

Python tanfolyam

Python nyelv történelme

- Guido van Rossum holland programozó 1989 végén kezdi el fejleszteni
- 1991-ben hozza nyilvánosságra
- Akkoriban a C, C++, BASIC, Ada, Pascal, COBOL nyelvek voltak ipari használatban
- Előtérbe kerül az olvashatóság és programozói munka megkönnyítése a program hatékonyságával szemben

A nyelv jellemzői

- A Python programozási nyelv egy általános célú, magas szintű programozási nyelv.
- Támogatja az imperatív, objektum orientált és funkcionális programozási paradigmákat.
- Dinamikus típusokat és automatikus memóriakezelést használ *(később majd tárgyaljuk)*

Betartandó kódolási konvenciók

- Minden programozási nyelvnek megvannak az iparban elterjedt konvenciói, amit a nyelvben jártas fejlesztők igyekeznek betartani.
- Ezek a konvenciók nyelvenként nagyon eltérőek lehetnek, még akkor is, ha a nyelv elemei hasonlóak

JAVA

```
int foo(int x) {  
    if ( x > 2) {  
        return 0;  
    } else {  
        return 1;  
    }  
}
```

C / C++

```
int foo(int x)  
{  
    if ( x > 2)  
    {  
        return 0;  
    }  
    else  
    {  
        return 1;  
    }  
}
```

PYTHON

```
def foo(x):  
    if x > 2:  
        return 0  
    else:  
        return 1
```

Szintaxis

- Egy (programozási) nyelv szabályait szintaxisnak nevezzük.
- A szintaxis határozza meg, hogy az egyes utasítások hogy néznek ki, mik azok a *mondatok*, amiket a nyelv elfogad szabályosnak.
- Az érvénytelen utasításokat szintaktikai hibaként jelzi a fordítóprogram vagy az interpreter
- Összefoglalva: Az a cél, hogy a forráskód az adott programozási nyelven legyen nyelvtanilag helyes.
- A fordítóprogram / interpreter ezt ellenőrzi először. Ha hibás, akkor fordítási hibát kapunk.

Szemantika

- Egy program szemantikusan helyes, ha azt és úgy csinálja, ahogyan a követelmények meghatározták.
- Míg a szintaktikai ellenőrzés strukturálisan vizsgálja a kódot, szemantikai ellenőrzés sokkal bonyolultabb
- A szemantikát csak teszteléssel, esetleg matematikai bizonyításokkal lehet belátni
- Szemantikai hiba, ha a program nem jól működik, “bugos”

Python nyelv szintaxisa

- Egyszerűsített jelöléstan
- Különbözik az ún. C típusú (C, C++, C#, Java, JavaScript) nyelvektől
 - Nincs { } a blokkok körül
 - Nincs ; az utasítások végén
 - A feltételeken nem kell () közé tenni
 - A változó létrehozásának nincs speciális kulcsszava (nincs let vagy var)
- A kódblokkokat a behúzás (indentálás) határozza meg
 - Ezt lehet szeretni vagy nem szeretni, de mindenképpen erős különbség a többi programozási nyelvhez képest.

Megjegyzés a kódban

- A kódolás közben írhatunk olyan szöveget, ami a programozóknak szól. Ezeket a sorokat, sor-részeket a fordítóprogram vagy interpreter figyelmen kívül hagyja.
- Ezeket megjegyzésnek, angoluk comment-nek nevezzük.
- Különbözik a C típusú nyelvekben megszokott `//` illetve `/* ... */` formától

Inline comment

```
#Ez egy comment
#Ez egy másik
def foo(x):
    if x > 2: #ez is comment
        return 0
    else:
        return 1
```

Block comment

```
'''
Persze van lehetőség
Hosszabb comment
Írására is
'''
def foo(x):
    if x > 2:
        return 0
    else:
        return 1
```


Szöveg kiírása / Beolvasása

Hova? / Honnan?

- Standard adatfolyamok
 - Standard output (STDOUT): Konzol, más néven a képernyő. Ha kiír valamit egy program, akkor azt a Standard Outputra írja ki.
 - Standard input (STDIN): Billentyűzet, vagy billentyűzetet imitáló eszköz (pl vonalkód olvasó) A billentyűzetről érkező bemenetet erről az adatfolyamról érjük el.
 - Standard error (STDERR): A programunk hibaüzeneteit írhatjuk ide is. Így a rendes kimenet (STDOUT) elkülöníthető a hibáktól. A Linux rendszerek ezt kitűnően kezelik.

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