

HÁLÓZATOK PROGRAMOZÁSA TECHNIKUMI OKTATÁSBAN PYTHON PROGRAMOZÁSI NYELVEN

PROGRAMMING NETWORKS IN TECHNICAL EDUCATION WITH PYTHON PROGRAMMING LANGUAGE

Agg Péter András ⁰⁰⁰⁰⁻⁰⁰⁰³⁻⁰¹⁴⁸⁻²⁸⁶⁸ ^{1*}

¹ Informatika Tanszék, GAMF Műszaki és Informatikai Kar, Neumann János Egyetem, Magyarország
<https://doi.org/10.47833/2025.2.CSC.006>

Kulcsszavak:

Programozás
Hálózatok
Hálózatok programozása
Python
SDN

Keywords:

Programing
Networks
Networks programming
Python
SDN (Software Defined Networks)

Cikktörténet:

Beérkezett 2025. október 10.
Átdolgozva 2025. október 31.
Elfogadva 2025. november 5.

Absztrakt

A digitális világban a számítógépek folyamatosan kommunikálnak egymással, beszélhetünk akár internetezésről, online játékokról, okos eszközökről, a háttérben hálózati protokollok és adatcsomagok dolgoznak. A hálózat programozása lehetővé teszi, hogy a felhasználó ne csak használja ezeket a rendszereket, hanem meg is értse és akár irányítsa őket. A technikai oktatás újbóli bevezetése óta jelentősen megnőtt a hálózati és a programozási ismeretekre szánt kötelező óraszám, mely elősegíti, hogy a tanulók mélyebb ismereteket szerezzenek mindkét témakörben. A két terület szorosan kapcsolódik egymáshoz, hisz a hálózatok vezérléséhez szükség van programozási ismeretekre is. A hálózat programozás lehetővé teszi a hálózati eszközök szoftveres vezérlést, a Python nyelv pedig ideális eszköz ehhez. Ez a cikk bemutatja, hogyan kapcsolódik a hálózatprogramozás és a Python programozási nyelv egymáshoz a középiskolai technikai oktatásban és mit jelent a modern hálózatkonfigurálás napjainkban.

Abstract

In the digital world, computers constantly communicate with each other, can talk about internet use, online games, smart devices, network protocols and data packets work in the background. Network programming allows the user to not only use these systems but also understand and even control them. Since the re-introduction of technical education, the number of mandatory hours devoted to network and programming knowledge has increased significantly, which helps students gain deeper knowledge in both topics. The two areas are closely related, because controlling networks also requires programming knowledge. Network programming allows for software control of network devices, and the Python language is an ideal tool for this. This article presents how network programming and the Python programming language are related to each other in high school

* Kapcsolattartó szerző. Email: agg.peter@nje.hu

1. Bevezetés

A hálózatok használata kikerülhetetlen napjainkban, szinte nincs olyan ember, aki valamilyen módon ne használná a különböző módon összekapcsolt eszközök kommunikációjának előnyeit. A használat nem egyenlő azzal, hogy a felhasználó pontosan tisztában van a működés folyamatával. Magyarországon a visszaállított technikai középfokú oktatás, nagy hangsúlyt helyez ezen ismeretek megszerzésére az adott speciális szakirányú képzés során. Az Informatika és távközlés ágazathoz tartozó Informatikai rendszer- és alkalmazásüzemeltető technikus szakmánál a programozási ismeretek megszerzése után, a hálózatok programozása is részét képezi az oktatásnak. A Python programozási nyelv térhódítása jelentős a középfokú oktatásban [1], így nem véletlen, hogy ez a nyelv lett az elsődleges, mind a programozási ismeretek elsajátításánál, mind a hálózatok vezérlésénél. A Python rendkívül alkalmas az alapvető hálózati programozási feladatok elvégzésére, mert a nyelv sajátosságai, tulajdonságai segítik középiskolai szinten a könnyebb logikai értelmezést és alkalmazást az elvárt hálózatvezérlési feladatok megoldására. Fontos, hogy a tanulók, ne csak a hagyományos hálózatok szempontjából lássák a programozási lehetőségeket, hanem kapjanak betekintés más megoldásokba is, például a Szoftver által definiált hálózatok (Software Defined Networks SDN) témakörben. A jelenlegi technikai oktatás kitér minden fenn említett területre.

2. Python – Programozási alapok a technikai oktatásban

A Python [2][3][4][5][6] programozási nyelvet Guido van Rossum fejlesztette ki 1989-ben. Azt a célt tűzte ki magának, hogy egy olyan nyelvet fejlesszen ki, amely a kódolvasás, a kódértés szempontjából nagyon könnyű, ezért jól tagolható, tömör forráskódot biztosít. Ez az oktatási szempontból jelentős szereppel bír [7]. Nem elhanyagolható az a tény a tanulók szempontjából, hogy könnyebben, és egyszerűbben sajátítják el az alapvető programozási ismereteket, Programozási szempontból fontos a futási sebesség, amely a hálózatok vezérlésénél kiemelt szereppel bír. A Python programozási nyelv gyakorlatilag minden platformon elérhető, (Windows, Linux, MacOS). A felhasználható parancssori alkalmazások fejlesztésénél, webes alkalmazásoknál, hálózati programozásnál is olyan megoldásokat kaphat a tanuló, ami a későbbi nagyvállalati hálózati fejlesztésnél hasznos lehet. Jól használható a nyelv az adatbányászathoz, adatelemzésnél is, melyek döntően befolyásolhatják a hálózatok vezérlését.

A Python a szkriptnyelvek közé tartozik, nyílt forráskódú, magas szintű programozási nyelv. Támogatja a szekvenciális, funkcionális és az objektum orientált programozást (Object-Oriented Programming OOP [8]) is. Kiterjedt könyvtárrendszerrel rendelkezik, melyek által könnyedén lehet implementálni előre elkészített lehetséges megoldásokat egy adott feladathoz, gyorsan és hatékonyan teremthető kapcsolat más programozási nyelvekkel. Python támogatja az egyszerű és összetett adattípusokat, dinamikus a típus- és változókezelése. A Python népszerűségét az objektumorientált jellege jelentősen elősegítette. Könnyen készíthetők és jól használhatók az objektumok, alkalmazva az öröklés adta lehetőségeket is. Fájlkezelése egyszerű, hatékony és gyors. Tekintettel a felsorolt jellemzőkre, nem véletlen, hogy a Python programozási nyelv az elmúlt öt évben rendre az első helyen végzett „A legnépszerűbb programozási nyelvek” kategóriában az IEEE SPECTRUM [9] értékelése alapján.

1. Táblázat: IEEE SPECTRUM értékelése a legnépszerűbb programozási nyelvek kategóriában

| Programozási nyelv | 2024 | 2023 | 2022 | 2021 | 2020 |
|--------------------|------|------|------|------|------|
| Python | 1 | 1 | 1 | 1 | 1 |
| C | 9 | 4 | 2 | 3 | 3 |
| C++ | 4 | 3 | 3 | 4 | 4 |
| C# | 7 | 6 | 4 | 6 | - |
| Java | 2 | 2 | 5 | 2 | 2 |

A középiskola technikai osztályában a 9-10. évfolyamon az alapvető programozási ismereteket sajátítják el a tanulók Python programozási nyelven, az elemi algoritmusok megtanulása után. Ez alapvetően jó megoldás, hisz ezen nyelv kifejezetten egyszerű szintaxissal rendelkezik. Azonban veszélyeket is rejt a későbbi programozási feladatok megoldásainál más programozási nyelvekben (pl. C++) [10].

A tanulók, elsősorban a 10. évfolyamon [11], amely után ágazati alapvizsgát kötelesek tenni, megismerik a Pythonban használható egyszerű és összetett adatszerkezetet, használni tudják az alapvető be- és kiviteli módszereket (input, print), szükség esetén típuskonverziót használnak, alkalmazzák a típuskényszerítést. Ismerik és alkalmazzák az elágazásokat, mind az egy- és többirányút, képesek egymásba ágyazott elágazást is készíteni (if, else, elif). Ciklusokat szerveznek a while és a for parancsok használatával, alkalmazzák a range függvényt is. Érteniük és alkalmazniuk kell a logikai operátorokat (and, or, not). A lista fogalmának megismerése után képesek a különböző listametódusok (append, insert) felhasználására különböző speciális feladatok megoldásánál. A következő lépés a függvények megismerése, mellyel lehetőséget kapnak a paraméterátadás alkalmazására (érték és cím szerint), ami jelentős szereppel bír majd a hálózatok vezérlésénél is. Elegendő gyakorlás után, körül belül a tanév kétharmadánál kezdenek el foglalkozni az objektumorientált programozás alapjaival. Képesek egyszerű osztályok (class) létrehozására, használatára, értik a függvények és metódusok közötti különbséget a Python programozási nyelvben. Ezen fogalmak megismerése, hibátlan használata elengedhetetlen a hálózatvezérlési módszerek elsajátításához. Szintén kiemelt szerepet kap a képzésben a fájlkezelés funkciójának ismerete. A kapott, lekért adatok feldolgozása, kiértékelése, nagy hatással lehet a hálózat működésére amennyiben távoli vezérlési beavatkozás történik (akár jóindulatú, akár rosszindulatú) [12][13]. Részletesen foglalkoznak a tanulók a Python programozási nyelv moduljainak és csomagjainak megismerésével és használatával, hisz ezek a 13. évfolyamon tanulandó hálózat programozása tantárgynál elengedhetetlenek lesznek.

3. Hálózatprogramozás technikumban

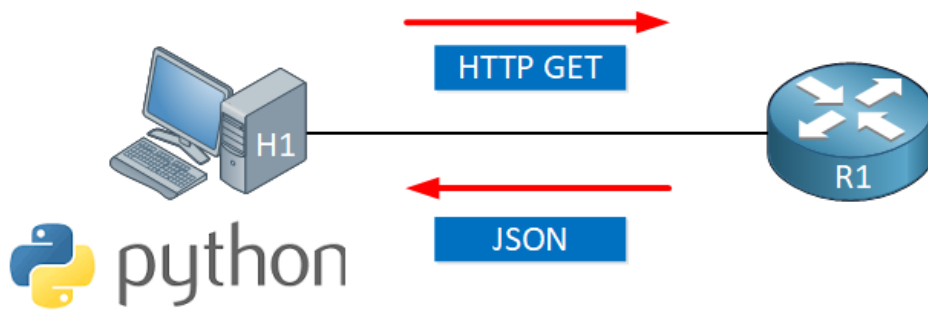
Programozási alapismeretek elsajátítása után viszonylag nagy szünet következik az oktatásban, mire újra programot írnak a tanulók. Az oka annak, hogy csak az utolsó évben folytatják a programozási ismereteket, hogy a hálózat programozása tantárgyhoz minden hálózattal kapcsolatos fogalmat ismerniük kell. Nem elég csak az elmélet, hanem a hagyományos hálózatok konfigurálásának módszerét is tudniuk kell, beszéljünk akár egy forgalomirányító vagy kapcsoló beállításáról.

A hálózatok programozásakor érdemes két csoportról beszélni, a hagyományos és a modern hálózatkezelésről. Hagyományos megközelítés esetén történhet közvetlen parancssorból (CLI) a konfigurálás, vagy távolról is (Telnet/SSH). Hátrány, hogy egyenként kell beállítani az eszközöket, amely időigényes, és emberi mulasztással is járhat. Modern konfigurálás esetében a hálózati eszköz szabványos interfészével történik a kommunikáció egy programon keresztül, amely végrehajtja a kiadott utasítást. Előnyös, mert így megírt program újra felhasználható, átalakítható, akár különböző gyártók eszközeihez is, jelentősen gyorsabb és megbízható lesz a beállítás folyamata. Ennél a módszernél elengedhetetlen egy API (Application Programming Interface) [5] és valamilyen szabványos adatleíró nyelv használata (YAML, XML, JSON).

Az API egy programozási interfészt, programozási felületet és annak részletes dokumentációját jelenti, amellyel egy másik programhoz csatlakozhat. Az API nem egy konkrét programozási nyelv, nem korlátozódik egy konkrét nyelvre. Lehetővé teszi, hogy a programok egymással kommunikáljanak, szükség esetén adatokat kérdezzenek le (1. ábra), módosítsanak, esetleg előre megírt funkciókat használjanak.

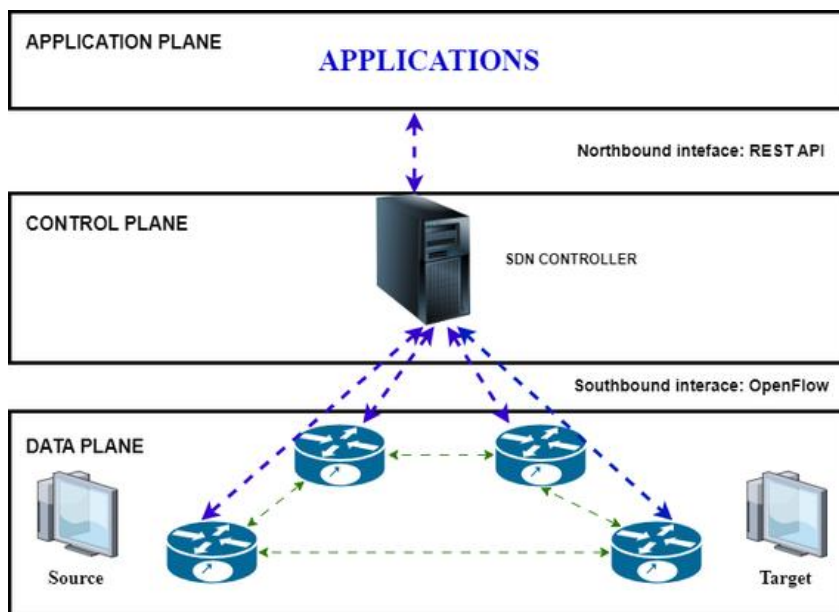
Az adatleíró nyelvek közül a JSON (JavaScript Object Notation) [14] egy jól olvasható és írható adatszerkezet megoldás. Programnyelvtől független, szöveges formátum. Felépítése két féle lehet: név-érték párok halmaza (rekord, struktúra, szótár), vagy értékek rendezett listája (tömb, lista, vektor). A JSON adatszerkezetét a legtöbb programozási nyelv támogatja, felépítése egyszerű, jól használható.

A YANG (Yet Another New Generation) [15] egy adatmodellező nyelv, egy szabvány, amelyet az RFC 6020 ír le. A YANG által használt adatmodellek leírják, hogy mit lehet konfigurálni, monitorozni, és milyen műveletek hajthatók végre egy eszközön. A YANG modellek két csoportba oszthatók: általános (iparági szintű, például IEEE, IETF által kiadott) és gyártóspecifikus adatmodellek (gyártóspecifikus modellek). A YANG az XML adatformátumhoz hasonló hierarchikus fastruktúrát alkalmaz. Szétválasztja a konfigurációs adatokat és az állapotinformációkat.



1. ábra. Adatok lekérése [7]

A modern hálózati infrastruktúra említésénél kikerülhetetlenek a Szoftver által definiált hálózatok (Software Defined Networks - SDN) [16][17][18] megismerése. Az SDN ellentétben a hagyományos hálózatokkal, szétválasztja a vezérlési síkot (control plane) és az adatsíkot (data plane) az eszközökben, így a hálózati eszközök csak továbbítóként működnek, a vezérlés pedig központosításra kerül. Ez a megoldás biztosítja a megfelelő irányíthatóságot a programozható interfészekon keresztül. Az SDN három fő réteget definiál. Ezek az adatsík (Data plane), a vezérlő sík (Control plane), és a menedzsment vagy alkalmazási sík (Management or Application plane) (2. ábra.)



2. ábra. Az SDN architektúra [20]

Az adatsík biztosítja a hálózati csomagok mozgását az egyes hálózati szegmensek vagy végpontok között. Két fő részre bontható a hálózati infrastruktúra (Network Infrastructure) és a déli interfész (Southbound API). A déli interfészek (Southbound API) biztosítják a kommunikációt az adat és a vezérlő sík között. Bár hardverfüggőek, de az nyílt szabványú OpenFlow [19] protokoll használata nagymértékben megkönnyíti hálózati infrastruktúra kialakítását.

A vezérlő sík az északi interfészen (Northbound Interface, Northbound API) keresztül

kommunikál a legfelső réteggel. Lehetővé teszi, hogy hozzáférjen a vezérlőhöz, információlekérés, vagy konfigurálás céljából. Ezek az API-k lehetnek grafikus vagy parancssoros felületűek is.

A menedzsment vagy alkalmazási sík magas szintű programozási nyelveket alkalmaz létrehozva magasabb szintű absztrakciókat annak érdekében, hogy egyszerűsödjenek az elvégzendő feladatok. A probléma-orientáltságra fókuszálva felgyorsul az adatkommunikáció, és biztosított lesz az adatok áramlásának megbízhatósága központi felügyelet mellett. Ebben a rétegben oldható meg a szoftverek modularizálása és a kód újra hasznosítása. Lekérdezhetők a például Python programok segítségével az eszközök adatai, az interfészek állapotának jellemzői, IP címek, forgalomirányítási protokoll fajtája. Megoldható a hozzáférési listák automatikus konfigurálása, és az energiamegtakarítás [20] is.

Modern hálózatok konfigurálásánál szükség van egy API-ra és valamilyen szabványos adateleíró nyelvre. Az SDN vezérlők REST API-t használják (REpresentational State Transfer) [21] a vezérléshez. Itt egy erőforrás információi egy szerver és egy kliens között továbbítódnak, reprezentálódnak. Az információk megjelenítéséhez JSON vagy XML szabványos adatformátum használható. Az állapotátvitel ebben az esetben azt jelenti, hogy minden REST API-val végzett művelet önálló. Minden kérés átküld minden olyan információt, amely az adott művelet befejezéséhez szükséges. Az SDN vezérlője és egy másik alkalmazásközötti információcserére a REST API http üzeneteket használ. HTTP GET az adatok lekérésére, a HTTP POST/PUT az információk frissítésére feltöltésére szolgál. Használható még a HTTP DELETE is törlésre. Az SDN általában a JSON és az XML adatformátumot alkalmazza.

A hálózat automatizálásához szükség van olyan protokollokhoz, melyek biztosítják a programozható interfészt az eszközkonfigurációhoz, a konfigurációs és alapadatok elkülönítését, valamint a hibaelhárítást is. Használható protokollok lehetnek például a NETCONF [22], vagy a RESTCONF [22]. A NETCONF a hálózati eszközök konfigurációját kezeli, távoli eljárás hívásokat alkalmazva hatékony hálózat automatizálást biztosít. SSH felett dolgozik. XML-t használ az adatok lekérésére, a használt elemek a <data> címkébe vannak beágyazva. Használt parancsok: <get-config>: konfiguráció lekérése, <edit-config>: konfiguráció elvégzése. Ezen műveletek távoli eljárás hívásként hajtódnak végre (RPC, Remote Procedure Call). Címkéi lehetnek: <rpc>, <rpc-reply>

A RESTCONF a RESTAPI-hoz hasonlóan működő protokoll. Az adatok szállításához HTTPS protokollt használ. Adatformátumként megfelelő számára a JSON vagy XML. A RESTCONF protokoll egy újabb verzió a NETCONF protokollhoz képest, de nem váltja ki. Az oktatásban az lehet elmondani a tanulóknak, hogy ez inkább egy kicsit könnyebb, másik protokoll a REST API-kat ismerő, és használó rendszergazdák számára.

Hangsúlyozni kell, hogy jelenleg sajnos nem minden eszköz támogatja ezt a modern megközelítést, különös tekintettel a középiskolai technikai oktatásban. Nem minden iskola rendelkezik olyan eszközökkel, amelyekkel az SDN hálózatok vezérlése is bemutatható. Folyamatos fejlesztés van, de a közbeszerzési folyamatok miatt még évekig is eltarthat míg minden középiskolai iskolában megfelelő eszköz fog rendelkezésre állni, a megfelelő REST API kapcsolódás megoldására. Az SSH viszont minden eszközön elérhető, így készíthetők olyan programok, amelyekkel elvégezhetők olyan beállítások, lekérések, mint modern hálózatkonfigurálásnál. Python [23] program segítségével lehet kapcsolódni az adott eszközhöz, lekérdezhetők a beállításai, módosíthatók a konfigurációk [5]. Az SSH hozzáférést a Pythonban a Paramiko könyvtár biztosítja, melyre a Netmiko függvénykönyvtár épül, amely a hálózati eszközök kezelését segíti. A Netmiko biztosítja, hogy különböző gyártók eszközein lehessen az előbbieken felsorolt műveleteket, gyorsan és hatékonyan végrehajtani.

Az alábbi két példában (ospf.py, sw_port.py) látható, hogy megfelelő hálózati ismerettel, és programozási tudással egyszerűen hozzáférhetőek és módosíthatóak a hálózati eszközök beállításai. A feladatokat meghatározott és megtárgyalt specifikáció után oldják meg a tanulók. Az első esetben (OSPF beállítása, ospf.py) egy forgalomirányító OSPF beállításának gyorsítása látható. Használni kell a reguláris könyvtárat (keresés miatt), illetve a Netmiko-ból a ConnectHandler metódust, mely az eszközhöz való kapcsolódást biztosítja. A kapcsolódási információk egy dictionary-be kerülnek, illetve a szükséges beállítási információk is tárolódnak. A jelszavak titkosítása elengedhetetlen, ez a Hálózatok I. és Hálózatok II. tantárgy oktatása során kellően hangsúlyozott. A Python implementációban nem tantárgyi követelmény a titkosítás megoldása. Az

eszközkapcsolódás után a kapcsolódó interface-ek nevei kerülnek meghatározásra, melyek egy listában kerülnek. Ezután a konfigurációs fájl létrehozása következik, melyet a routerhez kell továbbítani.

#ospf.py OSPF beállítása

```

from netmiko import ConnectHandler
import re

#kapcsolódási adatok, titkosítás itt most nem feladat
device = ConnectHandler()
    device_type="cisco_ios",
    ip="192.168.1.1",
    username="user",
    password="cisco",
    secret="cisco123"
)
#adatok bekérése a felhasználótól
oid = input("OSPF-ID: ")
rid = input("Router-ID: ")
area = input("Area: ")
device.enable()

#információ kérése az eszköztől, majd a kapott információ feldolgozása
output = device.send_command("show ip router connected | begin Gateway")
interflist = []
for line in output.splitlines():
    a = re.search(r'^C', line)
    if a:
        s = line.split()
        intface = s[5]
        interflist.append(intface)

#konfigurációs fájl létrehozása, parancsok kiírása fájlba
ofile = open("ospf.config", "w")
ofile.write("router ospf "+oid+"\n")
ofile.write("router-id "+rid+"\n")
for i in interflist:
    ofile.write("interface "+i+"\n")
    ofile.write("ip ospf "+oid+" area "+area+"\n")
ofile.close()
ofile="ospf.config"

#parancs kiadása, elküldése
p=device.send_config_from_file(ofile)

```

A következő példa (Kapcsoló beállítása, sw_port.py) egy tetszőleges kapcsoló nem használt interfészeit lekapcsolja, a használt portokat átrakja egy megadott vlanba, illetve a Gigabites interfészt (ami használt) trónkmódba váltja.

#sw_port.py kapcsoló beállítása

```

from netmiko import ConnectHandler
#kapcsolódási adatok, titkosítás itt most nem feladat
switch = {'host':'10.0.100.10',
          'username':'admin',
          'password':'cisco',

```

```
'device_type':'cisco_ios',
'secret':'cisco'}
#kapcsolódás
connect = ConnectHandler(**switch)

#információ kérése az eszköztől, majd a kapott információ feldolgozása
comm=connect.send_command('show interfaces status')
portdown=[]
portup=[]
portgib=[]

#parancs szétbontása
for l in comm.splitlines():
    if 'notconnect' in l:
        line=l.split()
        portdown.append(line[0])
    if 'connected' in l:
        line=l.split()
        portup.append(line[0])
    if 'Gi' in l:
        line=l.split()
        portgib.append(line[0])

#parancsok kiadása, elküldése
for l in portdown:
    comm=connect.send_config_set(["interface "+l,"sw mode ac","sw access vla
69","shutdown"])
    print(comm)
for l in portup:
    comm=connect.send_config_set(["interface "+l,'sw mode access','sw acc vla 100'])
    print(comm)
for l in portgib:
    comm=connect.send_config_set(["interface "+l,'sw mode trunk'])
    print(comm)

#kapcsolat bontása
connect.disconnect()
```

4. Következtetések

A középiskolai technikai oktatásban bevezetett hálózatok programozás tantárgy egyértelműen mutatja, hogy milyen fontos szerepet tölt be napjainkban a hálózatok használata és a programozási alapok ismerete, és milyen szoros a kapcsolat a két terület között. A programozási nyelvek közül a Python nyelv könnyen tanulható, népszerű nyelv a diákok körében és jól alkalmazható a hálózatok vezérlésében is. Jól látható, hogy a technikai képzésben az oktatók nem csak az alapos programozás tudás elsajátításában játszanak nagy szerepet, hanem a hagyományos hálózatkonfigurációs módszerek mellett a modern hálózatok programozására is nagy hangsúlyt fektetnek. Python nyelvhasználatával képesek gyorsan és hatékonyan beavatkozni szükség esetén a hálózat működésébe. A hálózati eszközök fejlődése megállíthatatlan, így a humán erőforrás folyamatos képzése is elengedhetetlen. A technikai oktatás gyakorlatorientált képzése elősegíti, hogy a megfelelő programozási tudást megszerző tanulók jól képzett hálózati szakemberekké váljanak, és alkalmazkodjanak a folyamatosan változó kihívásokhoz.

Irodalomjegyzék

- [1] Szerémi Éva Krisztina, Python programozási nyelv térhódítása a középfokú oktatásban, 2024. Január Gradus 11(3), DOI: [10.47833/2024.3.CSC.006](https://doi.org/10.47833/2024.3.CSC.006)
- [2] Python alapok: <https://prog.hu/cikkek/100424/python-alapok> Megtekintve: 2025.10.14.
- [3] Gerard Swinnen: Tanuljunk meg programozni Python nyelven: <https://mek.oszk.hu/08400/08435/08435.pdf> Megtekintve: 2025.10.01.
- [4] Pythonarium - Bevezetés a Python programozásba: <https://medium.com/pythonarium/bevezet%C3%A9s-a-python-programoz%C3%A1sba-6cef30b7137e> Megtekintve: 2025.09.20.
- [5] Vadlamani A, Kalicheti R, Chimalakonda S. API Scanner - Towards Automated Detection of Deprecated APIs in Python Libraries. May 2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), Madrid, ES. 2021; 5–8. DOI: <https://doi.org/10.1109/ICSE-Companion52605.2021.00022>
- [6] Loulergue F, Philippe J., Skeletons for the Python Skeleton Library. 2019 20th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), Gold Coast, QLD, Australia. 2019; 392–397. DOI: [10.1109/PDCAT46702.2019.00077](https://doi.org/10.1109/PDCAT46702.2019.00077)
Közösség az informatikaoktatás megújításáért <https://itmp.hu/tanar-vagyok> Megtekintve: 2025.09.20.
- [7] Kalb, Irv, Object-Oriented Python - Master OOP by Building Games and GUIs, Random House LLC US, 2022, ISBN: 1718502060
- [8] IEEE SPECTRUM: <https://spectrum.ieee.org/top-programming-languages-2024> Megtekintve: 2025.10.21.
- [9] Ulmann Kristóf: Programozási nyelvek: C++ 2018
https://people.inf.elte.hu/szelethus/LaTeX/cpp/cpp_book/cpp_book.pdf Megtekintve: 2025.09.10.
- [10] Programterv a 12. Informatika és Távközlés ágazathoz tartozó 506121202 Informatikai Rendszer- és Alkalmazásüzemeltető Technikus Szakmához
https://afg.hu/pdf/ptt_informatika_es_tavkozles_agazat_informatikai_rendszer-es_alk.pdf Megtekintve: 2025.09.30.
- [11] Mritunjay Kr. Ranjan, Krishna Barot, Vaishnavi Khairnar, Vaishnavi Rawal, Anujaa Pimpalgaonkar, Shilpi Saxena, Arif Md. Sattar, Python: Empowering Data Science Applications and Research, Journal of Operating Systems Development & Trends, June 2023, 10(1):27-33. DOI: [10.37591/joosdt.v10i1.576](https://doi.org/10.37591/joosdt.v10i1.576)
- [12] Soti Attila, A Python programozási nyelvről statisztikusoknak, On the Python programming language for statisticians, Statisztikai Szemle 98(4), April 2022, DOI: [10.20311/stat2020.4.hu0324](https://doi.org/10.20311/stat2020.4.hu0324)
- [13] JSON Model: <https://sapui5.hana.ondemand.com/sdk/#/topic/96804e3315ff440aa0a50fd290805116.html> Megtekintve: 2025.08.01.
- [14] YANG model: <https://standards.ieee.org/beyond-standards/ieee-yangsters/> Megtekintve: 2025.10.01.
- [15] Diego Kreutz, Fernando M. V. Ramos, Paulo Verissimo, Software-Defined Networking: A Comprehensive Survey, Proceedings of the IEEE, Vol. 103, No. 1, January 2015 pp. 14-76. DOI: [10.1109/JPROC.2014.2371999](https://doi.org/10.1109/JPROC.2014.2371999)
- [16] S. Mishra, M. A. R. AlShehri, (2017) "Software Defined Networking: Research Issues, Challenges and Opportunities", Indian Journal of Science and Technology, Vol 10, DOI: [10.17485/ijst/2017/v10i29/112447](https://doi.org/10.17485/ijst/2017/v10i29/112447)
- [17] Syed Taha Ali, Vijay Sivaraman, Adam Radford, Sanjay Jha, A Survey of securing Networks Using Software Defined Networking, IEEE Trans. Reliability, 2015, DOI: [10.1109/TR.2015.2421391](https://doi.org/10.1109/TR.2015.2421391)
- [18] A. Lara, A. Kolasani, and B. Ramamurthy, Network innovation using OpenFlow: A survey, IEEE Communications Survey & Tutorials, Vol. 16, NO. 1. First quarter 2014 493-512. DOI: [10.1109/SURV.2013.081313.00105](https://doi.org/10.1109/SURV.2013.081313.00105)
- [19] Péter András Agg, Zsolt Csaba Johanyák: Saving Energy Using the Modified Heuristic Algorithm for Energy Saving (MHAES) in Software-Defined Networks, Sensors 2023, 23(23), 9581; December 2023 DOI: <https://doi.org/10.3390/s23239581>
- [20] David Alfa Sunarna, Gede Putra Kusuma: Designing and Evaluating Representational State Transfer Architecture for School Management Information System, International Journal of Emerging Trends in Engineering Research Volume 8. No. 7, July 2020 <https://doi.org/10.30534/ijeter/2020/124872020>
- [21] Mahesh Ganesh Bhat; Sushmit Bhattacharjee; Cenk Gündoğan; Konstantinos Alexandris; Alexander Gogolev: CORECONF, NETCONF, and RESTCONF: Benchmarking Network Orchestration in Constrained IIoT Devices, IEEE Internet of Things Journal (Volume: 11, Issue: 7, 01 April 2024) DOI: [10.1109/JIOT.2023.3338470](https://doi.org/10.1109/JIOT.2023.3338470)
- [22] Abu Rayhan, David Gross, The Rise of Python: A Survey of Recent Research, License [CC BY 4.0](https://creativecommons.org/licenses/by/4.0/), September 2023, DOI: [10.13140/RG.2.2.27388.92809](https://doi.org/10.13140/RG.2.2.27388.92809)