 1, 1]

Ovako: prolazimo kroz kopiju a[:]

```
a = [1, 1, 2, 1, 1]
for element in a[:]:
    if element == 1:
        a.remove(element)
print(a)
```

ISPIS

[2]

Još kraće: `a = [e for e in a if e != 1]`

Pravimo inventar lika u igri

Cilj: za 15 minuta lik ima torbu u koju uzima i iz koje baca predmete. Torba prima najviše 4 predmeta.

- 1 Lista inventar sa jednim predmetom
- 2 while True: pitamo za akciju; kraj izlazi
- 3 uzmi: append, ali samo dok je len < 4
- 4 baci: remove, samo ako je predmet u listi (in)
- 5 Posle svake akcije ispisujemo torbu

KAKO IZGLEDA IGRA

```
uzmi, baci ili kraj? uzmi  
Predmet: mač  
Torba: ['baklja', 'mač']  
uzmi, baci ili kraj? uzmi  
Predmet: štit  
Torba: ['baklja', 'mač', 'štit']  
uzmi, baci ili kraj? baci  
Predmet: baklja  
Torba: ['mač', 'štit']
```

Ceo program „Inventar“

```
inventar = ["baki"]  
while True:  
    akcija = input("uzmi, baci ili kraj? ")  
    if akcija == "kraj":  
        break  
    predmet = input("Predmet: ")  
    if akcija == "uzmi" and len(inventar) < 4:  
        inventar.append(predmet)  
    elif akcija == "baci" and predmet in inventar:  
        inventar.remove(predmet)  
    else:  
        print("To ne može!")  
print("Torba:", inventar)
```

append

uzmi: predmet ide na kraj torbe

in i remove

baci: briše samo predmet koji postoji

len(inventar) < 4

torba prima najviše 4 predmeta

Pronađi grešku

```
boje = ["crvena", "zelena", "plava"]  
for i in range(1, len(boje) + 1):  
    print(boje[i])
```

ISPIS I PORUKA O GREŠCI

```
zelena  
plava  
Traceback (most recent call last):  
  File "boje.py", line 3, in <module>  
    print(boje[i])  
          ~~~^^^  
IndexError: list index out of range
```

Program ispiše dve boje, a zatim pukne.
Kako glasi ispravan drugi red?

zelena, plava

ispis pre greške: indeksi 1 i 2 postoje

line 3

red u kom je greška: boje[i]

IndexError

u trećem prolazu i je 3, a takav indeks ne postoji

Izaberite nivo i počnite

LEVEL 1

Moja paleta

Napravite listu od pet boja (stringova). Ispišite prvu boju, poslednju boju i srednje tri.

Pomoć: indeksi 0 i -1 i isecanje [1:4].

LEVEL 2

Najsvetlija boja

Za paletu torki sa slajda ispišite najsvetliju i najtamniju boju.

Pomoć: max i min primaju isti argument key kao sorted.

LEVEL 3

Sneg pada

Tri pahuljice (x, y). U četiri kadra pomerite svaku 30 piksela naniže; pahuljica ispod $y = 100$ vraća se na vrh.

Pomoć: petlja za kadrove, a u njoj nova lista pozicija.

Ista lista kasnije pravi sneg na ekranu

```
for (x, y) in pahuljice:  
    pg.draw.circle(ekran, "white", (x, y), 3)  
pahuljice = [(x, y + 1) for (x, y) in pahuljice]
```

Sledeća tema su kolekcije:

torke, skupovi i rečnici. Pozicija (x, y) je već torka, a lik u igri postaće rečnik.



U temi o PyGame-u petlja crta svaku pahuljicu iz liste, a zatim se cela lista pomeri za jedan piksel naniže.

Rezime

Indeksi i isecanje

```
b[0]    b[-1]  
b[1:3]  b[::-1]
```

Prvi indeks je 0; kraj isecanja se ne uključuje.

Menjanje liste

```
b.append(x)    b.insert(0, x)  
b.pop()        b.remove(x)
```

Metode menjaju samu listu.

Sortiranje

```
sorted(b, key=len)  
b.sort(reverse=True)
```

sorted pravi novu listu; sort menja postojeću.

Kopija i comprehension

```
c = b.copy()  
[x * 2 for x in b]
```

b = a ne kopira listu; comprehension pravi novu.

Pitanje pre odlaska

```
a = [1, 2, 3]  
a.append([4, 5])  
a.pop(0)
```

Koliko elemenata ima lista a posle ova tri reda?

Odgovor pošaljite pre nego što izađete. Jedan broj je dovoljan.



Literatura

- Python dokumentacija, Tutorial, poglavlje „Data Structures“, delovi „More on Lists“ i „List Comprehensions“: docs.python.org/3/tutorial
- Python dokumentacija, „Sorting Techniques“ (sorted, sort i argument key): docs.python.org/3/howto/sorting.html
- Wentworth, P., Elkner, J., Downey, A. B., Meyers, C. How to Think Like a Computer Scientist: Learning with Python 3 (besplatna knjiga)
- Sweigart, A. Invent Your Own Computer Games with Python, 4. izdanje, No Starch Press (besplatno na inventwithpython.com)

Dodatna literatura i zanimljivi sajtovi

Petlja: Priručnik za Python

petlja.org · Priručnik za Python

Besplatan priručnik na srpskom, sa zadacima po nivoima težine.

Python Tutor

pythontutor.com

Crta korak po korak šta program radi i šta je u kojoj promenljivoj.

Exercism: Python

exercism.org/tracks/python

Besplatni zadaci za vežbu, poređani po temama.

Lospec: palete boja

lospec.com/palette-list

Biblioteka gotovih paleta boja za piksel-art.

Hvala!

Pitanja su uvek dobrodošla: na času, u četvrtu ili međuvremeno.

Slajdovi, primeri i kartica za podsećanje su na sajtu predmeta.

PREDAVANJA

dr Staša Vujičić Stanković

stasa.vujicic.stankovic@math.rs

www.matf.bg.ac.rs/~stasa.vujicic.stankovic

VEŽBE

Nevena Ćirić

poincare.matf.bg.ac.rs/~nevena.ciric/p1uu.html