

## Beleške sa časa – 4.nedelja

### Terminologija

niska = niz karaktera = string = tekstualni podatak

podniska = podniz niza karaktera = podstring (eng. *substring*)

imutabilnost = nepromenljivost (stringova)

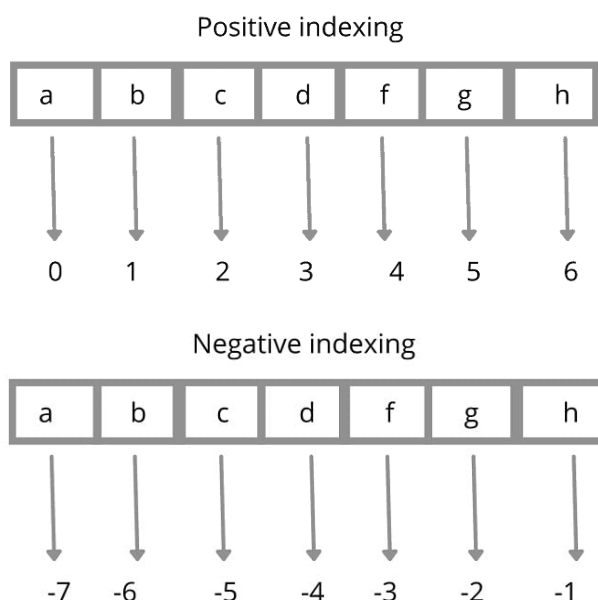
### Indeksiranje stringova

- u Python-u se može koristiti pozitivno ili negativno indeksiranje karaktera stringa

- pozitivno indeksiranje se koristi za indeksiranje stringa s leva na desno (u pozitivnom smeru), a negativno indeksiranje se koristi za indeksiranje stringa s desna na levo (u negativnom smeru)

- kod pozitivnog indeksiranja indeks prvog karaktera je **0**, indeks drugog karaktera je **1**, ..., indeks poslednjeg karaktera je za jedan manji od dužine stringa

- kod negativnog indeksiranja indeks poslednjeg karaktera je **-1**, indeks pretposlednjeg karaktera je **-2**, ..., indeks prvog karaktera je jednak negativnoj vrednosti dužine stringa



### Operator indeksiranja

- operator indeksiranja se koristi za izdvajanje pojedinačnih karaktera stringa

- operator indeksiranja se označava sa uglastim zagradama između kojih se navodi indeks karaktera koji želite da izdvojite

- indeks karaktera se može navesti eksplicitno (pozitivno ili negativno indeksiranje) ili da se izračunava kao rezultat nekog aritmetičkog izraza ili neke funkcije (npr. **len** funkcije) ili kombinacije aritmetičkog izraza i funkcije

### Imutabilnost stringova

- jedna od važnih karakteristika string tipa jeste imutabilnost (eng. *immutable*, nepromenljivo)

- imutabilnost stringova podrazumeva da kada se jednom kreira objekat tipa string i njegov sadržaj se više ne može menjati (niska, tekstualni podatak)

- to znači da ukoliko nam treba string objekat koji se neznatno razlikuje u odnosu na prethodno kreirani string objekat, to moramo uraditi tako što kreiramo novi string objekat kome će vrednost biti pridružena na osnovu tog već postojećeg string objekta

- **VAŽNO:** ne treba poistovećivati promenljivu kojoj je pridružen string objekat i sam string objekat; naime, jednoj istoj promenljivoj se mogu pridruživati različiti string objekti, tj. vrednost promenljive može biti promenjena, ali jedan isti string objekat ne može menjati vrednost (sadržaj) zbog svojstva imutabilnosti

- **vrednost stringa ≠ vrednost stringovske promenljive (promenljive kojoj je pridružen string objekat)**

- dakle, string na koji je primenjen operator indeksiranja se može naći isključivo sa desne strane jednakosti

- uostalom, promenljivoj kojoj je jednom pridružen string objekat ne samo da se može promeniti vrednost pridruživanjem drugog string objekat (sa drugačijim sadržajem), već se toj promenljivoj može pridružiti i objekat bilo kog drugog tipa (čas 3)

### Operator izdvajanja

- operator izdvajanja (eng. *slicing operator*) se koristi za izdvajanje podstringova
- dva oblika operatora izdvajanja:

```
substring = s[start : end]  
substring = s[start : end : step]
```

pri čemu svaki od parametara operatora izdvajanja ima posebno značenje kao i podrazumevanu vrednost:

**start** – indeks početnog karaktera podstringa (podrazumevano **0**)

**end** – indeks za jedan veći od poslednjeg karaktera podstringa (podrazumevano **len(s)**)

**step** – veličina koraka sa kojim želimo da se vrši izdvajanje karaktera podstringa (podrazumevano **1**)

- kao i kod operatora indeksiranja, svaki od parametara (indeks karaktera početka/kraja ili korak izdvajanja) se može navesti eksplicitno (pozitivno ili negativno indeksiranje) ili da se izračunava kao rezultat nekog aritmetičkog izraza ili neke funkcije (npr. **len** funkcije) ili kombinacije aritmetičkog izraza i funkcije
- u obe varijante operatora izdvajanja neki od parametara ili svi parametri mogu biti izostavljeni, i tada se koriste njihove podrazumevane vrednosti

### Metodi klase **str**

- klasa koja definiše tip **str** sadrži razne zgodne metode za rad sa stringovima
- spisak svih metoda definisanih klasom **str** možete naći na sledećem linku:

[https://www.w3schools.com/python/python\\_ref\\_string.asp](https://www.w3schools.com/python/python_ref_string.asp)

- neki od najčešće korišćenih metoda su:

- **islower** – proverava da li su svi slova alfabeta koji se pojavljuju u datom stringu mala slova; vraća **True** ili **False**
- **isupper** – proverava da li su svi slova alfabeta koji se pojavljuju u datom stringu velika slova; vraća **True** ili **False**
- **lower** – vraća string koji sadrži sve iste karaktere kao i dati string, s tim što su sva slova alfabeta konvertovana u mala slova
- **upper** – vraća string koji sadrži sve iste karaktere kao i dati string, s tim što su sva slova alfabeta konvertovana u velika slova
- **isalpha** – proverava da li su svi karakteri datog stringa slova alfabeta; vraća **True** ili **False**
- **isdecimal**, **isnumeric** – proverava da li su svi karakteri datog stringa slova cifre; vraća **True** ili **False**
- **startswith** – proverava da li dati string (nad kojim se poziva metod) počinje datim podstringom (zadat kao argument metoda); vraća **True** ili **False**
- **endswith** – proverava da li se dati string (nad kojim se poziva metod) završava datim podstringom (zadat kao argument metoda); vraća **True** ili **False**
- **index** – pokušava da pronađe dati podstring (zadat kao argument metoda) u okviru datog stringa (nad kojim se poziva metod); vraća indeks pozicije na kojoj počinje traženi podstring u datom stringu, u suprotnom izaziva **ValueError** grešku
- **find** – pokušava da pronađe dati podstring (zadat kao argument metoda) u okviru datog stringa (nad kojim se poziva metod); vraća indeks pozicije na kojoj počinje traženi podstring u datom stringu, u suprotnom **-1** (nevalidni indeks)
- **count** – vraća broj pojavljivanja traženog podstringa (zadat kao argument metoda) u datom stringu (nad kojim se poziva metod)
- **replace** – vraća string koji se dobija zamenom svih pojavljivanja traženog podstringa u datom stringu drugim podstringom (traženi podstring i zamenski podstring se zadaju kao argumenti metoda)
- **split** – vrši podelu datog stringa na delove, pri čemu se za mesto podele uzima svako pojavljivanje datog podstringa (zadat kao argument metoda); vraća listu dobijenih delova datog stringa

### Aritmetički operatori

- u Python-u postoji sedam aritmetičkih operatora: +, -, \*, /, //, %, \*\*

- aritmetički operatori poredani po prioritetu, počev od operatora najvišeg prioriteta ka operatorima nižeg prioriteta:

1. \*\*
2. \*, /, //, %
3. +, -

### Logički operatori

- u Python-u postoje tri logička operatora: **not**, **and**, **or**

- logički operatori poredani po prioritetu, počev od operatora najvišeg prioriteta ka operatorima nižeg prioriteta:

1. **not**
2. **and**
3. **or**

x	not x
True	False
False	True

x	y	x and y
True	True	True
True	False	False
False	True	False
False	False	False

x	y	x or y
True	True	True
True	False	True
False	True	True
False	False	False

### Relacioni operatori

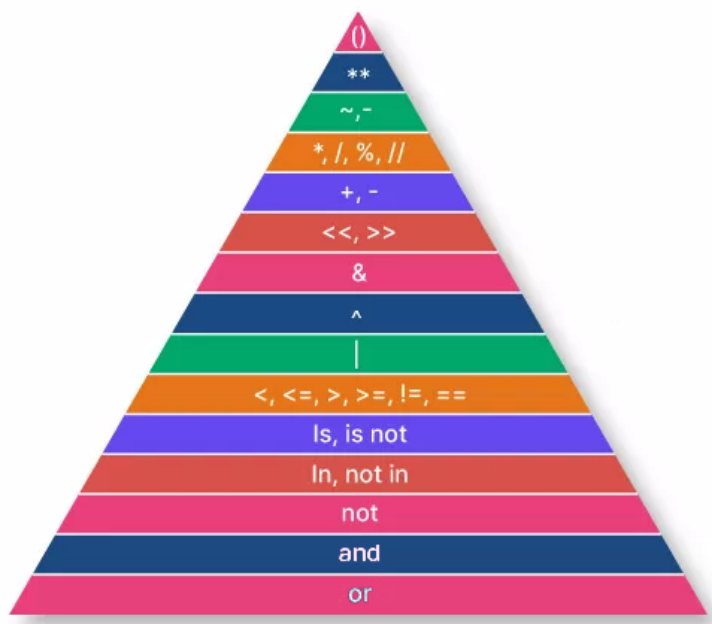
- u Python-u postoji 6 relacionih operatora: >, <, >=, <=, ==, !=

- svi relacioni operatori imaju jednak prioritet

- Python dozvoljava da se nadovezuju operatori poređenja u izrazima, i tada se primenjuje sledeći princip za izračunavanje: npr.  $x < y <= z$  izračunava se kao  $x < y$  and  $y <= z$

### Prioritet svih operatora u Python-u

- kada su operatori jednakog prioriteta, tada redosled primene operacija određuje asocijativnost operatora



## Biblioteka **math**

- biblioteka za rad sa matematičkim funkcijama i matematičkim konstantama (npr.  $\pi$ ,  $e$ ,...)
- da bi se mogle koristiti funkcije i konstante iz **math** biblioteke (modula) potrebno ih je uvesti u program pomoću naredbe **import**
- uvoženje biblioteke (modula) podrazumeva da se sadržaj te biblioteke (definicije funkcija i konstanti) stavljaju na raspolaganje za korišćenje u tekućem programu
- spisak svih funkcija i konstanti definisanih u biblioteci **math** možete naći na sledećem linku:  
[https://www.w3schools.com/python/module\\_math.asp](https://www.w3schools.com/python/module_math.asp)
- neki od najčešće korišćenih funkcija **math** biblioteke su:
  - **math.fabs(x)** – apsolutna vrednost od  $x$
  - **math.floor(x)** – nabliži manji ceo broj od broja  $x$
  - **math.ceil(x)** – najbliži veći ceo od broja  $x$
  - **math.exp(x)** – računa  $e^x$
  - **math.log(x)** – računa  $\ln(x)$ , za  $x > 0$
  - **math.log10(x)** – računa  $\log_{10}(x)$ , za  $x > 0$
  - **math.pow(x, y)** – računa  $x^y$
  - **math.sqrt(x)** – računa kvadratni koren od  $x$
  - **math.sin(x)**, **math.cos(x)**, **math.tan(x)**, **math.atan(x)**,... – razne trigonometrijske funkcije
- najčešće korišćene konstante **math** biblioteke su:
  - **math.pi**
  - **math.e**
  - **math.inf**
- ekvivalenti nekih od funkcija iz **math** biblioteke koji su dostupni u standardnoj biblioteci:
  - **abs(x)** – apsolutna vrednost od  $x$
  - **pow(x, y)** – računa  $x^y$
  - **round(x, n)** –  $x$  zaokružen na  $n$  decimalnih cifara iza decimalne tačke
  - **min(x1, x2, ..., xn)** – funkcija sa neograničenim brojem argumenata; računa minimum navedenih vrednosti
  - **max(x1, x2, ..., xn)** – funkcija sa neograničenim brojem argumenata; računa maksimum navedenih vrednosti