

Python tanfolyam

3. alkalom

Szöveg kiírása

- Szöveg kiírása STDOUT-ra:
 - `print()` függvény segítségével történik
 - Tetszőleges számú paramétert adhatunk át.
 - Beállíthatjuk az elválasztókaraktert is

print()

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

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

```
print("Hello", "World!", sep=";")
```

Kimenet

```
Hello World!
```

```
Hello World!
```

```
Hello;World!
```

Szöveg beolvasása

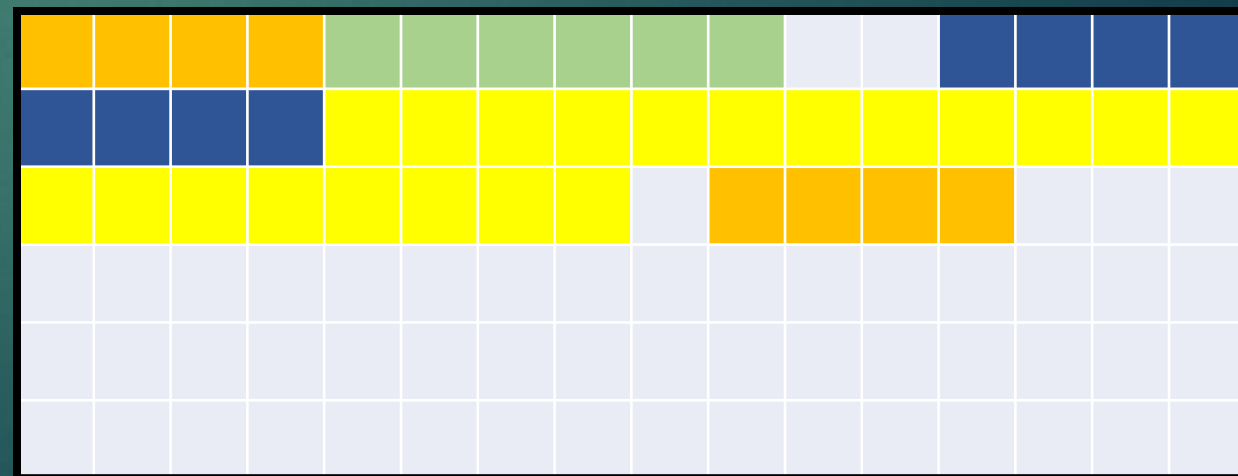
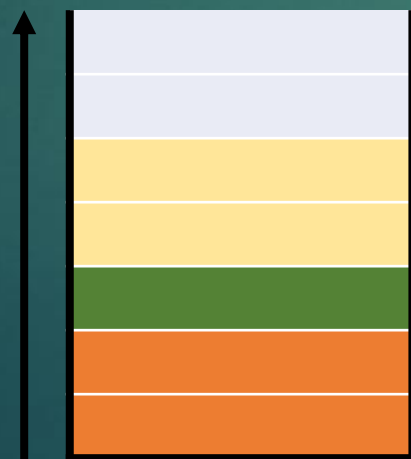
- Szöveg beolvasása a billentyűzetről (STDIN):
 - Az `input()` függvény segítségével történik.
 - A függvény visszatérési értéke a beolvasott szöveg, az “új sor” karakterig.

- Új sor karakter:
 - ASCII kódtábla 10-es (0x0A) és/vagy 13-as (0x0D) karaktere.
 - Operációs rendszer függő, hogy éppen melyik.
 - `\n` és `\r`

Hex	Value	Hex	Value	Hex	Value	Hex	Value	Hex	Value	Hex	Value	Hex	Value	Hex	Value
00	NUL	10	DLE	20	SP	30	0	40	@	50	P	60	`	70	p
01	SOH	11	DC1	21	!	31	1	41	A	51	Q	61	a	71	q
02	STX	12	DC2	22	"	32	2	42	B	52	R	62	b	72	r
03	ETX	13	DC3	23	#	33	3	43	C	53	S	63	c	73	s
04	EOT	14	DC4	24	\$	34	4	44	D	54	T	64	d	74	t
05	ENQ	15	NAK	25	%	35	5	45	E	55	U	65	e	75	u
06	ACK	16	SYN	26	&	36	6	46	F	56	V	66	f	76	v
07	BEL	17	ETB	27	'	37	7	47	G	57	W	67	g	77	w
08	BS	18	CAN	28	(	38	8	48	H	58	X	68	h	78	x
09	HT	19	EM	29	)	39	9	49	I	59	Y	69	i	79	y
0A	LF	1A	SUB	2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z
0B	VT	1B	ESC	2B	+	3B	;	4B	K	5B	[	6B	k	7B	{
0C	FF	1C	FS	2C	,	3C	<	4C	L	5C	\	6C	l	7C	
0D	CR	1D	GS	2D	-	3D	=	4D	M	5D	]	6D	m	7D	}
0E	SO	1E	RS	2E	.	3E	>	4E	N	5E	^	6E	n	7E	~
0F	SI	1F	US	2F	/	3F	?	4F	O	5F	_	6F	o	7F	DEL

Memória

- Minden program használ memóriát a futás közben.
- Ezt a memóriaterületet az operációs rendszer jelöli ki a program számára
- Két féle memóriaterületet használnak a programok
 - Stack – végrehajtási verem
 - Heap

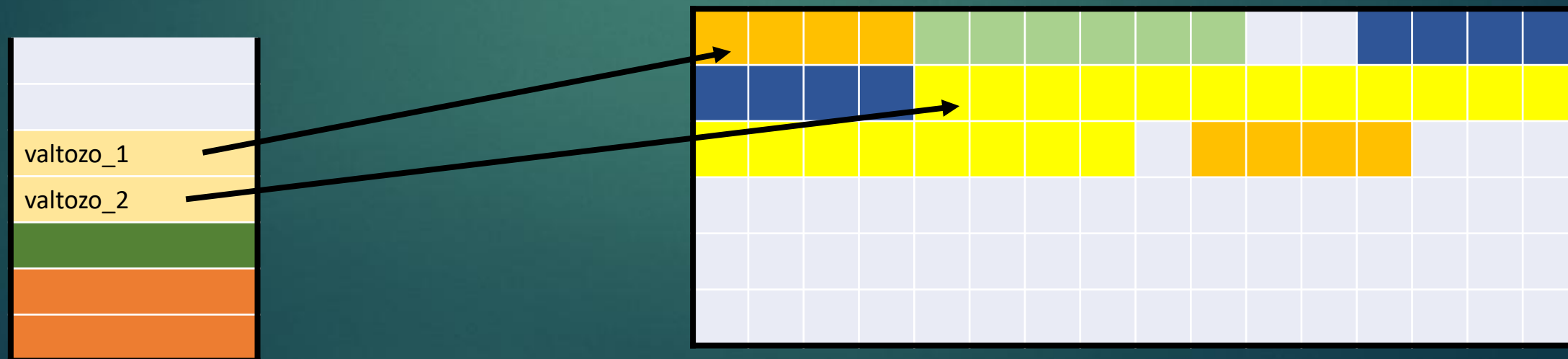


Változók (variables)

- A Python egy magas szintű programozási nyelv, így a memóriához csak nagyon ellenőrzött módon férhetünk hozzá: változókon keresztül.
- A változó “tudja”, hogy hol kezdődik az adat
- A lefoglalt memóriaterület (heap) tudja magáról, hogy
 - mekkora helyet foglal el,
 - milyen műveleteket lehet az ott tárolt adatokon végrehajtani
- Természetesen a változó is a memóriában van tárolva (stack)

Változók és memória Pythonban

- A változó nem más, mint egy referencia egy adott memóriaterületre.
- A lefoglalt memóriaterületnek van típusa
- A változónak nincs típusa, az bármilyen memóriaterületre ráállítható.



Tipusok

- A programozásban a típus meghatározza, hogy egy bizonyos dolognak milyen jellemzői vannak, és milyen műveleteket lehet elvégezni rajta.
- Például:
 - Numerikus típusok:
 - Numerikus (szám) értéke van
 - Matematikai műveleteket lehet rajta (vele) végrehajtani
 - Szöveges típusok:
 - Valamilyen hosszú karaktersorozatot tud tárolni
 - Szöveggel kapcsolatos műveleteket lehet rajta elvégezni, például megfordítás, hossz lekérdezés, összefűzés
 - Összetett típusok:
 - Olyan típus, ami elemekből épül fel, például a lista, vagy valamilyen tetszőleges saját típus.

Egyszerű típusok Pythonban

- *Numerikus típusok*
 - *Egész szám: int*
 - *Lebegőpontos (tört) szám: float*
 - *Komplex szám: complex*
- *Szöveg típus: str*
- *Logikai típus: bool*
- *NoneType (semmi): None*

None és NoneType

- *A legtöbb programozási nyelvben létezik hasonló kiemelt érték ami a semmit jelöli. Pythonban ez a None. Van ahol NULL-nak nevezik.*
- *Ezzel lehet jelölni, hogy változónak nem adtunk még értéket, vagy valamilyen extrémális értéket akarunk neki adni.*
- *A **None** értéken semmilyen műveletet nem lehet végrehajtani.*

Nem-nulla érték



0



None



Deklaráció és Értékadás

- Egy változó deklarálásakor meghatározzuk a változó típusát és nevét, de nem kap értéket.
 - Pythonban ilyen **NINCS**.
 - C, C++, Java: `int x;`
- Amikor egy változóhoz értéket rendelünk, azaz mutatott memóriaterületet rendelünk, azt értékadásnak nevezzük.
 - Az = operator segítségével lehet elvégezni a műveletet.
 - Az értékadás bal oldalán egy változó van, a jobboldalán pedig valamilyen kifejezés.
 - `valtozo_1 = 6`
 - `valtozo_2 = valtozo_1 + 2`

Operátorok

- Az operandus az informatikában nem más, mint egy kifejezés tagja, amelyen valamiféle módosítást hajtunk végre, vagy felhasználjuk egy másik operandus értékének módosításához.
 - Pl.: $x = 5 + 8$ kifejezésben az 5 és a 8 operandus, míg a + jel maga az operátor. A visszatérési értéke (matematikában az eredménye) 13, vagyis x értéke 13 lesz.
- A legtöbb programozási nyelvben, így a Pythonban is használhatóak a matematikai operátorok. (+, -, *, /, =, **)
- Arra is van lehetőség, hogy ezeknek az operátoroknak az operandus típusától függően más viselkedést határozzunk meg.
 - Pl.: `y = "alma" + "korte"`