

# Python tanfolyam

## 1. alkalom

# Programozási nyelv

- A programozási nyelv a számítástechnikában használt olyan, ember által is értelmezhető utasítások sorozata, amivel közvetlenül, vagy közvetve közölhetjük a számítógéppel egy adott feladat elvégzésének módját.

# Programozási nyelvek csoportosításai

- Memóriakezelés szempontjából
  - Alacsony szintű (C, C++, Assembly)
  - Magas szintű (Java, C#, Python)
- Használati terület szempontjából
  - Általános célú (C, C#, Java, Python)
  - Adatbázisok területe (SQL, PL/SQL)
  - Matematikai terület (MatLab, R)
- Futtatható program előállítási módja szempontjából
  - Fordítóprogram segítségével (C, C++, Java)
  - Interpreter segítségével (Python, PHP, JavaScript)
- Platformfüggőség szempontjából
  - A programot minden számítógép típusra el kell készíteni
  - A program hordozható, platformfüggetlen

# Programozási paradigmák, módszertanok

- Imperatív programozás
  - Utasítások egymás-utániséga határozza meg a program működését.
- Objektorientált programozás
  - A program egymással kommunikáló objektumok összességéből áll. Az objektumok egységbe zárják az adatokat és meghatározzák a viselkedést.
- Deklaratív programozás
  - A számítás folyamatának leírása helyett a számítás logikáját írjuk le.
- Funkcionális programozás
  - A programozási feladatot egy függvény kiértékelésnek tekintjük. A program függvények hívásából és azok kiértékeléséből áll.

# Programkódból... futó program

## Gépi kód

- A számítógép csak ún. gépi kódot tud értelmezni. Ez több mindentől függ
  - Operációs rendszertől (Windows/Linux/Mac)
  - Processzor architektúrától (32 bit / 64 bit / ARM)
- Gépi kódot nehéz írni, a program nem is lenne könnyen karbantartható.

# Programkódból... futó program

## Architektúra függő kód

- A program logikája nem gépi kódban készül, hanem egy absztrakt szinten, valamilyen programozási nyelven.
- De a programozási nyelvet nem érti a számítógép... Ezért le kell fordítani.
- Fordítás (Compile)
  - A forráskódból egy bonyolult folyamat során egy adott architektúra számára értelmezhető, gépi kódú állomány készül. (exe file)
  - C, C++, Assembly
- Parancsértelmezés (Interpret)
  - A forráskódot soronként értelmezi az interpreter és hajtja végre az utasításokat
  - Basic, PHP

# Programkódból... futó program

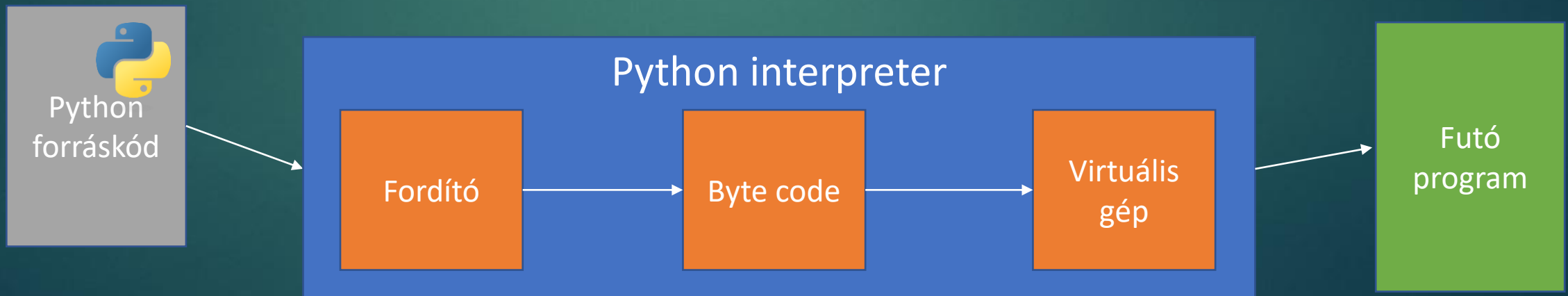
## Architektúra független kód

- Minél többféle gépen akarjuk futtatni a programunkat, annál több munka a fordítás, karbantartás
- Vannak olyan programozási nyelvek, amik nem gépi kódra fordíthatóak, hanem egy köztes nyelvre, amit byte kódnak hívunk.
- Ezt a köztes nyelvet pedig a programozási nyelv úgynevezett virtuális gépe továbbfordítja gépi kódra.
- Ilyen programozási nyelvek a Java, C#, Python

# Programkódból... futó program

## Python

- A Python egy
  - magas szintű,
  - platformfüggetlen,
  - általános célú,
  - interpreteres
  - multiparadigma programozási nyelv.





# Programozási környezet

## Milyen gépünk van?

- Ahhoz, hogy szoftvert fejlesszünk, be kell rendezkedni hozzá a számítógépen.
- Néhány fontos információ kell a számítógépünkről:
  - Milyen operációs rendszert használunk?
  - Milyen processzor van a gépünkben?
- Például
  - Windows 10, 64 bites
  - Windows 10, 32 bites
  - Ubuntu Linux 22.04, 64 bites
- Egy 64 bites processzoron menni fog egy 32 bites operációs rendszer, csak nem fogjuk kihasználni a gép teljesítményét.

# Programozási környezet

## Mi kell a programozáshoz?

- Be kell szerezni (letölteni) az adott programozási nyelv eszköztárát:
  - Fordító, Interpreter, stb
  - Általában SDK-nak rövidítik (Software Development Kit)
- Ha nagyon puritán módon szeretnénk fejleszteni, ennyi akár elég is lehet, hiszen egy Jegyzettömbben is lehet forráskódot írni
- De a kényelem fontos, a jó programozó meg lusta
  - Legyen minden szükséges eszköz egy helyen
- Szerezzünk be egy, az adott programozási nyelvhez ajánlott integrált fejlesztőkörnyezetet.
  - IDE (Integrated Development Environment)

# Szoftverfejlesztés lépései

- A szoftverfejlesztés nem egyenlő a programozással.
- Minden szoftvernek van egy életciklusa, melyen végig kell azt vezetni.
- Ezek a következők:
  - Követelményelemzés
  - Tervezés
  - Megvalósítás
  - Tesztelés
  - Szállítás, üzembehelyezés
  - Karbantartás

# Szakzsargon - Build

- Build, vagy buildelés
  - A forráskódunkat, a teljes projektet lefordítjuk.
  - Lefuttatjuk rá az automatiks teszteket
  - Összeépítünk egy működő verziót.
- 
- A build-et naponta akár többször is végrehajthatjuk.
  - Az elkészült szoftvercsomagot is build-nek nevezzük.

# Szakzsargon - Verzió

- Egy elkészült vagy éppen félkész szoftver állapotát verziónak nevezzük.
- A verziót verziószámmal szoktuk jelölni, mely több részből (általában 3 vagy 4) áll:
  - Például 4.6.8.0 vagy 3.1.2
- A legelső szám az ún fő verzió, azaz **major version**
- A második szám az alverzió, azaz **minor version**
- A harmadik szám a hibajavítás száma, azaz **bugfix revision**
- A negyedik szám pedig a build száma, azaz **build number**

# Szakzsargon - Release

- Általában csak azokat a build-eket verziózzuk, amiket publikálni szeretnénk, azaz release-elni.
- A release egy olyan speciális build, amikor nem csak a szoftvert állítjuk elő a már korábban ismertetett folyamat mentén, hanem:
  - Egyedi verziószámot adunk
  - Elkészítjük a szükséges felhasználói és adminisztratív dokumentációt
  - Eleget teszünk esetleges törvényben meghatározott kötelezettségeknek
  - Az elkészült szoftverterméket és a forráskódot is megfelelően archiváljuk

# Szakzsargon - Deploy

- Amennyiben nem “dobozos” terméket fejlesztünk, az adott rendszert valahova telepíteni is kell.
- Ez lehet akár saját szerver, akár valamilyen cloud-ban üzemeltetett erőforrás, vagy a megrendelő saját eszköze.
- A deploy magában foglalja a release-elt szoftver működőképes állapotba hozását egy cél rendszeren.
- Sokszor a leendő felhasználók betanítását is magában foglalja.

# Python telepítése

- A számítógépünknek megfelelő Python interpreter letölthető a <https://www.python.org/downloads/> címről.
- Jelenleg elterjedt verziók:
  - 3.9.x
  - 3.10.x
  - 3.11.x
- Egyszerre több Python verzió is lehet a számítógépünkön



# Fejlesztőkörnyezetek Pythonhoz

- A Python telepítése után parancssorból elindítva a python értelmezőt, azonnal adhatunk utasításokat Python nyelven.
- Idle
  - A Python alapértelmezett grafikus fejlesztőkörnyezete. Települ a Pythonnal.
- PyCharm
  - Professzionális fejlesztőkörnyezet Python alapú fejlesztésekhez
- Jupyter Notebook
  - Főleg adatelemzők használják
- Spyder
  - Az Anaconda nevű Data Science csomag fejlesztőkörnyezete
- Visual Studio Code
  - Python beépülő modulokkal akár Python nyelvű fejlesztésre is alkalmas

# PyCharm telepítése

- A legelterjedtebb Python IDE a JetBrains által fejlesztett PyCharm.
- <https://www.jetbrains.com/pycharm/download/other.html> oldalról letölthető a legfrissebb verzió.