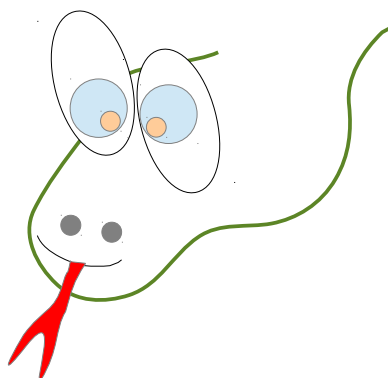


Python a gépben

*Programozási tankönyv teljesen kezdőknek,
ajánlott a középiskola első osztályától*

Koós Antal



Szöveg, rajzok, programok © Koós Antal, 2016
Hungarian edition © Typotex, Budapest, 2016
Engedély nélkül semmilyen formában nem másolható!

ISBN 978-963-279-884-4

Kedves Olvasó!
Köszönjük, hogy kínálatunkból választott olvasnivalót!
Újabb kiadványainkról, akcióinkról
a www.typotex.hu és a facebook.com/typotexkiado
oldalakon értesülhet.

Kiadja a Typotex Elektronikus Kiadó Kft.
Felelős vezető: Votisky Zsuzsa
Főszerkesztő: Horváth Balázs

A könyv teljes terjedelmű PDF változata csatolmányként tartalmaz rajzokat és a tárgyalt programok forráskódját. A csatolmányt nem minden megjelenítő program tudja kifejtetni, de az ingyenes és nyílt forráskódú Evince igen. A könyv írásakor ez a megjelenítő Windowshoz és Linuxhoz a következő címen volt elérhető: wiki.gnome.org/Apps/Evince/Downloads
(A Linux főbb terjesztései az Evince-t eleve tartalmazzák.)

Tartalomjegyzék

1. Bevezetés.....	6	19. Modulok készítése.....	156
2. Miért a Python 3-at tanuljuk meg?.....	7	20. Szótárak.....	160
3. Előkészületek.....	9	21. Készletek.....	168
4. Egy Python program futtatása.....	11	22. Függvények II.....	174
5. Alapozás interaktív módban.....	14	23. Rekurzió	183
6. Függvények I.....	20	24. Bejárók és generátorok	194
7. Változók és objektumok.....	26	25. Osztályok készítése.....	197
8. A modulok használata.....	32	26. Néhány tanács.....	215
9. SOS.....	35	27. Függelék.....	216
10. Összehasonlítások.....	36	F1. Terminológia.....	216
11. Elágazások és ciklusok.....	44	F2. Az objektumok rajzos modellje.....	217
12. A változók élettere.....	71	F3. Folyamatábrán a hazafelé.py.....	218
13. Számok.....	74	F4. Konfigurációs fájlok kezelése.....	219
14. Számrendszerek és kódok.....	87	F5. Sierpinski-háromszög teknőccel.....	222
15. Karakterláncok.....	92	F6. Utasítások, kifejezések és az eval().....	223
16. Listák.....	113	F7. Mátrix.....	225
17. Sokaságok.....	129	F8. Bitszintű műveletek.....	226
18. Fájlok és kódok.....	132	F9. A felhasznált programok.....	227

Néhány fontosabb fogalom fellelhetősége:

alapszintű és teljes körű másolás	16/114, 16/116, 17/129, 17/129, 20/162, 20/162, 22/177, 25/213						
ábécébe rendezés	15/106	ASCII				14/88, 18/132	
azonosság	5/19, 5/19, 13/82	bit, bájt	14/88, 14/89	behúzás	5/15, 6/23		
bájtszekvencia, bájtömb	18/138, 18/139	bool				5/19	
dokumentációs karakterlánc	15/110	jelentést módosító szekvencia				15/92, 18/133	
enumerate	11/56	exit	11/49	for	11/56, 18/143, 20/162, 21/173		
feltételes értékadás: if-else	10/42	format	15/101	függvényhívás és *			22/182
if	11/44, 11/46	indexelés	11/57, 16/113, 17/129, 20/162, 21/169			készletépítő	21/169
listát használó karakterlánc-metódusok	16/121	iteráció	23/183	komment #	6/24	lambda	22/179
konverzió (típusé)	7/28	kötési erősség (operátoré)			5/16, 9/35, 10/40, 10/43		
lebegőpontos szám	13/74	listaépítő	16/113, 16/122		metódus	7/27	
logikai kifejezés	10/38	munkakönyvtár	18/133	névtér	12/71, 22/174, 25/198		
nonlocal	22/178	operátor	10/36, 13/74, 21/169		pass	25/197	
pontoság	13/82	random	13/83				
range	11/58, 13/78	referencia	7/30		sorozat	13/78	
rendkívüli esemény	5/15, 11/60		sorba rendezés		16/120, 17/129, 21/172		
statikus metódus	15/106, 20/167, 25/202		számrendszer		14/88		
statikus változó	25/199						
szeletelés	15/94, 16/113, 17/129, 21/169			szótárépítő		20/161	
takarítás (cleanup)	11/50, 11/62		tizedes tört		6/22, 13/82		
tizedespont	5/14	try	11/60, 18/143		Unikód	14/89, 15/106, 18/137	
utasítássor megtörése /	10/40	UTF-8	14/89, 18/137		változó	5/16, 7/30	
vezérlő karakterek	14/89	while	11/50, 21/173		with open	18/144	
verem	23/184	zip	17/130, 20/161				

A programok listája

függvény.py (Függvények I.).....	23	listarend.py (Listák).....	122
függvények2.py (Függvények I.).....	24	join_split.py (Listák).....	124
függvények3.py (Függvények I.).....	25	favágás.py (Listák).....	126
modulok.py (A modulok használata).....	33	pparam.py (Listák).....	127
modulok2.py (A modulok használata).....	34	fizikusok.py (Listák).....	128
bool.py (Összehasonlítások).....	41	kptükrözés.py (Listák).....	129
értékadás.py (Összehasonlítások).....	42	scifi.py (Sokaságok).....	133
Pitagorasz.py (Összehasonlítások).....	43	file.py (Fájlok és kódok).....	145
if_pelda.py (Elágazások és ciklusok).....	46	pakol.py (Fájlok és kódok).....	153
behúzás.py (Elágazások és ciklusok).....	46	séta.py (Fájlok és kódok).....	154
eső.py (Elágazások és ciklusok).....	47	pywalk.py (Fájlok és kódok).....	156
eső2.py (Elágazások és ciklusok).....	48	személycsere.py (Fájlok és kódok).....	157
hazafelé.py (Elágazások és ciklusok).....	49	pypath.py (Modulok készítése).....	158
while1.py (Elágazások és ciklusok).....	52	ábécé.py (Modulok készítése).....	159
while2.py (Elágazások és ciklusok).....	52	szótár.py (Szótárak).....	168
while3.py (Elágazások és ciklusok).....	53	kalendor.py (Szótárak).....	169
lelet.py (Elágazások és ciklusok).....	55	készlet.py (Készletek).....	175
brekeke.py (Elágazások és ciklusok).....	56	fgv1.py (Függvények II.).....	177
conti.py (Elágazások és ciklusok).....	56	fgv2.py (Függvények II.).....	177
for.py (Elágazások és ciklusok).....	58	fgv3.py (Függvények II.).....	177
while_index.py (Elágazások és ciklusok).....	59	argumentum1.py (Függvények II.).....	178
range.py (Elágazások és ciklusok).....	60	argumentum2.py (Függvények II.).....	178
range_index.py (Elágazások és ciklusok).....	60	argumentum3.py (Függvények II.).....	179
try.py (Elágazások és ciklusok).....	63	belsőfgv.py (Függvények II.).....	180
try1.py (Elágazások és ciklusok).....	64	transformers.py (Függvények II.).....	181
try2.py (Elágazások és ciklusok).....	65	rekurzió.py (Rekurzió).....	185
try3.py (Elágazások és ciklusok).....	66	rekurzió_esemény.py (Rekurzió).....	185
robotautó_feladat.py (Elágazások és ciklusok).....	67	sumsum.py (Rekurzió).....	187
kitalálás.py (Elágazások és ciklusok).....	70	sumi.py (Rekurzió).....	187
szakaszok.py (Elágazások és ciklusok).....	71	sumsum_tükör.py (Rekurzió).....	189
névtér.py (A változók élettere).....	73	szárazföld.py (Rekurzió).....	190
névtér1.py (A változók élettere).....	73	radar.py (Rekurzió).....	192
e_sorozat.py (Számok).....	81	Hanoi.py (Rekurzió).....	195
végtelen1.py (Számok).....	82	első_osztály.py (Osztályok készítése).....	201
végtelen2.py (Számok).....	82	első_osztály1.py (Osztályok készítése).....	202
fibonacci.py (Számok).....	86	alga.py (Osztályok készítése).....	204
zavaros.py (Karakterláncok).....	99	Sierpinski.py (Osztályok készítése).....	212
L-system.py (Karakterláncok).....	101	del_obj.py (Osztályok készítése).....	216
kódok.py (Karakterláncok).....	106	konfig1.py (Konfigurációs fájlok kezelése).....	230
egyeb_betűrend.py (Karakterláncok).....	111	konfig2.py (Konfigurációs fájlok kezelése).....	231
kódok2.py (Karakterláncok).....	112	teknőc.py (Sierpinski-háromszög teknőccel).....	233
betűkódok.py (Karakterláncok).....	113		

1. BEVEZETÉS

Az általános és a középiskolában a magyar nyelv és az irodalom oktatása igen magas óraszámokban történik. A többségünk olyan szakmát választ, amihez közvetlenül ezek az ismeretek nem szükségesek, csak kevesünkben lesz író, újságíró, szerkesztő vagy magyar nyelvi tanár stb. Közvetve azonban mindenki hasznát veszi ezen ismereteknek, mert társadalmunk már igen bonyolulttá vált: életünk során rengeteg emberrel szükséges kialakítanunk hosszabb-rövidebb idejű kapcsolatokat, a párpercestől az egész életen át tartóig, s a legváltozatosabb témákban kell információt cserélnünk. A társadalom akkor működik hatékonyan, ha a polgárok helyesen (a tanult magyar nyelv segítségével) és pontosan (a tanult irodalom segítségével) tudják magukat kifejezni akár szóban, akár írásban. Könnyebben célt ér az, aki jól kommunikál, aki szabatosan tudja közölni a gondolatait, problémáit.

A dolgok mai állása szerint rengeteg számítógépes rendszer vesz körül bennünket és úgy tűnik ez egyre inkább így lesz. Gondoljunk csak az okostelefonokra, tabletekre, jegykiadó automatákra, navigációs eszközökre stb.-re. Az apró vagy kevésbé apró „kütyükön” túlmenően, hatalmas informatikai rendszerek működtetésére is szükség van ahhoz, hogy a társadalom a jelenlegi formájában fenn tudjon maradni. Ilyenek például a villamos-energiarendszer irányítása, a vasúti- és légi közlekedés szervezése, az áruk elosztását végző logisztikai rendszerek, a pénzügyi elszámoló rendszerek, a közigazgatási adatnyilvántartás, a tudományos adatbázisok stb. Láthatjuk, hogy az emberek közti kommunikáció mellett egyre többször van szükség az ember és valamely számítógépes rendszer közötti kommunikációra. Ez utóbbihoz szerencsére nem kell programozási ismeret, a rendszerek kezelése általában valamilyen képernyős-grafikus felületen történik, ugyanakkor a szóban történő vezérlés még nem elég megbízható.

Az elmondottakból kitűnik, hogy a számítógépes rendszerek teljesen átszövik az életünket, és így talán érdemes lehet az ezek alapját képező programok készítésével valamilyen szinten megismerkedni. Amint az irodalom tanításánál sem az a cél, hogy mindenki író legyen, a programkészítés oktatásánál sem lehet az a követelmény, hogy mindenkiből hivatásos programozó váljék. Az okos „kütyük” gombjait, ikonjait nyomkodva azonban néha nem árt tudni vagy legalább sejteni, hogy mi is történik a gépben. Ezen túlmenően egy ún. magasabb szintű programozási nyelv megtanulása sokat segíthet az iskolai-, később pedig a munkahelyi feladataink elvégzésében. Azt sem tartom kizártnak, hogy a nem távoli jövőben, amellett hogy grafikus felületen vagy szóban irányítjuk a környezetünkben lévő intelligens eszközöket és robotokat, az ügyesebbek közülünk egy ilyesféle programozási nyelven fognak tudni pontosabb, kifinomultabb működtetést lehetővé tevő parancsokat kiadni.

E könyvben a *Python 3* magas szintű programozási nyelvvel ismerkedhet meg a kedves olvasó, akiről feltételezzük, hogy az általános iskolai tanulmányokat már befejezte. Nem szükséges semmilyen programozási előismeret a könyv olvasáshoz, de az asztali számítógép alapszintű kezelésének tudása elvárás. Matematikából sem támasztunk magas követelményt az olvasóval szemben, bár néhány fogalmat használni fogunk a középiskola első évfolyamán tanultakból. A könyv fejezetei egymásra épülnek, s ezért azok sorban történő olvasása az ajánlott, de ha valamelyik elsőre túl nehezen érthetőnek bizonyulna, akkor inkább csak fussuk át és egy későbbi időpontban térjünk vissza hozzá, mintsem hogy abbahagyjuk a könyv olvasását. Vannak fejezetek, amelyeket egy-két óra alatt fel lehet dolgozni, de vannak olyanok is, amelyekhez még két-három hét is kevés lehet. Nem kell sietni az olvasással, „aki lassan jár, tovább ér”. Tudjuk, hogy egy beszélt idegen nyelvet különböző szinteken lehet elsajátítani, de még az alapfokú nyelvismeret is hasznosnak bizonyul, ha külföldre utazunk. A programozásnál is ez a helyzet, egy alapszintű programozási tudás is sikerre vezethet, és egyébként is „a gyakorlat teszi a mestert”.