

12. KARAKTEREK A PYTHON PROGRAMOZÁSBAN

A programozásban sokszor előfordul, hogy karakterekkel kell dolgoznunk. A karakterek ASCII kódjuk, bájt nagyságú számok formájában tárolódnak. A lenti táblázatban egyértelműen látjuk, hogy melyik karakterhez melyik szám tartozik.



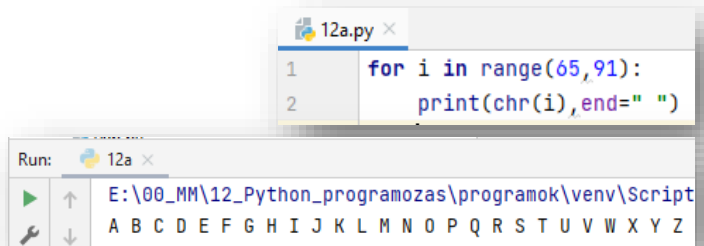
Megkaphatjuk **egy karakter ASCII kódját** az `ord` utasítással. Pl.: `A → 65`, `a → 97`,
 Egy beírt **számból a megfelelő ASCII karaktert** a `chr` utasítással állítjuk elő. pl.: `90 → Z`, `122 → z`

ASCII kódtábla (32-255):

32= 42=* 52=4 62=> 72=H 82=R 92=\	33=! 43=+ 53=5 63=? 73=I 83=S 93=]	34=" 44=, 54=6 64=@ 74=J 84=T 94=^	35=# 45=- 55=7 65=A 75=K 85=U 95=_	36=\$ 46=. 56=8 66=B 76=L 86=V 96=`	37=% 47=/ 57=9 67=C 77=M 87=W 97=a	38=& 48=0 58=: 68=D 78=N 88=X 98=b	39=' 49=1 59=; 69=E 79=O 89=Y 99=c	40=(50=2 60=< 70=F 80=P 90=Z 100=d	41=) 51=3 61== 71=G 81=Q 91=[101=e
102=f 112=p 122=z 132=, 142=Ž	103=g 113=q 123={ 133=... 143=	104=h 114=r 124= 134=† 144=	105=i 115=s 125=} 135=‡ 145='	106=j 116=t 126=~ 136=^ 146='	107=k 117=u 127=• 137=% 147="	108=l 118=v 128=€ 138=Š 148="	109=m 119=w 129=␣ 139=◀ 149=•	110=n 120=x 130=, 140=œ 150=--	111=o 121=y 131=f 141=␣ 151=---
152=˘ 162=¢ 172=¬	153=™ 163=£ 173=	154=š 164=¤ 174=®	155=› 165=¥ 175=¯	156=œ 166= 176=°	157=␣ 167=§ 177=±	158=ž 168="" 178=²	159=ÿ 169=© 179=³	160= 170=ª 180='	161=ı 171=« 181=µ
182=¶ 192=À 202=Ê	183=· 193=Á 203=Ë	184=¸ 194=Â 204=Ì	185=' 195=Ã 205=Í	186=° 196=Ä 206=Î	187=» 197=Å 207=Ï	188=¼ 198=Æ 208=Ð	189=½ 199=Ç 209=Ñ	190=¾ 200=È 210=Ò	191=¿ 201=É 211=Ó
212=Ô 222=Þ	213=Õ 223=ß	214=Ö 224=à	215=× 225=á	216=Ø 226=â	217=Ù 227=ã	218=Ú 228=ä	219=Û 229=å	220=Ü 230=æ	221=Ý 231=ç
232=è 242=ò	233=é 243=ó	234=ê 244=ô	235=ë 245=õ	236=ì 246=ö	237=í 247=÷	238=î 248=ø	239=ï 249=ù	240=ð 250=ú	241=ñ 251=û
252=ü	253=y	254=þ	255=ÿ						

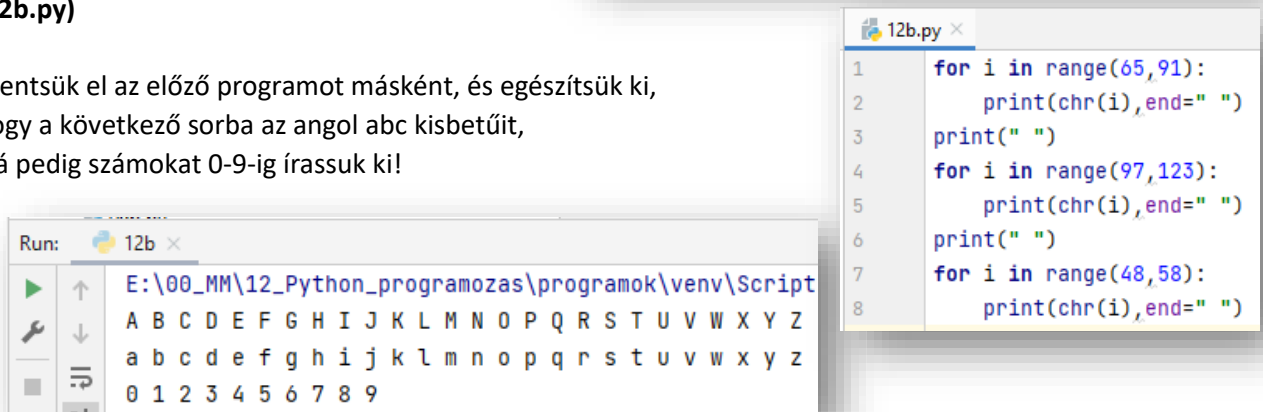
(12a.py)

Írassuk az angol ABC nagybetűit egymás mellé!



(12b.py)

Mentsük el az előző programot másként, és egészítsük ki, hogy a következő sorba az angol abc kisbetűit, alá pedig számokat 0-9-ig írassuk ki!



(12c.py)

Készítsünk programot, melyben egy begépelte karakterről eldöntjük, hogy nagybetű-e vagy nem!

```

12c.py x
1 betu=input("Adj meg egy karaktert: ")
2 if "A"<=betu<="Z":
3     print("A karakter nagybetű.")
4 else:
5     print("A karakter nem nagybetű.")

Run: 12c x
E:\00_MM\12_Python_program
Adj meg egy karaktert: F
A karakter nagybetű.
    
```

(12d.py)

Készítsünk programot, mellyel régi magyarországi rendszámot állítunk elő véletlenszerű karakterekből és számokból.

```

Run: 12d x
E:\00_MM\1
DCI-480
    
```

```

12d.py x
1 from random import *
2 x1=randrange(65,91)
3 x2=randrange(65,91)
4 x3=randrange(65,91)
5 sz1=randrange(0,10)
6 sz2=randrange(0,10)
7 sz3=randrange(0,10)
8 n1=chr(x1)
9 n2=chr(x2)
10 n3=chr(x3)
11 print(n1,n2,n3,"-",sz1,sz2,sz3, sep="")
    
```

(12e.py)

Általában decimális (tízes) számrendszert használunk a mindennapokban. Informatikában a decimális (kettes) számrendszer mellett a hexadecimális (tizenhatos) számrendszert is használjuk. Ebben 0-9-ig és A-F-ig használunk számokat és karaktereket.

Készítsünk programot, melyben bekérünk egy hexadecimális számot, majd kiírjuk a decimális értékét!

```

12e.py x
1 hexa=input("Kérek egy 0-F karaktert: ")
2 if hexa <= '9':
3     dec=ord(hexa)-48
4 else:
5     dec=ord(hexa)-55
6 print("Tízes számrendszerben: ",dec)

Run: 12e x
E:\00_MM\12_Python_programoz
Kérek egy 0-F karaktert: B
Tízes számrendszerben: 11
    
```

(12f.py)

A feladat az, hogy kérjen be egy szót, majd karakterenként tabulátorral elválasztva írassuk ki a betűket!

- Kérjünk be egy „szó” változóba egy szót (karakterláncot)!
- Majd a **len()** utasítással határozzunk meg a szó **karaktereinek számát!** (Milyen hosszú a karakterlánc.)
- For utasítással 0-tól hossz karakterszámig megy a ciklus!
- Majd kiírjuk a szó „i”-edik karakterit, tabulátorokkal elválasztva, a minta szerint.
- A „szó[i]” arra szolgál, hogy egyesével végig lépkedünk karakterenként.
- Az **end="\t"** utasítással tabulátorokkal tagoljuk a karaktereket:

```

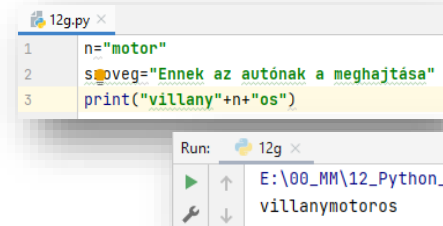
12f.py x
1 szo=input("Írjál be egyszót: ")
2 hossz=len(szo)
3 for i in range(hossz):
4     print(szo[i],end="\t")

Run: 12f x
E:\00_MM\12_Python_programozas\progra
Írjál be egyszót: Háromszög
H á r o m s z ö g
    
```

Ahogy már többször is használtuk, a karaktereket idézőjelek, vagy aposztrófok közé kell írni. (pl.: „villanymotor”, 'villanymotor').

(12g.py)

Nézzünk egy példát, amikor idézőjelben megadott szöveget és változóban tárolt **szöveget fűzünk össze plusz (+) műveleti jelekkel!**



```
1 n="motor"
2 szoveg="Ennek az autónak a meghajtása"
3 print("villany"+n+"os")
```

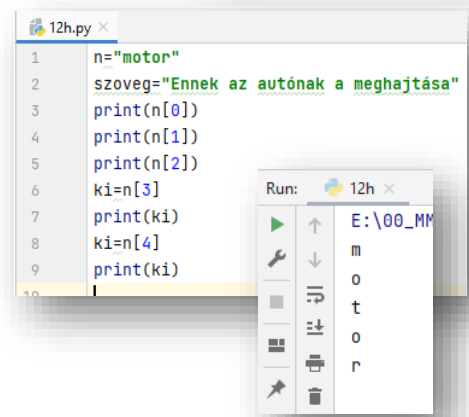
Run: 12g x
E:\00_MM\12_Python_\villanymotoros

(12h.py)

Ha hivatkozni szeretnének egy karakterlánc valahányadik karakterére akkor az **adott változó neve után négyzetes zárójelben megaduk, hogy hányadik karaktert szeretnénk felhasználni.**

Arra figyelni kell, hogy mindig nulla értéktől számolunk. Tehát az „n” változó első betűje az az **n[0]**, és print() utasítással ki tudjuk írni közvetlenül.

A példánkban az „n” harmadik karakterét beletesszük egy „ki” nevű változóba és az írattuk ki!



```
1 n="motor"
2 szoveg="Ennek az autónak a meghajtása"
3 print(n[0])
4 print(n[1])
5 print(n[2])
6 ki=n[3]
7 print(ki)
8 ki=n[4]
9 print(ki)
```

Run: 12h x
E:\00_MM\12_Python_\m
o
t
o
r

(12i.py)

Ebben a példában a változóinkban tárolt **karakterláncunk elejéről és végéről „vágunk” le felesleges részeket.**

A 3. sorban a „szoveg” változóban lévő karakterlánc első 16+1 karakterét tesszük bele a „vagott” változóba.

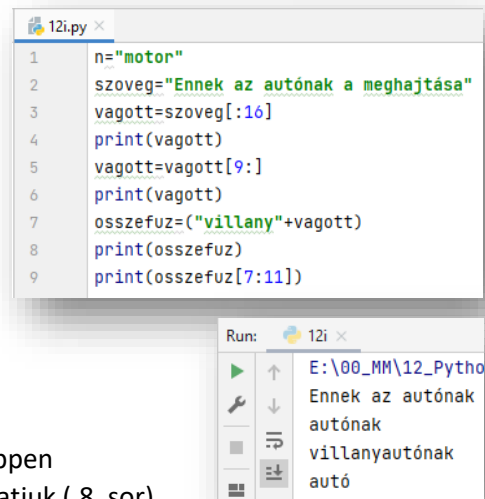
Amit a következő 4. sorban kiíratunk. (Ez a futtatásunk első sora.)

Majd az 5. sorban az éppen aktuálisan a „vagott” változóban lévő stringláncból (amit az előbb kiíratunk) használja fel a 9. karakter utáni részt, és most ez kerül be a „vagott” nevű

változóba! **Fontos tehát, hogy a kettőpontokat hova tesszük!**

A 7. sorban egy „osszefuz” változóba összefűzzük a „villany” és az éppen aktuálisan a „vagott” változóban lévő „autónak” szöveget. Majd kiíratjuk (.8. sor).

A 9. sorban látható módon pedig karakterszámtól, karakterszámgig írathatunk ki szöveget.



```
1 n="motor"
2 szoveg="Ennek az autónak a meghajtása"
3 vagott=szoveg[:16]
4 print(vagott)
5 vagott=vagott[9:]
6 print(vagott)
7 osszefuz=("villany"+vagott)
8 print(osszefuz)
9 print(osszefuz[7:11])
```

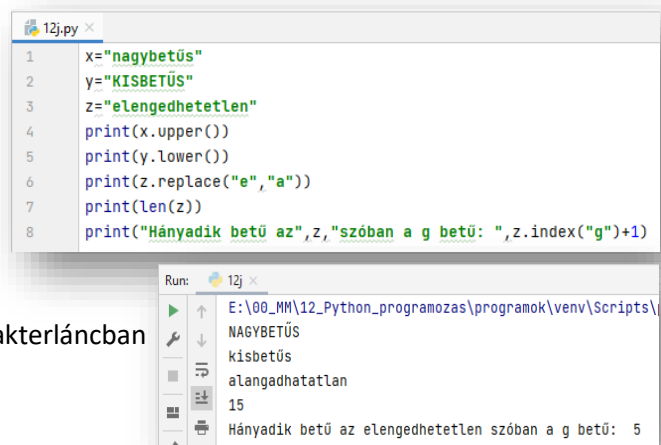
Run: 12i x
E:\00_MM\12_Python_\Ennek az autónak
autónak
villanyautónak
autó

(12j.py)

Végül egyetlen programban nézzünk meg több hasznos utasítást:

- **upper()** – nagybetűssé teszi a karakterláncot
- **lower()** – kisbetűssé teszi a karakterláncot
- **replace('x','y')** – cseréli a karkterket
- **len(z)** – visszaadja a karakter hosszát
- **index('n')** – megkeresi az első előfordulását a karakterláncban

A futtatott programban látszik az eredmény.



```
1 x="nagybetűs"
2 y="KISBETŰS"
3 z="elengedhetetlen"
4 print(x.upper())
5 print(y.lower())
6 print(z.replace("e","a"))
7 print(len(z))
8 print("Hányadik betű az",z,"szóban a g betű: ",z.index("g")+1)
```

Run: 12j x
E:\00_MM\12_Python_programozas\programok\venv\Scripts\
NAGYBETŰS
kisbetűs
alangedhatatlan
15
Hányadik betű az elengedhetetlen szóban a g betű: 5