



Ниске

Поред бројева, рачунари су веома добри и у раду са текстом. Текст се састоји од слова (малих и великих), цифара, размака, интерпункцијских знакова (на пример тачака, зареза, упитника, узвичника) и слично. Све те знакове једним именом називамо **карактери**.

Неки програмски језици подржавају само веома узак скуп карактера (од слова је могуће користити само слова енглеске абецедe), међутим, програмски језик Python3 користи широк скуп карактера који обухвата и све карактере потребне за писање на већини језика света, укључујући и слова ћириличног и латиничког писма која се користе у српском језику.

Поменути основни скуп карактера је довољан само за запис текста на енглеском језику назива се **ASCII**, док се овај шири скуп карактера назива **Unicode**.

Низ карактера чини ниску или стринг (од енглеске речи string која значи ниска). Ниске се у програму записују између наводника. На пример, ниске су "Zdravo" или "Programski jezik Python.". Уместо двоструких равноправно се могу користити и једноструки наводници (на пример, 'Zdravo'), међутим, да се не би збуњивао користећемо двоструке наводнике.

Тема овог часа су управо ниске и рад са нискама у програмском језику Python. За почетак одгледај следећу видео-лекцију

 [Python – ниске](#)

У програмском окружењу Python испиши и покрени команду којом се исписује поздравна реченица Здраво, свете!. Команда би могла да изгледа овако:

```
print("Zdravo, svete!")
```



Операције и уграђене функције за рад са нискама

Слично као и над бројевима и над нискама се могу вршити одређене операције. Једна од основних операција је **спајање две ниске**. Ова операција донекле подсећа на сабирање и обележава се знаком +. На пример,

вредност израза "abraka" + "dabra" је "abrakadabra".

Задатак 1.

Напиши програм у којем се од ниске која представља име и ниске која представља презиме једне особе формира и испише ниска која садржи име и презиме те особе.

Предлог решења

```
1 ime = "Petar"
2 prezime = "Kralj"
3 ime_i_prezime = ime + prezime
4 print(ime_i_prezime)
```

Ако желиш да име и презиме имају размак између додај размак испред презимена.

```
2 prezime = " kralj"
```

Размак у тексту се, такође, сматра карактером, па се ниска која садржи један размак записује овако " ". Тако је претходно решење кориговати тако да се не прави промена у ниски која бележи презиме већ да се размак уметне као додатне ниска између имена и презимена.

```
1 ime = "Petar"
2 prezime = "Kralj"
3 ime_i_prezime = ime + " " + prezime
4 print(ime_i_prezime)
```

Још једна интересантна операција је да се ниска **помножи природним бројем**. Слично као што код бројева множење представља узастопно сабирање, исти је случај и овде и може се



наслутити да множење ниске бројем заправо представља њено понављање одређени број пута. На пример,

```
вредност израза "ba"*2 је "baba".
```

Сада пробај да одговорип на прво питање у [интерактивном уџбенику](#) које се налази у секцији Надовезивање ниски.

Дужина ниске, издвајање делова ниске

За рад са нискама Python нуди више уграђених функција. Једна од њих је `len`, функција којом се одређује број карактера ниске, тј. дужина ниске. Дужину ниске тј. број њених карактера можемо добити помоћу функције `len`. Тако је

```
len("Zdravo") једнако 6,
```

јер ниска "Zdravo" има тачно 6 карактера. Своје разумевање ове функције провери тако што ћеш одговорити на питања у [интерактивном уџбенику](#) која се налазе у секцији Дужина ниске, издвајање делова ниске.

Карактери у ниски имају своје редне бројеве тј. **позиције**. Први карактер се налази на позицији 0, други на позицији 1 и тако даље. На пример, карактери у ниски "Zdravo svima!" се броје на следећи начин.

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| Z | d | r | a | v | o | | s | v | i | m | a | ! |

Могуће је **издвојити појединачни карактер из ниске**. На пример, ако је

```
ime = "Zorana"
```

тада се

```
карактер Z може добити изразом ime[0],  
а карактер r изразом ime[2].
```

Подржани су и негативни индекси тако што -1 означава последњи карактер, -2 претпоследњи и тако даље.

| | | | | | | | | | | | | |
|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|
| -13 | -12 | -11 | -10 | -9 | -8 | -7 | -6 | -5 | -4 | -3 | -2 | -1 |
| Z | d | r | a | v | o | | s | v | i | m | a | ! |



На пример, ако је

```
ime = "Zorana"
```

тада је

```
ime[-1] карактер а  
док је ime[-3] карактер r.
```

Још једна операција која је често корисна је **издвајање дела ниске**. Приликом издвајања може се навести распон позиција, при чему се издвајају карактери из тог распона рачунајући прву, а не рачунајући другу наведену позицију. На пример, `ime[2:5]` издваја карактере имена на позицијама 2, 3 и 4 (распон `[2:5]` је полуотворен тј. позиција 2 је урачуната, а позиција 5 није). Ако је

```
ime = "Predrag"
```

тада је

```
ime[2:5] једнако "edr".
```

Ако се горња граница не наведе, тада се издваја део ниске до њеног краја. Тако је

```
ime[3:] једнако "drag".
```

Претрага ниске

Често је потребно да проверимо да ли једна ниска садржи неки карактер или садржи неку другу ниску. То можеш урадити употребом функције **find**. На пример, нека је

```
Rec = "ponedeljak"
```

Тада је

```
Rec.find("d") једнако 4
```

```
Rec.find("c") једнако -1, јер траженог карактера нема у стрингу
```

```
Rec.find("e") једнако 3, јер функција враћа позицију прве појаве траженог карактера или тражене ниске
```

```
Rec.find("P") једнако -1, јер се велика мала слова сматрају различитим карактерима
```

**Задатак 2.**

Напиши програм у којем се на основу дате ниске која садржи име и презиме добијају ниска са именом и ниска с презименом те особе.

Да би решио проблем, за почетак одреди позицију размака, који у датој нисци раздваја име од презимена. Затим на основу те позиције издвој делове полазног стринга и то део до дате позиције за име и део који садржи карактере од првог наског размака па до краја стринга.

Предлог решења

```
1 ime_i_prezime = "Љубица Љубичић"  
2 razmak = ime_i_prezime.find(" ")  
3 ime = ime_i_prezime[0:razmak]  
4 prezime = ime_i_prezime[razmak+1:]  
5 print("Име: ", ime)  
6 print("Презиме: ", prezime)|  
7
```

Задатак 3.

Нинџа-корњача се креће кроз лавиринт тако што слуша упутства на тајном језику. Горе се на том језику каже `gif`, доле `el`, лево `ома`, а десно `lun`. Напиши програм који помаже корњачи да пронађе излаз из лавиринта.

Покушај да решиш овај задатак. Задатак можеш пронаћи и проверити сопствено решење у окружењу [интерактивног уџбеника](#).