

## ПРИПРЕМА ЧАСА

### ИНФОРМАТИКА И РАЧУНАРСТВО ЗА ШЕСТИ РАЗРЕД

Наставна тема:	Рачунарство
Редни број часа:	13.
Наставна јединица:	Торке и листе
Тип часа:	Обрада
Циљ часа:	Упознавање ученика са структурама података као што су: <ul style="list-style-type: none"><li>• Торке</li><li>• Листе</li></ul>
Исход часа:	Ученик ће након часа бити у могућности да: <ul style="list-style-type: none"><li>• Самостално препозна део програма који се може реализовати уз помоћ поменутих структура података</li><li>• Увиди разлику између сличних типова структура података</li><li>• Претражује, сортира, надовезује</li><li>• Кобинује више типова података и користи сложене структуре података</li></ul>
Облици рада:	Фронтални, индивидуални, рад у пару
Наставне методе:	Метода усменог излагања, илустровано – демонстративна метода, метода вођеног разговора
Место реализације часа:	Рачунарски кабинет

## **ТОК ЧАСА**

### **УВОДНИ ДЕО ЧАСА (5 МИНУТА)**

У уводном делу часа потребно је са ученицима обновити рађене типове података (цели и реални бројеви, логичне вредности и ниске) и закључити да су се до ове наставне јединице сви подаци чували у појединачним променљивама. У сложенијим програмима и реалним ситуацијама потребно је обрадити веће количине података, па коришћење појединачних променљивих не би било практично. У тим случајевима се користе структуре података које омогућавају да се уз помоћ једне променљиве представљају читаве колекције података које се састоје од више вредности различитих типова. Предочити ученицима да постоје различите врсте структура података и да бирамо коју ћемо користити у односу на то које податке желимо сачувати и које операције са њима обављати.

### **ГЛАВНИ ДЕО ЧАСА (35 МИНУТА)**

#### **ТОРКЕ И ЛИСТЕ**

У главном делу часа, упознати ученике са појмовима торка и листа. Дати пример једног одељења и дискутовати са њима како би се списак ученика могао представити. Када би сваки ученик представљао засебну променљиву та репрезентација података не би била практична за обраду нити визуелно прихватљива. Објаснити им да ће уз помоћ торке и листе моћи једноставно да одговарају на питања као што су - колико укупно има ученика, колики просек има који ученик, ко је број који се уноси са улаза у дневнику итд. Списак ученика можемо представити на следећи начин:

```

1 ucenici = ("Aleksić Jovan", "Bojanić Igor", "Vasiljević Aleksa",
2           "Davidović Milica", "Đorđević Jovana", "Ivanović Vladimir",
3           "Jovanović Ana", "Kojić Petra", "Lukić Tijana",
4           "Maksimović Magdalena", "Miletić Petar", "Novaković Ivan",
5           "Obradović Jasna", "Petrović Jovan", "Petrović Aleksandra",
6           "Radović Selena", "Rakočević Zorica", "Stanić Predrag",
7           "Tanasković Uroš", "Čirković Aleksandar", "Urošević Srđan",
8           "Cvetković Janko", "Čukić Nikola", "Štulović Marko")
9

```

Слика 1

Синтакса за дефинисање торке није компликована. Подаци се наводе између две мале заграде ( и ) , а елементи су одвојени зарезима. Допуштено је делити наредбе у више линија, јер је садржај окружен заградама. Дефинисање листе се разликује од дефинисања торке по томе што се уместо малих заграда употребљавају угласте заграде- [ и ] . Списак ученика одељења из нашег примера би као листа било дефинисано на следећи начин:

```

1 ucenici = ["Aleksić Jovan", "Bojanić Igor", "Vasiljević Aleksa",
2           "Davidović Milica", "Đorđević Jovana", "Ivanović Vladimir",
3           "Jovanović Ana", "Kojić Petra", "Lukić Tijana",
4           "Maksimović Magdalena", "Miletić Petar", "Novaković Ivan",
5           "Obradović Jasna", "Petrović Jovan", "Petrović Aleksandra",
6           "Radović Selena", "Rakočević Zorica", "Stanić Predrag",
7           "Tanasković Uroš", "Čirković Aleksandar", "Urošević Srđan",
8           "Cvetković Janko", "Čukić Nikola", "Štulović Marko"]
9

```

Слика 2

Појединачни чланови торке и листе се називају елементи и они могу бити било ког типа података. Могуће је да иста листа или торка садржи елементе различитог типа. Разлика између торке и листе је чињеница да код торке не можемо мењати вредности када их једном дефинишемо. Листе омогућавају мењање, брисање и додавање нових елемената па се оне и користе када се очекују ове операције.

Елементима се приступа на основу индекса, односно њихове позиције. Бројање креће од нуле. Први ученик у нашем примеру се налази на позицији нуле иако је први у дневнику, други у дневнику се налази на позицији 1 и тако даље. Чак и да су у питању бројеви, а не имена и презимена, то нас не треба збунити, јер се њима приступа на основу позиције, а не вредности, па се тако напр. на 3. месту може наћи број 5.

Првом елементу је могуће приступити са **ucenici[0]**, а вредност овог израза је “Aleksić Jovan”. Негативни индекси представљају бројање од краја, са десна на лево. Последњем у нашем дневнику можемо приступити са **ucenici[-1]**, претпоследњем са **ucenici[-2]** итд.

Издавање дела тројке или листе можемо извршити помоћу **ucenici[a:b]**, где улазе елементи укључујући позицију а, али искључујући позицију b. Ако изоставимо број b, излиставају се елементи од позиције а до краја. Ако желимо да прикажемо ученике од прве до пете позиције, распон [1:5] је полуотворен, па ће ученик који има позицију 1 бити приказан, док ученик са позије 5 не. Након наредбе **ucenici[1:5]**, излистани ће бити Bojanić Igor, Vasiljević Aleksa, Davidović Milica и Đorđević Jovana.

*Пример:*

Написати програм којим за дати број приказује име и презиме ученика који је тај број у дневнику, имајући у виду да је индекс за 1 мањи од броја у дневнику, јер позиције крећу од 0.

```
1  ucenici = ("Aleksić Jovan", "Bojanić Igor", "Vasiljević Aleksa",  
2           "Davidović Milica", "Đorđević Jovana", "Ivanović Vladimir",  
3           "Jovanović Ana", "Kojić Petra", "Lukić Tijana",  
4           "Maksimović Magdalena", "Miletić Petar", "Novaković Ivan",  
5           "Obradović Jasna", "Petrović Jovan", "Petrović Aleksandra",  
6           "Radović Selena", "Rakočević Zorica", "Stanić Predrag",  
7           "Tanasković Uroš", "Čirković Aleksandar", "Urošević Srđan",  
8           "Cvetković Janko", "Čukić Nikola", "Štulović Marko")  
9  
10 broj = int(input("Broj u dnevniku: "))  
11 print(ucenici[broj - 1])  
12
```

Слика 3

Претпоставимо да су се ученици изјаснили шта желе бити кад порасту. Ако знамо дужину торке или листе може се сваки елемент сместити у посебну променљиву. Направимо посебне променљиве које ће носити имена неких занимања и које ће садржати пуна имена ученика који баш то желе постати. Направимо листу изјашњених ученика која ће се називати *zanimanja*.

```

1 zanimanja = ("Aleksić Jovan", "Bojanić Igor", "Vasiljević Aleksa",
2             "Davidović Milica", "Đorđević Jovana")
3 lekar = zanimanja[0]
4 pilot = zanimanja[1]
5 voditelj = zanimanja[2]
6 pevacica = zanimanja[3]
7 arhitekta = zanimanja[4]

```

Слика 4

Постоји једноставнији начин да се ово постигне уз помоћ поступка који се назива распакивање:

```

1 zanimanja = ("Aleksić Jovan", "Bojanić Igor", "Vasiljević Aleksa",
2             "Davidović Milica", "Đorđević Jovana")
3 lekar, pilot, voditelj, pevacica, arhitekta = zanimanja
4 print(pilot)
5

```

Слика 5

Видимо да је решење Бојанић Игор, јер се променљивој са леве стране додељује први елемент листе са десне стране, другој други и тако даље. Променљиве са леве стране – лекар, пилот, водитељ, певачица, архитекта су могле бити наведене у склопу торке или листе, резултат би био исти.

## ФУНКЦИЈЕ ЗА РАД СА ТОРКАМА И ЛИСТАМА

- `len` - дужина листе
- `sum` - збир елемената листе
- `min` - најмањи елемент у листи
- `max` - највећи елемент у листи

*Пример:*

Дате су оцене из неколико предмета једног ученика. Израчунај просечну оцелу и прикажи највишу и најнижу оцелу коју је тај ученик добио.

```

1 ocene = [3, 2, 5, 5, 5, 4]
2 prosek = sum(ocene) / len(ocene)
3 najniza = min(ocene)
4 najvisa = max(ocene)
5 print(prosek)
6 print(najniza)
7 print(najvisa)

```

Слика 6

## ПРЕТРАГА У ТОРКАМА И ЛИСТАМА

Уз помоћ оператора **in** проверавамо да ли елемент припада одређеној торки или листи. Напр. наредбом **“Aleksić Jovan” in ucenici** проверавамо да ли је Алексић Јован на списку ученика неког одељења.

*Пример:*

Написати програм којим се за унето име и презиме ученика проверава да ли је ученик одељења чији списак поседујемо.

```

1 ucenici = ("Aleksić Jovan", "Bojanić Igor", "Vasiljević Aleksa",
2           "Davidović Milica", "Đorđević Jovana", "Ivanović Vladimir",
3           "Jovanović Ana", "Kojić Petra", "Lukić Tijana",
4           "Maksimović Magdalena", "Miletić Petar", "Novaković Ivan",
5           "Obradović Jasna", "Petrović Jovan", "Petrović Aleksandra",
6           "Radović Selena", "Rakočević Zorica", "Stanić Predrag",
7           "Tanasković Uroš", "Čirković Aleksandar", "Urošević Srđan",
8           "Cvetković Janko", "Čukić Nikola", "Štulović Marko")
9 ucenik = input("Uneti ime i prezime učenika:")
10 if ucenik in ucenici:
11     print(ucenik, "je u ovom odeljenju.")
12 else:
13     print(ucenik, "nije u ovom odeljenju.")

```

Слика 7

## НАДОВЕЗИВАЊЕ

Више торки можемо надовезати коришћењем операције **+** уз помоћ које се креира нова торка тј. листа. Није могуће надовезати листу и торку.

Пример:

Написати програм којим се за дате висине девојчица и дечака у одељењу рачуна просечна висина свих ученика.

```
1 visine_devojčica = [165, 153, 155, 155, 157]
2 visine_decaka = [170, 168, 173, 156, 172]
3 visine = visine_devojčica + visine_decaka
4 print(sum(visine) / len(visine))
```

Слика 8

## СОРТИРАЊЕ

Сортирање елемената торке или листе се врши помоћу функције **sorted**. Ова функција уређује елементе од најмањег до највећег или од највећег до најмањег при чему гради нову торку или листу. Метода **sort** применљива је само на листе и она мења листу тако што јој елементе гради у сортираном редоследу. Ово је ефикасније, али се оригинална листа, тј. њен редослед елемената губи.

Пример:

За дату листу ученика која није сортирана, сортирај их по абecedном реду.

```
1 učenici = ["Obradović Jasna", "Čukić Nikola", "Petrović Jovan",
2 | | | | | "Davidović Milica", "Lukić Tijana", "Aleksić Jovan"]
3 print(sorted(učenici))
```

Слика 9

## **ЗАКЉУЧНИ ДЕО ЧАСА (5 МИНУТА)**

У закључном делу часа поновити најважније појмове из данашње лекције и укратко препричати њихову улогу. Разјаснити евентуалне нејасноће и тиме осигурати разумљивост наставне јединице. Нагласити важност торки и листи у програмирању, не само у већ и у Python-у осталим програмским језицима.

*Домаћи задатак:*

1. За дати списак ученика и њихових оцена одредити који ученик има најмањи просек и приказати тај просек.
2. Уз помоћ распакивања написати програм којим се за унету улогу на школској представи добија име и презиме ученика који је игра.
3. За дате листе поена који су резултати теста групе А и групе Б одредити највећи и најмањи број поена за обе групе, као и просечан број поена на овом тесту.