(PTIA1201) Elemi programozás

Dr. Facskó Gábor, PhD
tudományos főmunkatárs
facskog@gamma.ttk.pte.hu

Pécsi Tudományegyetem, Természettudományi Kar, Matematikai és Informatikai Intézet, 7624 Pécs, Ifjúság utja 6.
Wigner Fizikai Kutatóközpont, Űrfizikai és Űrtechnikai Osztály, 1121 Budapest, Konkoly-Thege Miklós út 29-33.
`https://facsko.ttk.pte.hu`

2024. szeptember 20.

Programozási tételek

- Maximum/minimum kiválasztás
- Elem kiválasztása
- Megszámlálás
- Tömb elemeinek összege
- Buborék rendezés
- Quickshort

String kezelés I

- A forráskódban megengedett több egymás utáni string literál, whitespace-el elválasztva (pl.: "hello" 'world'). Ez ekvivalens a két literál konkatenációjával ("helloworld"). Fontos tudni, hogy az e fajta összefűzés fordítási időben történik. Futási időben ez a konkatenáció nem működik, helyette a + operátor használandó.
- Mivel a Stringek immutábilis sorozatok, ezért az ezekre értelmezett műveletek természetesen használhatók a karakterláncokra is.
  - x in str Igaz, ha x strning része az str string-nek (substring)
  - s + t s és t stringek konkatenációja
  - s * n, n * s s replikálása n-szer, egymáshoz konkatenálva
  - s[i] s string i-edik karaktere (0 az első)
  - s[i:j] s string i-től j-ig tartó szelete
  - s[i:j:k] s i-től j-ig tartó szelete, k lépésközzel
  - len(s) s hossza
  - min(s) s-ben a legkisebb ASCII kódú karakter
  - max(s) s-ben a legnagyobb ASCII kódú karakter

String kezelés II

- String metódusok
  - str.capitalize() Visszatér str másolatával úgy, hogy az első karaktert nagybetűssé alakítja
  - str.center(width[, fillchar]) Visszatér str középre igazított, width hosszúságú másolatával. A kitöltő karaktert az opcionális fillchar határozza meg (alapértelmezettként szóköz).
  - str.count(sub[start[, end]]) Visszatér sub előfordulásainak számával str-ben, a start-tól az end pozícióig.
  - str.decode([encoding[, errors]]) A megadott encoding objektum által kódolja a string-et.
  - str.endswith(suffix[, start[, end]]) Igazzal tér vissza, ha str suffix stringre végződik. A suffix lehet egy stringehet tartalmazó tuple is, ebben az esteben igazat ad a metódus, ha valamelyik eleme szuffixe az str stringnek. A tesztelés az opcionális start paraméternél kezdődik és az opcionális end paramétrenél ér véget.

String kezelés III

- str.expandtabs([tabsize]) Visszatér str egy olyan másolatával, ahol minden tab karakter egy vagy több szóköz karakterre van lecserélve. A szóközök számát az opcionális tabsize paraméter határozza meg (alapértelmezettként 8)
- str.find(sub[, start[, end]]) Visszaadja azt a legkisebb indexet, ahonnan kezdve sub előfordul az str stringben. -1-gyel tér vissza, ha sub nem található str-ben.
- str.format(*args, **kwargs) str-en végrehajt egy string formázási műveletet és visszaadja a formázott szöveget. A formázandó szöveg literálokból állhat, illetve -vel határolt kicserélendő mezőkből. Minden egyes kicserélendő mező vagy egy pozicionális argumentum indexét, vagy egy kulcsszavas argumentum nevét tartalmazza.
  >>> "The sum of 1 + 2 is 0".format(1+2)
  'The sum of 1 + 2 is 3'
- str.index(sub[, start[, end]]) Ugyanaz, mint a find(), de ValueError kivételt dob ha a string nem volt megtalálható.

String kezelés IV

- str.isalnum() Igazzal tér vissza, ha a stringben minden karakter alfanumerikus, és legalább egy karakter hosszú.
- str.isalpha() Igazzal tér vissza, ha a stringben minden karakter alfabetikus, és legalább egy hosszú.
- str.isspace() Igazzal tér vissza, ha a strig csak whitespace karaktereket tartalmaz és legalább egyet.
- str.istitle() Igazzal tér vissza, ha a string címszerű, azaz minden szava nagybetűvel kezdődik, és legalább egy karaktert tartalmaz.
- str.isupper() Igazzal tér vissza, ha a string minden karaktere nagybetűs és legalább egy hosszú.
- str.join(iterable) Visszatér egy olyan stringgel, amely az iterable objektumból való stringek konkatenációja. A stringek közötti elválasztó az str string.
  >>> ", ".join(["alma", "korte", "dio"])
  "alma, korte, dio

String kezelés V

- str.ljust(width[, fillchar]) Visszatér azzal a width hosszú stringgel, amelyben str balra van igazítva. Az opcionális fillchar paraméter határozza meg a kitöltő karaktert.
- str.lower() Visszatér azzal a stringgel, amelyben str minden karaktere kisbetűs.
- str.lstrip([chars]) Visszatér str azon másolatával, ahol a string eleji karakterek el vannak távolítva. A chars paraméter egy string, amely az eltávolítandó karaktereket tartalmazza. Ha hiányzik, akkor a metódus csak a whitespace karaktereket távolítja el.
- str.partition(sep) Patricionálja str stringet az első sep elválasztónál, és visszaad egy 3 elemű tuple-t, ahol az első elem a sep előtti, a második maga a sep, a harmadik pedig a sep utáni rész. Ha sep nem része str-nek, akkor a 3 elemű tuple első eleme maga a teljes str lesz, a további két elem pedig üres string.
- str.replace(old, new[, count]) Visszatér str azon másolatával, ahol old összes előfordulása le van cserélve new-re. Az opcionális count paraméter azt határozza meg, hogy legfeljebb hány előfordulás legyen lecserélve.

String kezelés VI

- str.rfind(sub[, start[, end]]) Visszaadja azt a legnagyobb indexet, amelytől kezdődően sub megtalálható az str stringben. Az opcionális start és end paraméterek segítségével str adott szeletében kereshetünk.
- str.rindex(sub[, start[, end]]) Ugyanaz, mint a rfind(), de ValueError kivételt dob, ha a string nem volt megtalálható.
- str.rjust(width[, fillchar]) Visszaadja az adott stringet jobbra igaítva egy width hosszúságú stringben. Az opcionális fillchar paraméter határozza meg a kitöltő karaktert (alapértelmezett: szóköz).
- str.rpartition(sep) Patricionálja str stringet az utolsó sep elválasztónál, és visszaad egy 3 elemű tuple-t, ahol az első elem a sep előtti, a második maga a sep, a harmadik pedig a sep utáni rész. Ha sep nem része str-nek, akkor a 3 elemű tuple első eleme maga a teljes str lesz, a további két elem pedig üres string.
- str.rsplit([sep[, maxsplit]]) Visszatér az adott string szavait tartalmazó listával, ahol sep a szavakat elválasztó jel(sorozat). Ha a maxsplit paraméter meg van adva, akkor a metódus jobbról kezdve legfeljebb maxsplit szeletelést végez.

String kezelés VII

- str.rstrip([chars]) Visszatér str azon másolatával, ahol a string végi karakterek el vannak távolítva. A chars paraméter egy string, amely az eltávolítandó karaktereket tartalmazza. Ha hiányzik, akkor a metódus csak a whitespace karaktereket távolítja el.
- str.split([sep[, maxsplit]]) Visszatér az adott string szavait tartalmazó listával, ahol sep a szavakat elválasztó jel(sorozat). Ha a maxsplit paraméter meg van adva, akkor a metódus balról kezdve legfeljebb maxsplit szeletelést végez.
- str.splitlines([keepends]) Feldarabolja az str stringet a sor határoknál, és visszatér a sorokat tartalmazó listával. Az egyes sorok a sorvége jeleket nem tartalmazzák, hacsak a keepends opcionális paraméter nem True igazságértékű.
- str.startswith(prefix[, start[, end]]) Igazzal tér vissza, ha str a prefix string-gel kezdődik, ellenkező eseten hamissal. Ha az opcionális start ill. end paraméter adott, akkor az összehasonlítás a start indexnél kezdődik, illetve az end indexnél fejeződik be.

String kezelés VIII

- str.strip([chars]) Visszatér str azon másolatával, ahol a string eleji és végi karakterek el vannak távolítva. A chars paraméter egy string, amely az eltávolítandó karaktereket tartalmazza. Ha hiányzik, akkor a metódus csak a whitespace karaktereket távolítja el.
- str.swapcase() Visszaadja azt a stringet, amelyben str kisbetűi nagybetűssé vannak konvertálva és fordítva.
- str.title() Visszaadja azt a stringet, amelyben str minden szava nagybetűvel kezdődik és a szavak további karaktere kisbetűs.
- str.translate(table[, deletechars]) Visszaadja str-nek azt a másolatát, amelyben minden karakter ki van törölve, amely szerepel a deletechars stringben, a maradék karakterek pedig le vannak képezve az adott konverziós tábla (table) által. A table kötelezően egy 256 hosszúságú string.
- str.upper() Visszadja str azon másolatát, amelyben a karakterek csupa nagybetűsek.

String kezelés IX

- str.zfill(width) Visszaadja az str numerikus stringnek azon másolatát, amely balról fel lett töltve annyi 0-val, hogy az új string width hosszúságú legyen. A metódus a negatív numerikus strinkeket is jól kezeli, továbbá ha width kisebb, mint az str hossza, akkor visszaadja az eredeti stringet.

- String formázás
  - A String és Unicode objecteknek van egy egyedi, beépített operátora: a % operátor. Ez az operátor az ún. string formázó operátor.

String kezelés X

- Az operátor működése megegyezik a C-ből ismert sprintf() függvénnyel: a format % értékek utasítás az értékeket kiértékei a format operandus konverziós specifikációinak (mezőinek) megfelelően, majd behelyettesíti azokat. Ha a format egyetlen egy argumentumot kíván, akkor az értékek lehet egy nem-tuple objektum. Több argumentum esetén viszont kötelezően tuple az értékek operandus.
  >>> "The counter named %s is %02d" % ("clock", 3+2)
  "The counter named clock is 05"
  >>> "the counter named 0 is 1:02d".format("clock", 3+2)
  'the counter named clock is 05'
- A konverziós mező kettő, vagy több karakterből áll, és az alábbi elemekből állhat (ebben a sorrendben):
  - A '%' karakter, ami a specifikáció kezdetét jelöli. format() esetén ez a ':' karakter.
  - Egy opcionális mapping key, ami egy tetszőleges karaktersorozat zárójelben. Pl (someone)
  - Konverziós flag-ek (opcionális), amik a konverzió típusát és végeredményét befolyásolják.

String kezelés XI

- Minimális mező szélesség (opcionális). Ha a '*' karakterként van megadva, akkor az aktuális szélességet az értékek tuple következő eleme határozza meg.
- Pontosság (opcionális), amit egy '.' és egy azt követő szám határoz meg. Ha a '*' karakterként van megadva, akkor az aktuális pontosságot az értékek tuple következő eleme határozza meg.
- Hossz módosító (opcionális).
- Konverzió típusa.

- Ha a % operátor jobb argumentuma egy dictionary (vagy egyéb mapping típus), akkor a konverziós mezőknek kötelezően tartalmazniuk kell a zárójelezett mapping key-t. Pl.:
  >>> print '%(language)s has %(#)03d quote types.' % \
  ... {'language': "Python", "#": 2}
  Python has 002 quote types.
- Ebben az esetben a '*' karakter nem szerepelhet a specifikációkban, hiszen annak feldolgozása szekvenciális argumentumot kíván.

String kezelés XII

- A konverziós flagek:

Flag	Jelentés
'#'	A konverzió az adott konverziós típus alternatív formáját fogja használni. Lásd lejjebb.
'0'	A konverzió 0-val fogja balról kiegészíteni a numerikus stringet.
'-'	A konvertált érték balra igazított lesz. (Felülírja a '0' flag-et, ha az is adott).
' '	(szóköz) Pozitív numerikus string elé a konverzió betesz egy szóközt (előjeles konverziónál használatos).
'+'	Az előjel mindig megjelenik a konvertált stringben (felülírja a szóköz flag-et).

- A hossz módosító (h, l, L) jelen lehet, de a Python esteében ez figyelmen kívül van hagyva, tehát pl. a '%d' ugyanaz, mint a '%ld'. E módosító kizárólag a C-vel való kompatibilitás céljából került bele a Pythonba.

String kezelés XIII

- A konverzió típusok a következők:

Flag	Jelentés
'd'	Előjeles integer (decimális).
'i'	Előjeles integer (decimális).
'o'	Előjeles oktális érték.
'u'	Elavult típus, megegyezik 'd'-vel.
'x'	Előjeles hexadecimális (kisbetűs).
'X'	Előjeles hexadecimális (nagybetűs).
'e'	Lebegőpontos szám exponenciális alakban (kisbetűs).
'E'	Lebegőpontos szám exponenciális alakban (nagybetűs).
'f'	Lebegőpontos szám tizedestört alakban.
'F'	Lebegőpontos szám tizedestört alakban.
'g', 'G'	Lebegőpontos szám. Ha az exponens kisebb, mint -4 vagy nemkisebb, mint a

String kezelés XIV

'c'	Egy karakter. E típus paraméterként elfogad integer, illetve egyhosszú stringet is.
'r'	String. Bármely típusú Python objektumot elfogad, és a repr() függvény eredmén
's'	String. Bármely típusú Python objektumot elfogad, és az str() függvény eredmén
'%'	Nincs argumentum-konverzió, hanem egy '%' karaktert ír ki.

- Az oktális érték alternatív formája 0-kal van kitöltve a bal oldalon, ha a szám első számjegye nem nulla.
- A hexadecimális érték alternatív formái: x32fd helyett 0x32fd és X32fd helyett 0X32fd.
- A lebegőpontos számok alternatív formája mindig tartalmaz tizedespontot, még akokr is ha nem követi számjegy azt. Az alapértelmezett pontosság: 6.
- Stringeknél a pontosság azt határozza meg, hogy maximum milyen hosszú lehet a string.

Vége

Köszönöm a figyelmüket!