

PROGRAMIRANJE 1

- **Ugrađena (gotova) funkcija** programskog jezika je imenovana programska cjelina namijenjena rješavanju određenih, većih ili manjih zadataka.
 - koristi se kao crna kutija: dovoljno je znati što radi, nije pretjerano bitno kako to radi.
- Funkcija se u istom programu, pa i u različitim programima, može pozvati po volji mnogo puta s različitim vrijednostima kao **parametrima**.
- Sam bi naziv funkcije trebao biti ključem njezina značenja i njezine svrhe.
- Upotreba funkcije naziva se pozivom funkcije:
`ime_funkcije(parametri)`
- Funkcija vraća **rezultat**.

- Ugrađene su funkcije svoje ime dobile po tome što ih možemo koristiti čim pokrenemo Python – one su ugrađene u osnovni kostur *Pythona*.
- Postoje funkcije koje su grupirane na temelju nekih zajedničkih svojstava u tzv. module. **Moduli** su pohranjeni u datoteke smještene u jedan posebni poddirektorij osnovnog direktorija *Pythona*
- Modul moramo najprije uvesti (importirati) - naredba **import**
- U ljesci se naredba utipka iza znaka *prompt*

```
>>> import math
>>> y = math.sqrt(16)
>>> y
4.0
```

```
>>> from math import sqrt
>>> print(sqrt(81))
9.0
```

Indeksiranje znakova stringa

- Pojedinim znakovima se može pristupiti pomoću indeksa
- Općeniti način korištenja indeksa:
<string>[<izraz>]

D	o	b	a	r		d	a	n
0	1	2	3	4	5	6	7	8

Ovdje ima 9 znakova, ali idu od nule...

```
a = "Dobar dan"
print(a[0])
print(a[0],a[2],a[4])
x = 6
print(a[x-2])
```

- Indeksiranjem dobijemo jedan znak iz stringa

- dohvaćanje znakova od kraja stringa prema njegovu početku pri čemu su indeksi negativni cijeli brojevi
- zadnji znak u nizu ima indeks -1, a prvi je jednak duljini stringa, ali s negativnim predznakom
- označimo li pozitivne indekse s i , a negativne s j , vrijedi da je $i - j = n$, ili $j = i - n$

```
>>> s = 'ABCDEFGH'
```

```
>>> s[-1]
```

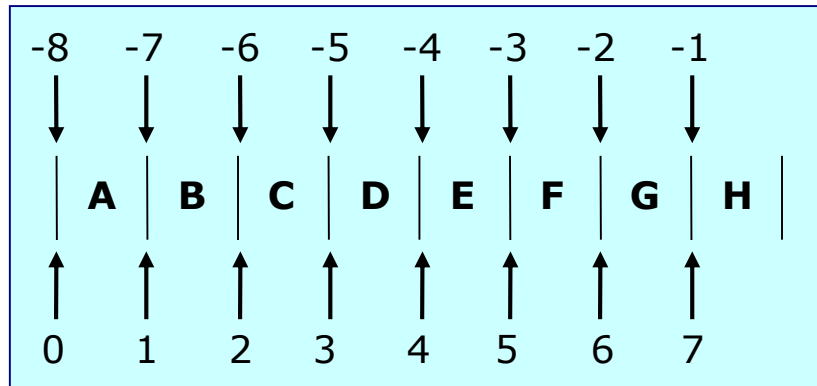
```
'H'
```

```
>>> s[-2]
```

```
'G'
```

```
>>> s[-8]
```

```
'A'
```



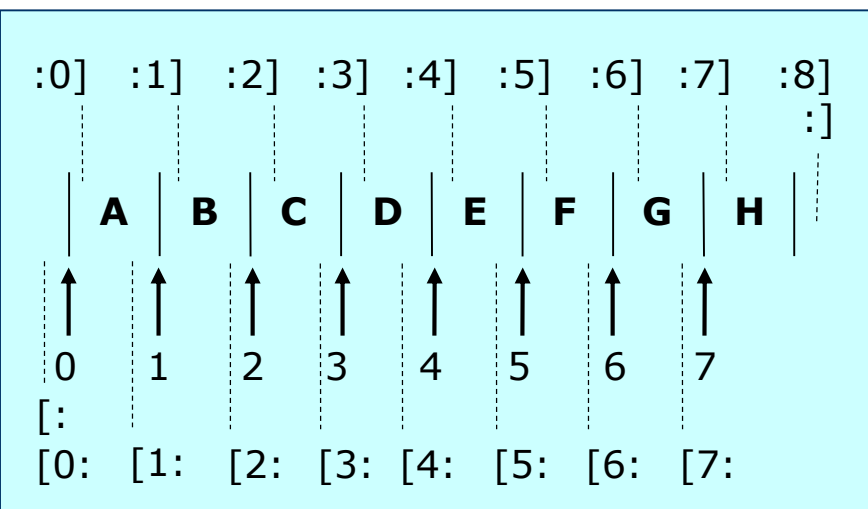
Isječak stringa - substring

- Možemo dobiti i substring, odnosno isječak (eng. slice):

`<string>[<pocetak>:<kraj>]`

Početna pozicija

Zadnja pozicija (**nije uključena u substring!**)



`[start:end]`
Indexes refer to places the knife "cuts."



Defaults are beginning of sequence and end of sequence.

- `a = "Dobar dan"`

Dob

- `print(a[0:3])`

bar d

- `print(a[2:7])`

- `print(a[:3])`

Dob

- `print(a[3:])`

ar dan

- `print(a[:])`

Dobar dan

- korak isijecanja

- uz korak jednak 1 u isječak ulaze svi znakovi redom
- uz korak jednak 2 u isječak ulazi svaki drugi znak
- uz korak 3 svaki treći znak itd.
- kada u isječku nema koraka, onda se prešutno pretpostavlja da je taj broj 1
- isijecanje počinje s indeksom koji je prvi napisan i zatim se u njega uključuju znakovi s razmakom određenim trećim brojem

```
>>> 'ABCDEFGH' [1:7:1]
```

```
'BCDEFG'
```

```
>>> 'ABCDEFGH' [1:7:2]
```

```
'BDF'
```

```
>>> 'ABCDEFGH' [::1]
```

```
'ABCDEFGH'
```

```
>>> 'ABCDEFGH' [::2]
```

```
'ACEG'
```


- korak može biti i s negativnim predznakom, što znači da će se indeksi smanjivati
- u isječak uzimaju znakovi počevši od većih indeksa prema manjim, pa to tako treba napisati i unutar uglate zagrade: najprije najveći indeks isječka i zatim donju granicu

```
>>> 'ABCDEFGH' [7:1:-1]
```

```
'HGFEDC'
```

```
>>> 'ABCDEFGH' [7:0:-1]
```

```
'HGFEDCB'
```

```
>>> 'ABCDEFGH' [7::-1]
```

```
'HGFEDCBA'
```

```
>>> 'ABCDEFGH' [::-1]
```

```
'HGFEDCBA'
```

Funkcije sa stringovima

Funkcija

Opis djelovanja

len(s)	vraća duljinu stringa
min(s)	vraća znak s najmanjom kodnom vrijednošću
max(s)	vraća znak s najvećom kodnom vrijednošću
ord(s)	vraća dekadni kod pojedinog znaka
chr(n)	vraća znak pojedinog dekadnog koda
str(n)	vraća znakovni prikaz broja n

- duljina stringa

```
>>> len('')
```

```
0
```

```
>>> len('A')
```

```
1
```

```
>>> len('ABCD')
```

```
4
```

```
>>> len('ABCDEFGH')
```

```
8
```

Python broji znakove u stringu od 0

```
>>> s = 'EGHFKMDABCL'
>>> len(s)
11
>>> min(s)
'A'
>>> max(s)
'M'
>>> ord('a')
97
>>> chr(65)
'A'
>>> str(65)
'65'
```

Metode sa stringovima

Ime metode	Način uporabe	Opis djelovanja
<code>center(w)</code>	<code>s.center(w)</code>	vraća kopiju stringa <code>s</code> centriranog u polju širine <code>w</code>
<code>ljust(w)</code>	<code>s.ljust(w)</code>	vraća kopiju stringa <code>s</code> smještenu lijevo u polju širine <code>w</code>
<code>rjust(w)</code>	<code>s.rjust(w)</code>	vraća kopiju stringa <code>s</code> smještenu desno u polju širine <code>w</code>
<code>capitalize()</code>	<code>s.capitalize()</code>	vraća kopiju stringa <code>s</code> s prvim velikom slovom
<code>lower()</code>	<code>s.lower()</code>	vraća kopiju stringa <code>s</code> sa svim malim slovima
<code>upper()</code>	<code>s.upper()</code>	vraća kopiju stringa <code>s</code> sa svim velikim slovima
<code>replace(stari, novi)</code>	<code>s.replace(stari, novi)</code>	vraća kopiju stringa <code>s</code> u kojem su sve pojave podstringa <code>stari</code> zamijenjene podstringom <code>novi</code>
<code>strip()</code>	<code>s.strip()</code>	vraća kopiju stringa <code>s</code> iz kojeg su izbačeni znakovi, navedeni kao argumenti metode, sa početka i kraja stringa, (ukoliko nema navedenih argumenata podrazumijvaju se razmaci)
<code>index(s1)</code>	<code>s.index(s1)</code>	vraća poziciju prvog pojavljivanja stringa <code>s1</code> u zadanom stringu <code>s</code>

- “ulančavanje metoda”

- kada neka metoda kao rezultat daje string onda se na taj novi string može opet primijeniti metoda i to tako da metode „ulančujemo“

```
>>> s_1 = 'centrirani tekst'
>>> s_2 = 'lijevo smješteni tekst'
>>> s_3 = 'desno smješteni tekst'
>>> s_1.upper().center(50)
'                CENTRIRANI TEKST                '
>>> (s_2.capitalize() + '.').ljust(50)
'Lijevo smješteni tekst.'
>>> (s_3.capitalize() + '.').rjust(50)
'                Desno smješteni tekst.'
```

Funkcije s brojevima

- **ceil(x)** – izračunava najveću cjelobrojnu vrijednost koja je manja ili jednaka x
- % - daje ostatak kod cjelobrojnog dijeljenja
- **abs(x)** – daje apsolutnu vrijednost broja
- **round(broj,decimale)**
zaokružuje broj na definirani broj decimala
- **round(broj)**
zaokružuje broj (bez decimala)
round(3.9) → 4
- **int(broj)**
pretvara u cijeli broj tako da se odbaci decimalni dio
int(3.9) → 3

Funkcija	Opis	Primjer
<code>min(a, b, c, ...)</code>	vraća minimalnu vrijednost niza brojeva a, b, c, \dots	<code>min(2, 1, 5) = 1</code>
<code>max(a, b, c, ...)</code>	vraća maksimalnu vrijednost niza brojeva a, b, c, \dots	<code>max(2, 1, 5) = 5</code>
<code>pow(a, b)</code>	vraća potenciju $a^{**}b$ (jednako kao i izraz $a^{**}b$)	<code>pow(2, 5) = 32</code>
<code>pow(a, b, n)</code>	vraća vrijednost $((a^{**}b) \% n)$ – ostatak dijeljenja potencije $a^{**}b$ s brojem n	<code>pow(123, 98, 19) = 17</code>

Matematičke funkcije

- $\text{sqrt}(x)$ – korijen broja
- Trigonometrijske: $\sin(x)$, $\cos(x)$, $\text{atan}(x)$, $\tan(x)$

Funkcija	Opis	Primjer
<code>ceil(x)</code>	Najmanji cijeli broj veći ili jednak x	<code>ceil(3.4) = 4</code> <code>ceil(4.9) = 5</code>
<code>floor(x)</code>	Najveći cijeli broj manji ili jednak od x	<code>floor(3.2) = 3</code> <code>floor(7.8) = 7</code>
<code>exp(x)</code>	e^x	<code>exp(1) = 2.71828</code>
<code>log(x, b)</code>	$\log_b x$	<code>log(100, 10) = 2.0</code>
<code>sin(x)</code>	$\sin x$	
<code>cos(x)</code>	$\cos x$	
<code>tan(x)</code>	$\text{tg } x$	
<code>asin(x)</code>	Arcus sinus	
<code>acos(x)</code>	Arcus kosinus	

- Program koji će učitati jednu riječ i ispisati je naopako (bez primjene isječka).

S unce

ecnus

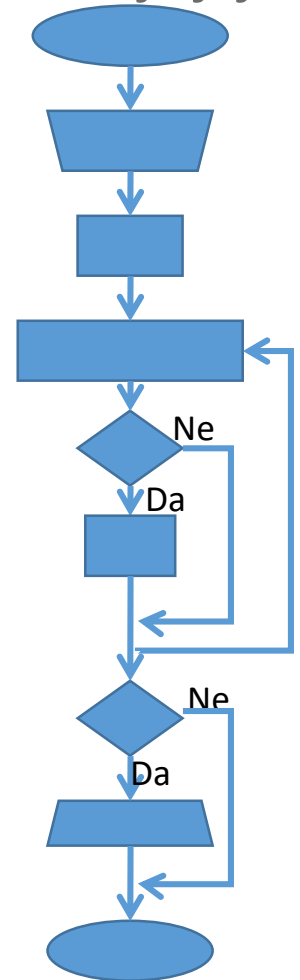
```
unesi (riječ)
b=""
za i=(duljina unesene rijeci) do 1 korak -1
    b=b+i.slovo
ispis(b)
```

```
a=input()
b=""
for i in range (len(a)-1,-1,-1):
    b=b+a[i]
print(b)
```

- Upisati riječ. Ispisati tu riječ na ekran samo ako joj je duljina prost broj.

```
unesi (riječ)
br=duljina unesene rijeci
f=0
za i=2 do br-1
    ako je (br MOD i = 0) onda
        f=1
ako je (f=0) onda
    ispis (riječ)
```

```
a=input()
br=len(a)
f=0
for i in range(2,br):
    if (br%i==0):
        f=1
if (f==0):
    print (a)
```



- Obavezno pročitati do sljedećih vježbi i predavanja:
“Rješavanje problema programiranjem u Pythonu”
stranice: 84-91; 144-147; 169-170; 177-180

Na kolokviju nije dozvoljeno koristiti: `split()`, `rsplit()`, `splitfields()`, `find()`, `rfind()`, `index()`, `rindex()`, `count()`, `swapcase()`, `endswith()`, `partition()`, `startswith()`, `min()`, `max()`, `sort()`, `sum()`, `reverse()`