

Előadás -2

Bevezetés a Pythonba és a számítógépes programozásba



Instructor : Dr. AALWAHAB DHULFIQAR

Advisor : Dr. Tejfel Mate

Mit fogsz megtanulni:

A számítógép működése

A program végrehajtása

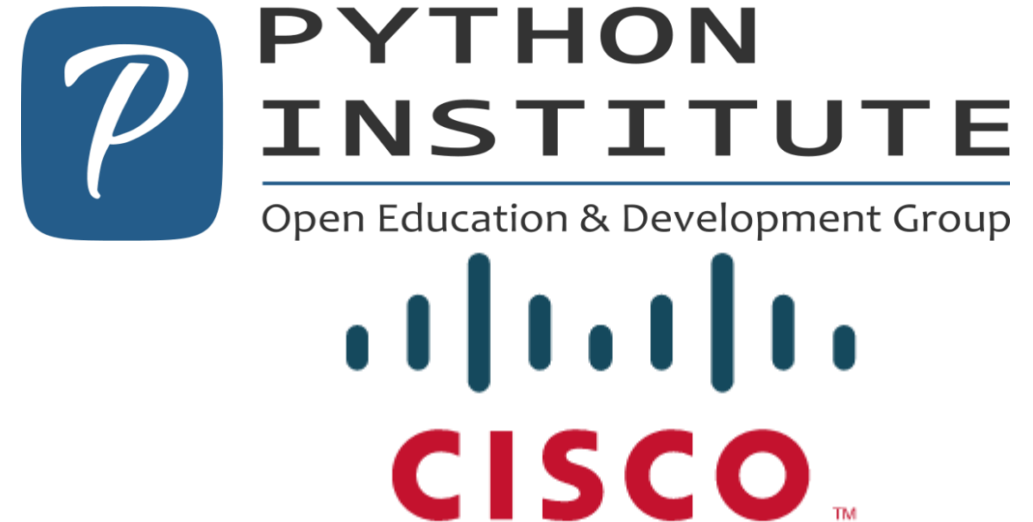
Compilation vs interpretation

Mi a Python

Python verziók

Mik a Python literálok, operátorok és kifejezések;

Mik azok a változók, és milyen szabályok vonatkoznak rájuk;



Hogyan működik a számítógép

Átlagsebesség kiszámítása:

- Fogadjon be egy számot a távolságot reprezentáló értéként;
- Fogadjon be egy számot az utazási idő reprezentálójaként;
- Ossa el az előző értéket az utazási idővel, majd tárolja az eredményt a memóriában;
- Jelenítse meg az eredményt (az átlagsebességet) olvasható formátumban.

Természetes nyelvek vs. programozási nyelvek

A számítógépeknek is van saját nyelvük, az úgynevezett gépi nyelv, amely nagyon rudimentáris.

Az általa felismert parancsok nagyon egyszerűek. Elképzelhetjük, hogy a számítógép olyan parancsokra válaszol, mint "vedd ezt a számot, oszd el egy másikkal, és az eredményt mentsd el".

Az ismert parancsok teljes instrúciók listát nevezzük, amit néha IL-nek rövidítenek.

Jelenleg egyetlen számítógép sem képes új nyelvet létrehozni.

az emberi nyelvek természetes módon fejlődtek ki.

Mitől lesz egy nyelv?



A magas szintű programozási nyelven írt programot forráskódnak nevezzük (ellentétben a számítógépek által végrehajtott gépi kóddal). Hasonlóképpen a forráskódot tartalmazó fájlt forrásfájlnak nevezzük.

```
1 FUX 12:01a 23- 1
A 002000 C2 30 REP #$30
A 002002 18 CLC
A 002003 F8 SED
A 002004 A9 34 12 LDA #$1234
A 002007 69 21 43 ADC #$4321
A 00200A 8F 03 7F 01 STA $017F03
A 00200E D8 CLD
A 00200F E2 30 SEP #$30
A 002011 00 BRK
A 2012

r
PB PC NUMDIZC .A .X .Y SP DP DB
; 00 E012 00110000 0000 0000 0002 CFFF 0000 00
g 2000

BREAK

PB PC NUMDIZC .A .X .Y SP DP DB
; 00 2013 00110000 5555 0000 0002 CFFF 0000 00
m 7f03 7f03
>007F03 55 55 00 00 00 00 00 00 00 00 00 00 00 00 00 00:JU.....
```



Compilation vs. interpretation

A programod

Compilation vs.
interpretation

ML

```
class StudentsDataException(Exception):  
    pass  
  
class BadLine(Exception):  
    def __init__(self, line, message=''):  
        Exception.__init__(self, message)  
        self.line = line  
  
class FileEmpty(Exception):  
    def __init__(self, message=''):  
        Exception.__init__(self, message)
```

ADVANTAGES

DISADVANTAGES

COMPILATION

- a lefordított kód végrehajtása általában gyorsabb;
- csak a programozó felhasználónak kell rendelkeznie a fordítóval - a végfelhasználó anélkül is használhatja a kódot.;
- csak a programozó felhasználónak kell rendelkeznie a fordítóval - a végfelhasználó anélkül is használhatja a kódot.
- maga a fordítás nagyon időigényes folyamat lehet - előfordulhat, hogy a módosítások után nem tudja azonnal futtatni a kódot;
- annyi fordítóra van szükséged, ahány hardverplatformon szeretnéd futtatni a kódot.

INTERPRETATION

- a kódot azonnal futtathatja, amint elkészült - nincsenek további fordítási fázisok;
- a kódot a programozási nyelven tároljuk, nem a gépi nyelven - ez azt jelenti, hogy különböző gépi nyelveket használó számítógépeken futtatható; nem kell külön-külön lefordítani a kódot minden egyes architektúrára..
- ne várd, hogy az interpretáció nagy sebességre fogja gyorsítani a kódot - a kódot megosztja a számítógép teljesítményét az értelmezővel, így nem lehet igazán gyors;
- mind neked, mind a felhasználónak rendelkeznie kell az interpreterrel a kód futtatásához.



Megjegyzés: az interpretációs módon történő felhasználásra tervezett nyelveket gyakran nevezik **szkriptnyelveknek**, míg a velük kódolt forrásprogramokat **szkripteknek**.

Mi is az a Python?

A Pythont az 1956-ban a hollandiai Haarlemben született Guido van Rossum alapította.

In December 1989, I was looking for a "hobby" programming project that would keep me occupied during the week around Christmas. My office (...) would be closed, but I had a home computer, and not much else on my hands. I decided to write an interpreter for the new scripting language I had been thinking about lately: a descendant of ABC that would appeal to Unix/C hackers. I chose Python as a working title for the project, being in a slightly irreverent mood (and a big fan of Monty Python's Flying Circus).

Guido van Rossum



Python célok

- **egy egyszerű és intuitív nyelv**, amely ugyanolyan erős, mint a nagy versenytársaké;
- **nyílt forráskódú**, így bárki hozzájárulhat a fejlesztéséhez;
- olyan kódot, amely **érthető**, mint az egyszerű angol;
- **mindennapi feladatokra alkalmas**, rövid fejlesztési időt lehetővé téve.

Mitől különleges a Python?



A Pythonnak két közvetlen versenytársa van

Perl - egy szkriptnyelv, amelynek szerzője eredetileg Larry Wall volt;

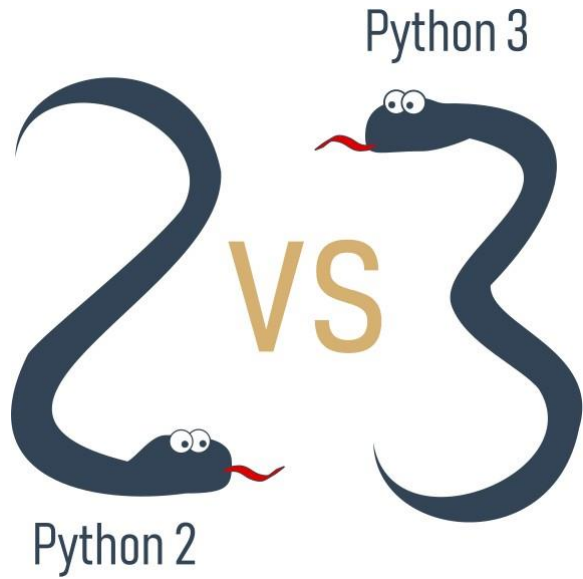
Ruby - egy szkriptnyelv, amelynek szerzője eredetileg Yukihiro Matsumoto volt.

Hol láthatjuk a Pythont akcióban?

Internetes szolgáltatások, például keresőmotorok, felhőalapú tárhelyek és eszközök, közösségi média stb.

Számos fejlesztőeszköz Pythonban van implementálva. Egyre több mindennapi használatú alkalmazást írnak Pythonban. Sok tudós elhagyta a drága, szabadalmaztatott eszközöket, és áttért a Pythonra. Rengeteg informatikai projekt tesztelője kezdte el használni a Pythont megismételhető tesztelési eljárások elvégzésére.

Python verziók



a PSF (Python Software Foundation) köré tömörült emberek tartják fenn.



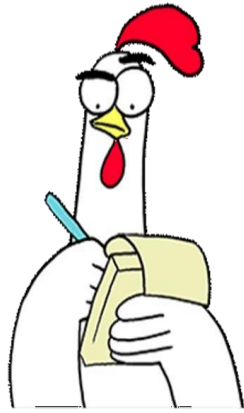
<https://www.python.org/psf-landing/>

Guido van Rossum a "C" programozási nyelvet használta a python legelső verziójának megvalósításához, és ez a döntés még mindig érvényben van. A PSF-től származó összes Python a "C" nyelven van megírva. Ezért a PSF implementációját gyakran CPython néven nevezik.



Checkpoint

- ✓ a számítógépes programozás alapjai, azaz a számítógép működése, a program végrehajtása, a programozási nyelv leírása és felépítése;
- ✓ az összeállítás (compilation) és az interpretáció (interpretation) közötti különbség;
- ✓ az alapvető információk a Pythonról és arról, hogy hogyan helyezkedik el a többi programozási nyelv között, és mi különbözteti meg a különböző változatait;
- ✓ a tananyagokat és a különböző típusú felületeket, amelyeket a kurzus során használni fog.



Literálisok - az adatok önmagukban

A literál olyan adat, amelynek értékeit maga a literál határozza meg. Példa: a 123 egy literál, a c pedig nem.



Integers

English			Python		
11,111,111	11.111.111	11 111 111	11111111	11_111_111	+or - 11_111_111

Egész (integer) számok: oktális és hexadecimális számok

Ha egy egész számot 00 vagy 0o előtag (nulla-o) előz meg, akkor azt oktális értékként kezeljük. 0o123 egy nyolcjegyű szám, amelynek (decimális) értéke 83.

Ha 0x vagy 0X (nulla-x). A 0x123 egy hexadecimális szám, amelynek (decimális) értéke 291.

Floats

Python nyelven 0.4 .4 4.

Ha nagyon nagy vagy nagyon kicsi számokat szeretne használni, használhatja a tudományos jelölést. 8

3000000000 = 3 x 10

Strings

A karakterláncokat akkor használjuk, ha szöveget kell feldolgozni (például mindenféle neveket, címeket, regényeket stb.), nem pedig számokat.

. Példa : "I am a string."

Hogyan nyomtassuk ki ezt ? -> I like "Monty Python"

`print("I like \"Monty Python\"")`

`print('I like "Monty Python"')`

Boolean értékek

English	Python
Yes, this is true;	True
No, this is false	False

Mik azok a változók?

Mi van minden Python változóban?

- egy név;
- egy érték (a tároló tartalma)



Ha nevet akarsz adni egy változónak, akkor néhány szigorú szabályt kell követned:

1. a változó neve nagy- vagy kisbetűkből, számjegyekből és az _ (aláhúzás) karakterből állhat.
2. a változó nevének betűvel kell kezdődnie;
3. az aláhúzás karakter egy betű;
4. a nagy- és kisbetűket különbözőként kezeljük (egy kicsit másképp, mint a való világban - Alice és ALICE ugyanaz a keresztnév, de a Pythonban két különböző változó neve, és következésképpen két különböző változó);
5. a változó neve nem lehet a Python által fenntartott szavak (a kulcsszavak - erről hamarosan többet fogunk mondani) egyike sem.

Helyes és helytelen változónevek

Itt van néhány helyes, de nem mindig kényelmes változónév:

MyVariable, i, t34, Exchange_Rate, counter, days_to_christmas, TheNameIsSoLongThatYouWillMakeMistakesWithIt, _. Ezek a változónevek is helyesek:

Adiós_Señora, sùr_la_mer, Einbahnstraße, переменная.

helytelen nevek:

10t (nem kezdődik betűvel), Exchange Rate (szóközt tartalmaz)

Keywords

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

Creating variables

```
var = 1  
print(var)
```

Megjegyzés: False Nem egyenlő false

Változók használata

```
var = 1
account_balance = 1000.0
client_name = 'John Doe'
print(var, account_balance, client_name)
print(var)
```

```
var = "3.8.5"
print("Python version: " + var)
```

Új érték hozzáadása egy már létező változóhoz

```
var = 1
print(var) # OT = 1
var = var + 1
print(var) # OT = 2
```

```
var = 100
var = 200 + 300
print(var) # OT = 500
```

Shortcut operators

$x = x * 2 \Rightarrow x *= 2$

$\text{Sheep} = \text{Sheep} + 1 \Rightarrow \text{Sheep} += 1$

$i = i + 2 * j \Rightarrow i += 2 * j$

$\text{var} = \text{var} / 2 \Rightarrow \text{var} /= 2$

$\text{rem} = \text{rem} \% 10 \Rightarrow \text{rem} \% = 10$

$\text{em}) \Rightarrow j -= (i + \text{var} + \text{rem})$

$x = x ** 2 \Rightarrow x ** = 2$

Kommentek hagyása a kódban: miért, hogyan és mikor?

Előfordulhat, hogy nem a Pythonhoz, hanem az emberekhez szól néhány szó, általában azért, hogy a kód többi olvasójának elmagyarázza, hogyan működnek a kódban használt trükkök, vagy a változók jelentése, és végül azért, hogy tárolt információ maradjon arról, hogy ki a szerző és mikor íródott a program.

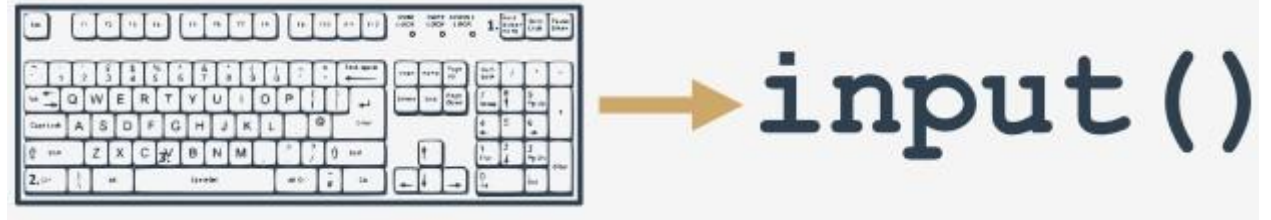
A programba beszúrt megjegyzést, amely futáskor elhagyható, kommentnek nevezzük.

```
# This program evaluates the hypotenuse c.  
# a and b are the lengths of the legs.  
a = 3.0  
b = 4.0  
c = (a ** 2 + b ** 2) ** 0.5 # We use ** instead of square root.  
print("c =", c)
```

The `input()` függvény

Az `input()` függvény képes beolvasni a felhasználó által megadott adatokat, és ugyanezeket az adatokat visszaadni a futó programnak.

```
print("Tell me anything...")  
anything = input()  
print("Hmm...", anything, "... Really?")
```



```
anything = input("Tell me anything...")  
print("Hmm...", anything, "...Really?")
```

The result of the `input()` function

```
anything = input("Enter a number: ")  
something = anything ** 2.0  
print(anything, "to the power of 2 is", something)
```

`TypeError: unsupported operand type(s) for ** or pow(): 'str' and 'float'`

Type casting (Típus öntvény)

A Python két egyszerű függvényt ad meg az adattípus megadására és a probléma megoldására - ezek a következők: `int()` és `float()`.

```
anything = float(input("Enter a number: "))
something = anything ** 2.0
print(anything, "to the power of 2 is", something)
```

```
leg_a = float(input("Input first leg length: "))
leg_b = float(input("Input second leg length: "))
hypo = (leg_a**2 + leg_b**2) ** .5
print("Hypotenuse length is", hypo)
```

String operators - A karakterlánc-operátorok - bevezetés

Konkatenáció : `string + string`

```
fnam = input("May I have your first name, please? ")
lnam = input("May I have your last name, please? ")
print("Thank you.")
print("\nYour name is " + fnam + " " + lnam + ".")
```

Replikáció : `string * number`
`number * string`

```
print("+" + 10 * "-" + "+")
print(("|" + " " * 10 + "|\\n") * 5, end="")
print("+" + 10 * "-" + "+")
```

The slide features decorative geometric shapes in the corners. The top-left corner has overlapping light blue and medium blue squares. The top-right corner has overlapping light purple and medium purple squares. The bottom-right corner has overlapping light blue and medium blue squares.

Találkozunk a labor ülésen