



PYTHON ALAPOK

```
print("Hello, World!")
```

Tematika

Témakör	Óraszám
Előadások:	
Python programozás alapjai	9
Alkalmazások rekurzióra: Mohó algoritmusok, dinamikus programozás, backtrack algoritmus	3
Adatszerkezetek rendszerezése. Absztrakt társzerkezetek. Verem/sor. Statikus megvalósítás, dinamikus megvalósítás. Lengyel forma	3
Láncolt listák létrehozása Láncolt listák bejárása. Keresés, beszúrás, törlés, hulladékgyűjtés. Fejelt listák. Kétirányú listák	3
Fa adatszerkezetek. Fát reprezentáló adatstruktúrák. Adatmodell, eljárásmodell. Vektor, lista. Bináris fák. Megvalósításuk a memóriában. Bináris fák bejárása, keresés. Beszúrás, törlés. BST fák,	3
Rectori szünet	3
ZH	3
Kupac rendezés. Radix rendezés. Útvonalak hosszúsága, a Huffman algoritmus.	3
AVL fák. Általánosított fák, piros-fekete fa, B-fa.	3
Gráfok, Gráfalgoritmusok	3
A hasítótábla, mint adatszerkezet. Hasítás. Hasító függvények. Az összeütközések feloldása. Hasítás és keresőfák. Hasítás alkalmazásai	3
ZH pótlás	3

1 db elméleti ZH

Követelmények

- Az aláírás megszerzésének feltétele: A **két gépes ZH**, és az **elméleti ZH** megírása, a **házi feladat** beadása. A ZH-k és házifeladatból megszerezhető pontok 50%-nak teljesítése, vagyis a megszerezhető 120 pontból minimum 60 pontot kell elérni.
- **A vizsga jegy:** A gyakorlatokon összesen 100 pontot szerzhetnek. Az előadás anyagából a 8. héten egy 20 pontos elméleti ZH lesz.
- Összesen elérhető 120 pont. A vizsgán további 120 pont érhető el. A hozott pontszám egyszeres, a vizsga pontszám 2-szeres súllyal számít.

Osztályozás:

83% 5

75% 4

62% 3

50% 2

<50% 1

25% alatt letiltás

Könyv

- https://www.w3schools.com/python/python_operators.asp
- **Peter Wentworth, Jeffrey Elkner, Allen B. Downey and Chris Meyers: Hogyan gondolkozz úgy, mint egy informatikus: Tanulás Python 3 segítségével, 2019. március**
- **Cormen, Leiserson, and Stein Rivest. Új algoritmusok. 2003.**

Python általában

- Nagyon magas szintű programozási nyelv
- Interpretált nyelv-parancsértelmező
 - Interaktív mód >>> Python prompt
 - Script mód
- IDE (Integrated Development Environment)-nagyon sokféle
 - VS Code
 - Google Colab

Változók, típusok

and	as	assert	break	class	continue
def	del	elif	else	except	exec
finally	for	from	global	if	import
in	is	lambda	nonlocal	not	or
pass	raise	return	try	while	with
yield	True	False	None		

- Konstans (literál)-Változó
- Típusok type függvénnyel lekérdezhető
 - Text Type: **str**
 - Numeric Types: **int, float, complex**
 - Sequence Types: **list, tuple, range**
 - Mapping Type: **dict**
 - Set Types: **set, frozenset**
 - Boolean Type: **bool**
 - Binary Types: **bytes, bytearray, memoryview**
 - None Type: **NoneType**
- Foglalt szavak

Kifejezések

https://www.w3schools.com/python/python_operators

- Ugyanúgy, mint C#-ban
- Operátorok
- Precedencia

```
>>> 17
17
>>> y = 3.14
>>> x = len("helló")
>>> x
5
>>> y
3.14
```

Operator	Name	Example
+	Addition	x + y
-	Subtraction	x - y
*	Multiplication	x * y
/	Division	x / y
%	Modulus	x % y
**	Exponentiation	x ** y
//	Floor division	x // y

perator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
//=	x //= 3	x = x // 3
**=	x **= 3	x = x ** 3
&=	x &= 3	x = x & 3
=	x = 3	x = x 3
^=	x ^= 3	x = x ^ 3
>>=	x >>= 3	x = x >> 3
<<=	x <<= 3	x = x << 3
:=	print(x := 3)	x = 3 print(x)

Utasítások- adatbekérés

```
>>> n = input("Kérem, adja meg a nevét: ")
```


IF

Ami új

```
if BOOLEAN_KIFEJEZÉS:
    UTASÍTÁS_1          # Végrehajtódik, ha a feltétel kiértékelése igaz,
    ↳értéket ad
else:
    UTASÍTÁS_2          # Végrehajtódik, ha a feltétel kiértékelése hamis,
    ↳értéket ad
```

```
if x < y:
    UTASÍTÁSOK_A
elif x > y:
    UTASÍTÁSOK_B
else:
    UTASÍTÁSOK_C
```

FOR, WHILE - ITERÁCIÓ

```
for b in ["Misi", "Petra", "Botond", "Jani", "Csilla", "Peti", "Norbi"]:  
    meghivas = "Szia, " + b + "! Kérlek gyere el a bulimba szombaton!"  
    print(meghivas)
```

```
for i in [12, 16, 17, 24, 29]:  
    if i % 2 == 1: # Ha a szám páratlan ...  
        break    # ... azonnal hagyd el a ciklust  
    print(i)  
print("Kész.")
```

```
for i in [12, 16, 17, 24, 29, 30]:  
    if i % 2 == 1: # Ha a szám páratlan ...  
        continue # ... ne dolgozd fel!  
    print(i)  
print("Kész.")
```

```
def osszeg_eddig(n):  
    """ Számsorozat összege 1+2+3+...+n """  
    ro = 0  
    e = 1  
    while e <= n:  
        ro = ro + e  
        e = e + 1  
    return ro  
  
# A tesztkészlethez  
teszt(osszeg_eddig(4) == 10)
```

Függvények

```
def NÉV ( PARAMÉTEREK ) :  
    UTASÍTÁSOK
```

- `def negyzet_rajzolas(t, h):`
- `"""Egy h oldalhosszúságú négyzet rajzoltatása a t teknoccel"""`
- `for i in range(4):`
- `t.forward(h)`
- `t.left(90)`
- `#ez egy megjegyzés`
- Dokumentációs string- nem távolítja el a fordító

Egységteszt készítése

```
import sys

def teszt(sikeres_teszt):
    """ Egy teszt eredményének megjelenítése. """
    sorszam = sys._getframe(1).f_lineno    # A hívó sorának száma
    if sikeres_teszt:
        msg = "A(z) {0}. sorban álló teszt sikeres.".format(sorszam)
    else:
        msg = ("A(z) {0}. sorban álló teszt SIKERTELEN.".format(sorszam))
    print(msg)
```

```
x = "welcome"
```

```
#if condition returns False, AssertionError is raised:
assert x != "hello", "x should be 'hello'"
```

m format(self, args, kwargs)	str
_abc_cache	
_abc_negative_cache	
_abc_negative_cache_version	
_abc_registry	
m center(self, width, fillchar)	str
m count(self, x, __start, __end)	str
m encode(self, encoding, errors)	str
m endswith(self, suffix, start, end)	str
m expandtabs(self, tabsize)	str
m find(self, sub, __start, __end)	str
m format_map(self, map)	str
m index(self, sub, __start, __end)	str
m isalnum(self)	str
m isalpha(self)	str
m isdecimal(self)	str
m isdigit(self)	str
m isidentifier(self)	str
m islower(self)	str
m isnumeric(self)	str
m isprintable(self)	str
m isspace(self)	str
m istitle(self)	str
m isupper(self)	str
m join(self, iterable)	str

Stringek

- Szövegkezelő függvények
- Figyelem! A stringek megváltoztathatatlanok!
- Stringek formázása- formátumstring hasonló a C#-hoz.

File-kezelés

- Elv:
 - Megnyitás: Memória változó és a tényleges file összerendelése, file használat körének megadása
 - Használat: hasonló mint a standard I/O használata
 - Bezárás
- Utasítások: https://www.w3schools.com/python/python_file_handling.asp
- Példa

```
sajat_fajl = open("elso.txt", "w")
sajat_fajl.write("Az első Python fájlom!\n")
sajat_fajl.write("-----\n")
sajat_fajl.write("Helló, világ!\n")
sajat_fajl.close()
```

File-típusok – file használat módjai

- File típusok
 - Bináris állományok - pl. kép, hang stb. bináris módon megnyitva alacsony szinten byte-onként olvashatók, írhatók
 - Szöveges állományok – tartalma változó hosszúságú sorokból álló stringek- szekvenciálisan használható – akár soronkénti olvasással, de nem lehet „ugrálni”
 - Adatállományok – rögzített szerkezet – kereshető
- File használat
 - Pufferelten történik
 - Operációs rendszer kezeli és időzíti
 - Veszélye művelet – sok hiba lehetőséggel
 - Kivételeket gondosan kezelni!!!!

Összetettebb példa

- Érdemes minden sort külön-külön átgondolni:

```
uj_kezelo = open("elso.txt", "r")  # Nyisd meg a fájlt
while True:
    sor = uj_kezelo.readline()      # Próbáld beolvasni a következő sort
    if len(sor) == 0:               # Ha nincs több sor
        break                       # Hagyd el a ciklust
    # Most dolgozd fel az éppen aktuálisan beolvasott sort
    print(sor, end="")

uj_kezelo.close()                  # Zárd be a fájlt
```


File beolvasása listába

```
f = open("baratok.txt", "r")
xs = f.readlines()
f.close()

xs.sort()

g = open("rendezett.txt", "w")
for v in xs:
    g.write(v)
g.close()
```

```
f = open("szoveg.zip", "rb")  
g = open("masolat.zip", "wb")
```

```
while True:  
    buf = f.read(1024)  
    if len(buf) == 0:  
        break  
    g.write(buf)
```

```
f.close()  
g.close()
```

Bináris állomány feldolgozása

Szöveg beolvasása az internetről

```
import urllib.request

def weboldal(url):
    """ Visszatér a weboldal tartalmával.
        A tartalmat sztringgé alakítja, mielőtt visszatérne.
    """
    csatlakozo = urllib.request.urlopen(url)
    adat = str(csatlakozo.readall())
    csatlakozo.close()
    return adat

szoveg = weboldal("http://www.gnu.org/software/make/manual/make.txt")
print(szoveg)
```

Összefoglalás

- Python jellemzése
 - Adatok, utasítások
 - Lista és string adatszerkezet
 - Függvények
 - Egységteszt
-
- Feladat: Készítsd el a programozási tételeket Python nyelven- használhatsz lista adatszerekezetet.