



Takács Edit
edit.takacs@gmail.com
GitHub:
<https://github.com/gtakacse/>
CS class scripts



AZ EMBERI
ERŐFORRÁSOK
MINISZTERIUMA ÚJ
NEMZETI KIVÁLÓSÁG
PROGRAMJÁNAK
TÁMOGATÁSÁVAL
KÉSZÜLT



Programozás nyelvészeknek (intro)

- Kinek és miért?
- Hogyan működik a számítógép?
- Mit jelent az algoritmikus goldolkodás?
- Scratch



Miért érdemes
programozni?



A számítógép

- gyors
- nagy a memóriája
- jó matekból
- pontos
- nem fárad el
- bírja a monotóniát
- igények szerint alakítható



Mire használható a programozási tudás a kutatásban?

- adatgyűjtésre, adatkinyerésre
- adatok statisztikai elemzésére
- adatok rendezésére
- adatok intelligens keresésére és szűrésére
- vizualizációra
- kísérletek megírására
- kísérletek online terjesztésére



És azon kívül

- a programozási tudásnak komoly értéke van az egyetemen kívül is
 - Data Science
 - Data Mining
 - Big Data
 - Machine Learning
 - Artificial Intelligence
- jó móka

+



Mit nyújt a kurzus?



Mi a kurzus célja?

- elűzze a számítógéphez kötődő esetleges félelmeket
- programozási alapokat adjon
- megmutassa, milyen problémákat képes a számítógép hatékonyan kezelni
- felismerjétek, hogy a saját területeteken milyen részfeladatokat tudnátok automatizálni és a számítógéppel elvégzettetni



Mit csinál a számítógép?

1. számításokat végez
2. az eredményt eltárolja későbbi használatra

És ez valóban elég?

- 1 millió egyszerű számítás másodpercenként
- 1 bit = 1 gr
- 4 millió tonna tárolókapacitás (egy átlagos gében)



0010111010100 000000101000100100100100010
0100 0100010111000010 101111100011
010001000001000100010101110101100010
110100110011011110010 0011010110
1111011100010111110110110110110101100
0 001001101010011011001110000011011110
100011110010001100110001101101010000100
0100011100100110011111110111011101111
0011110100110101011111010011011101110
0110101100010101111111010 100000000
11101010110001100011000001001001001010
101100010100000110010001100100011001100
001001111111011011011000000001001001010
1110000010000101010111110000000001001010
000110110001111000111000010011110000011010
1110110001011111110111111111110011111111



Mit ért a számítógép?

- 1 bit >> 0/1

1	0	1	0	0	1	0	0
128	64	32	16	8	4	2	1

$$1*128 + 0*64 + 1*32 + 0*16 + 0*8 + 1*4 + 0*2 + 0*1 = \\ \mathbf{164}$$

Programozás

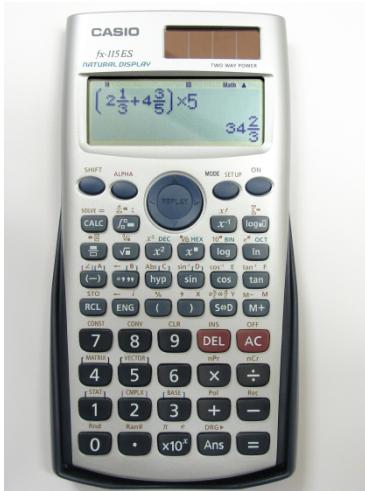
+

=

A számítógépet rávenni arra, hogy azt csinálja, amit szeretnénk.



Fix programú gépek



Programozható gépek



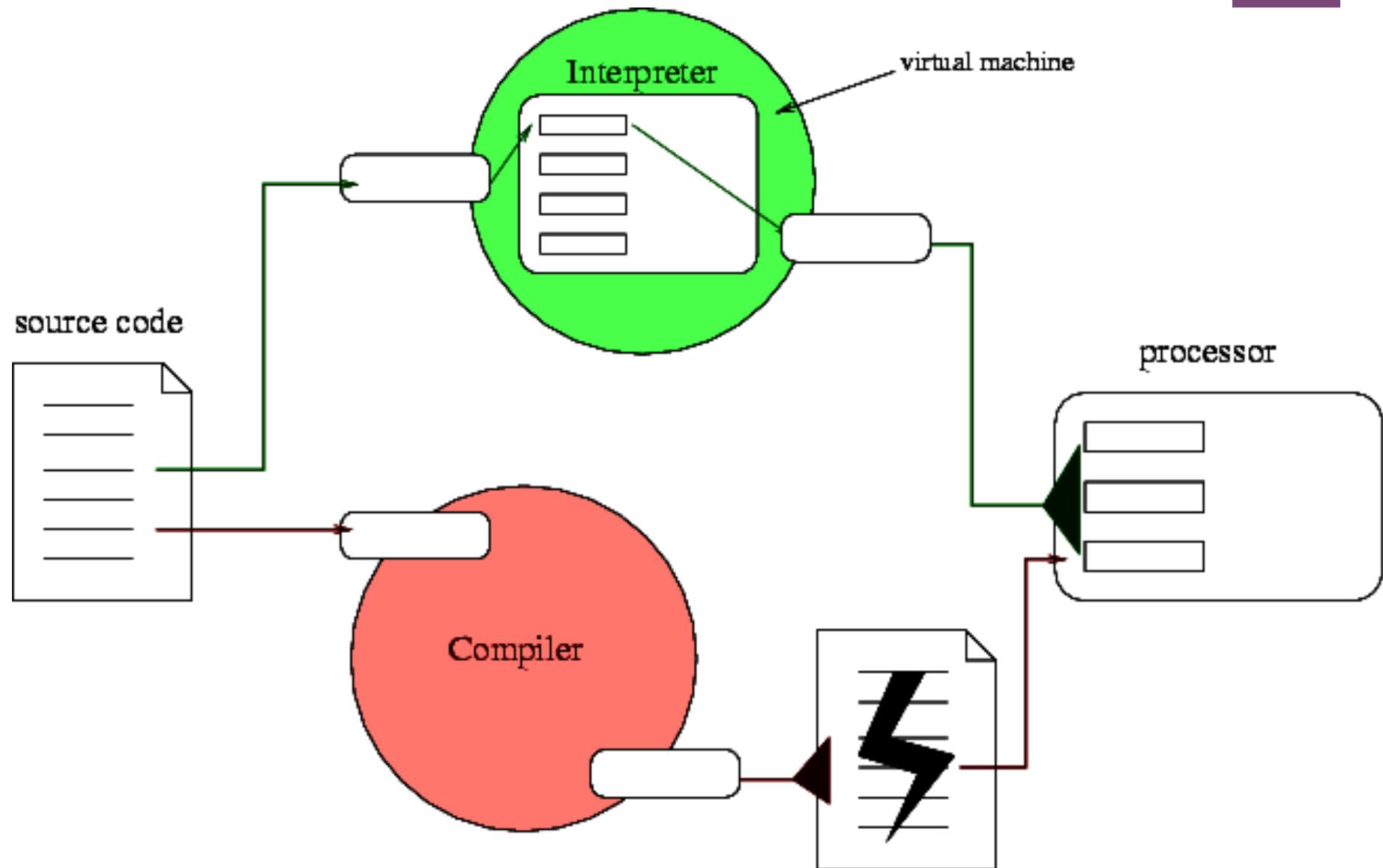
Hogyan kommunikálhatunk a számítógéppel?



Introducing the world's simplest keyboard.

ASCII Code: Character to Binary

0	0011 0000	O	0100 1111	m	0110 1101
1	0011 0001	P	0101 0000	n	0110 1110
2	0011 0010	Q	0101 0001	o	0110 1111
3	0011 0011	R	0101 0010	p	0111 0000
4	0011 0100	S	0101 0011	q	0111 0001
5	0011 0101	T	0101 0100	r	0111 0010
6	0011 0110	U	0101 0101	s	0111 0011
7	0011 0111	V	0101 0110	t	0111 0100
8	0011 1000	W	0101 0111	u	0111 0101
9	0011 1001	X	0101 1000	v	0111 0110
A	0100 0001	Y	0101 1001	w	0111 0111
B	0100 0010	Z	0101 1010	x	0111 1000
C	0100 0011	a	0110 0001	y	0111 1001
D	0100 0100	b	0110 0010	z	0111 1010
E	0100 0101	c	0110 0011	.	0010 1110
F	0100 0110	d	0110 0100	,	0010 0111
G	0100 0111	e	0110 0101	:	0011 1010
H	0100 1000	f	0110 0110	;	0011 1011
I	0100 1001	g	0110 0111	?	0011 1111
J	0100 1010	h	0110 1000	!	0010 0001
K	0100 1011	i	0110 1001	'	0010 1100
L	0100 1100	j	0110 1010	"	0010 0010
M	0100 1101	k	0110 1011	(0010 1000
N	0100 1110	l	0110 1100)	0010 1001
				space	0010 0000





Programnyelvek



Hasonlóságok a természetes emberi nyelvekkel?

- Primitívumok (fonmémák, szavak, jelek vs. számok, betűk, matematikai műveleti jelek)
- Szintaxis (*kézet ‘kezet’)
- Szemantika (*lábott ‘láb múlt ideje’)
- Nyelvjárások és nyelvek
- Lefordíthatóság (ami implementálható egy programnyelvben, az minden más programnyelvben is megvalósítható)
- Hemzsegnek bennük az angol kifejezések (while, if, if else, for, not, in, type, ...)



Különbéségek

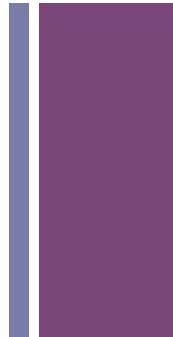
- a programnyelvekben nincs helye a kétértelműségnek
(1 kifejezés kizárolag 1 jelentéssel bírhat)
- kevesebb funkciót tud betölteni, mint a természetes nyelvek (receptek írása a számítógép számára)

+

Mindennapi algoritmusok



Mindennapi algoritmusok

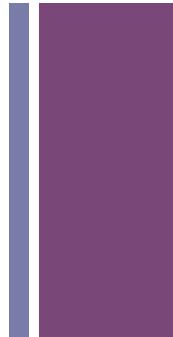


■ DOLGOZAT ÉRTÉKELESE

1. nézd meg a hallgató pontszámát
2. **ha** a pontszám < maximum 60%-a:
3. a hallgató megbukott
4. **egyébként:**
5. a hallgató megfelelt



Mindennapi algoritmusok 2.



■ GÖDÖR BETEMETÉSE

1. **amíg** a gödör nincs tele:
2. vegyél egy lapát földet
3. a lapát tartalmát ürítsd a gödörbe
4. igyál egy hideg üdítőt



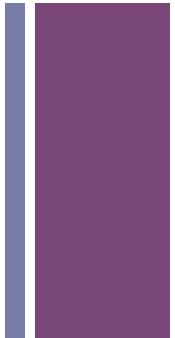
Feladat – Mindennapi algoritmusok

1. Írd le, milyen lépések szükségesek ahhoz, hogy egy szót megtalálj a szótárban.



Bináris keresés

1. vedd kezedbe a szótárt
2. **amíg** nem fogynak el a könyv lapjai:
3. felezd meg a könyvet
4. **ha** a szó a lapon van:
 5. olvasd el a szócikket
 6. **egyébként ha** a keresett szó előbb van az abc-ben mint a lapon szereplő első szó:
 7. hagyd figyelmen kívül a könyv hátsó felét
 8. ismételd meg a felező keresést a könyv első felében
 9. **egyébként:**
 10. hagyd figyelmen kívül a könyv első felét
 11. ismételd meg a felező keresést a könyv hátsó felében
 12. a szó nem szerepel a szótárban

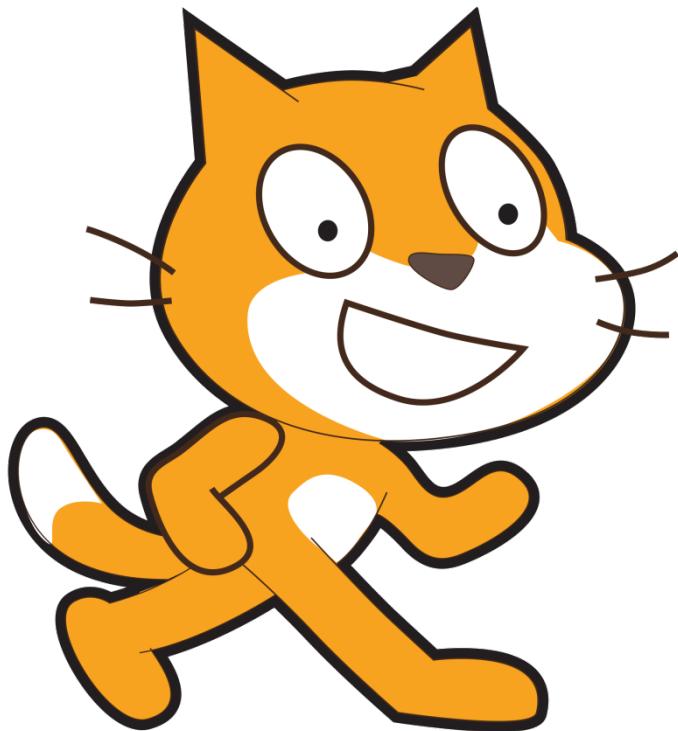
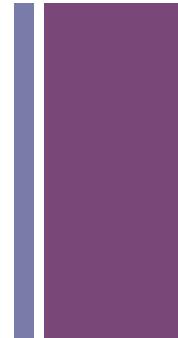


```
1 def say_hello(name):  
2     print('Hello' + name)  
3  
4 say_hello('Emberek')
```

‘Hello Emberek’

+

<https://scratch.mit.edu>





Otthoni feladat

- Írj egy programot Scratch-ben és küldd el mailben hétfő délig!
- Töltsd le (<https://www.continuum.io/downloads>)és telepítsd a következő alkalomra az Anacoda nevű keretrendszert, ami tartalmazza a Spyder programot, amivel Pythonban fogunk programozni!