



## Пресликавања

Врло често се дешава да је у решавању различитих проблема потребно од једне колекције података добијеш другу. На пример, на основу листе површина учионица твоје школе добијеш процену капацитета сваке од учионица на основу правила које каже да је потребно оставити један квадратни метар за сваког ученика и 4 квадратна метра за таблу и катедру. Очекује се да за сваки елемент листе површина учионица одредиш процену и сместиш је у другу листу. Дакле, за сваки елемент прве листе се према задатом правилу одређује одговарајућа вредност, такозвана **слика**. Читав процес одређивања слике, тј. придруживања вредности слике полазној вредности, такозваном **оригиналу**, се у математици назива **пресликавањем**. На овом часу ћеш научити неколико начина на које у Пајтону можеш извести пресликавање једне листе у другу. За почетак одгледај следећи видео

### [Python – Пресликавања](#)

Прочитај сада следећи задатак и покушај да изведеш задато пресликавање.

#### Задатак 1.

Дата је листа дужина страница. Формирај и испиши листу њихових површина.

За почетак покушај да испишеш површине без формирања нове листе. Нека дужине страница буду 1, 2, ..., 9. То би могао да изведеш на следећи начин:

```
1 for a in range (1,10):  
2     print(a*a)
```

Поступак исписа се не би пуно разликовао ни када би листа величина страница била другачије задата, на пример навођењем вредности елемената.

```
1 stranice = [1,2,4,6]  
2 for a in stranice:  
3     print(a*a)
```



Покушај сада да измениш свој код тако што ћеш посао који он треба да обави поделити на два мања и сваки појединачно описати сваки. Како поделити? У суштини проблем који се твојим кодом решава се може поделити на два потпроблема:

- један јесте обрада колекције листе страница и испис добијених вредности, тј. површина за сваку страницу појединачно,
- а други дефинисање начина на који ће се добити вредност у коју треба да буде пресликана задата вредност странице, тј. одређивање површине.

Што се првог потпроблема тиче његово решење ће бити готово идентично претходном коду, са том разликом што ћеш на месту на којем је потребно израчунавање површине за одређену страницу предвидети позив функције која израчунава површину квадрата на основу величине странице. Ту функцију би користио на начин на који си користио и уграђене функције, на пример апсолутну вредност. Нажалост, функција која одређује површину квадрата на основу дужине његове странице не спада у уграђене Пајтон функције, па ћеш морати сам да је напишеш.

Део кода којим се решава први потпроблем би могао да изгледа овако:

```
1 stranice = [1,2,4,6]
2 for a in stranice:
3     print(povrsina(a))
```

Сада недостаје функција `povrsina()` којом се решава други потпроблем. Дефинисањем функције ћеш рећи Пајтон окружењу шта тачно треба да уради када наиђе на позив `povrsina(a)`. Дефиниција функције обухвата:

- њено име (у твојем случају `povrsina`),
- попис података које функција добија при позиву (у нашем случају један број који представља величину станице),
- поступак којим се одређује вредност коју функција треба да израчуна (у нашем случају је то одређивање вредности израза  $a*a$ ) и
- позив специјалне команде `return` облика `return vrednost` којом се наглашава коју ће израчунату вредност (`vrednost`) функција да врати.



Дефиниција функције површина би могла да изгледа овако:

```
1 def povrsina(a):  
2     return a*a  
3
```

Обједињено у једном коду дефиниција саме функције и петља којом се исписују површине могу изгледати овако:

```
1 def povrsina(a):  
2     return a*a  
3     stranice = [1,2,4,6]  
4     for a in stranice:  
5         print(povrsina(a))
```

Обрати пажњу да је важно да прво дефинишеш функцију `povrsina`, па тек онда напишеш петљу у којој се функција користи. Уколико би редослед био обрнут Пајтон окружење у тренутку када буде извршавало петљу не би имало информацију о томе шта тачно треба да ради када добије позив функције `povrsina`. На погрешан редослед ћете упозорити само окружење обавештењем о грешци.

Покушај сада да искористиш ово што си до сада урадио и да решиш задатак онако како је и тражено. Покушај да измениш код тако што ћеш направити нову листу од израчунатих површина, уместо да само да их испишеш.

Један од начина за решавање овог задатка био би могао да буде овај:

```
1     stranice = [1,2,4,6]  
2     povrsine = [a*a for a in stranice]  
3     print(povrsine)  
4
```

Дакле, листа `povrsine` је дефинисана командом

```
povrsine = [a*a for a in stranice]
```



којом се каже да ће за свако  $a$  из листе `stranice` бити израчуната вредност  $a*a$  која ће постати члан нове листе.

Уколико напишеш и одговарајућу функцију твој код могао да изгледа и овако.

```
1 def povrsina(a):
2     return a*a
3 stranice = [1,2,4,6]
4 povrsine = [povrsina(a) for a in stranice]
5 print(povrsine)
6
```

Погледај сада један сличан пример.

**Нека је дата функција**

```
def obim(a, b, c):
    return a + b + c
```

**и нека је дата листа торки које престављају величине страница троуглова, на пример:**

```
troglovi = [(3, 4, 5), (5, 12, 13), (7, 24, 25)]
```

**и нека је потребно исписати обим сваког од троугла из листе.**

Ако желиш да употребиш дату функцију за рачунање обима да би израчунао обиме троуглова описаних у листи потребно је да имаш у виду да није могуће да напишеш овако нешто

```
for t in trouglovi:
    print(obim(t))
```

јер је `t` торка, док функција не очекује торку већ три броја, на пример исправан позив би био `obim(5,6,7)`. Да би торку функција могла да прихвати, потребно је нагласити Пајтон окружењу да је мора да дати функцији у облику који функција може да прихвати, тј. у такозваном је развијеном облику, што се изводи следећим позивом

```
print(obim(*t))
```

Покушај сада да дефинишеш листу која ће садржати обиме задатих троуглова.

### Задатак 2.

Напиши програм који би задати текст исписао тако што би сваку појаву самогласника заменио једним истим самогласником, на пример *и*.

### Упутство за решавање

Овај задатак се решава на начин на који је решаван и претхони задатак. Наиме, текст се посматра као иста карактера. Сваки њен елемент треба пресликати посебном функцијом у одговарајућу слику. Функција ће одређивати слику тако што ће сваки сугласник сликати у исту вредност, а



---

самогласник у слово 'i'. Дакле, твој задатак је да напише функцију која добија једно слово и одређује његову слику, и да је затим искористиш у пресликавању листе слова. Уколико се не сналазиш, помоћ потражи у [интерактивном уџбенику](#).