

Extra – Python a hálózatban (gyorsaknak)

Név: _____ Osztály: _____

Ezt akkor csináld, ha a napi feladatlapppal kész vagy. Nem kötelező. Haladj sorban: minden feladat a korábbiakra épít, és egyre nehezebb. **Mentés:** minden programot külön fájlba, pl. `P1_binaris.py`. A végén mutasd meg a tanárnak, ami működik. **Elakadtál?** Van egy külön lap tippekkel: **Python segítség** (extra-python-segitseg). Csak akkor nyisd meg, ha egyedül már nem megy.

Elmélet: miért kell a hálózatosnak programozni?

Egy kis cégben 3 kapcsoló van: ezeket kézzel is beállítod. Egy nagy cégben, egy iskolában vagy egy szolgáltatónál viszont **több száz** eszköz van. Ha mindegyikre kézzel írod be a hostname-et, a jelszavakat és az IP-címet:

- napokig tart,
- és biztosan elgépelsz valahol (egy rossz IP-cím, egy kimaradt `no shutdown`).

Ezért a hálózatos szakemberek ma már **programmal** dolgoznak. Ezt hívják **hálózatautomatizálásnak** (network automation).

Feladat	Kézzel	Programmal
200 kapcsoló alapkonfigurációja	napok, sok elírás	percek, mind egyforma
„Melyik porton van hiba?”	200 × <code>show ip interface brief</code> , végigolvasva	a program kiírja a hibás portokat
IP-címterv egy új telephelyre	számolgatás papíron	a program kiszámolja

A Python a hálózatosok kedvenc nyelve: egyszerű, és sok kész eszköz van hozzá. Valódi eszközökhöz például a **Netmiko** (SSH-n belép a routerre, és kiadja a parancsokat) vagy az **Ansible** (egyszerre sok eszközt állít be). A Cisco saját vizsgálója is kéri az automatizálás alapjait (CCNA, DevNet).

Ezen a lapon valódi eszköz helyett a Packet Tracerrel dolgozunk: a Python **elkészíti** a konfigurációt, te pedig **bemásolod** a Packet Tracerbe.

0. Indítás

- Nyiss egy parancssort (Start → cmd), és írd be: `python --version`. Ha kiír egy verziót (3.x), van Python a gépen.
 - **Írhatsz a VS Code-ban** (ha fent van) vagy az **IDLE-ben** (Start → IDLE → File → New File; futtatás: F5).
 - **Ha nincs Python a gépen:** használj online szerkesztőt, pl. **online-python.com** vagy **programiz.com/python-programming/online-compiler**. A 4. feladat fájlt írsz, az online szerkesztőben a fájl helyett írsz ki a szöveget a képernyőre (`print`).

Gyorsemlékeztető

Mit?	Hogyan?
kiírás	<code>print("Szia", nev)</code>
beolvasás	<code>ip = input("IP-cím: ")</code> (mindig szöveget ad)
szövegből szám	<code>int("192") → 192</code>
szöveg darabolása	<code>"192.168.10.1".split(".") → ['192', '168', '10', '1']</code>
lista összefűzése	<code>".".join(['a', 'b']) → 'a.b'</code>
szám binárisan, 8 jegyre	<code>format(10, "08b") → '00001010'</code>
ciklus egy listán	<code>for r in reszek:</code> (utána behúzás!)
feltétel	<code>if x == 1: ... elif ...: ... else:</code>

P1 · IP-cím binárisan (nehézség: 1/5, kb. 10 perc)

Mire jó? A router és a kapcsoló mindent binárisan „lát”. Az alhálózatok megértéséhez látnod kell a biteket. Írj programot, amely bekér egy IP-címet, és kiírja binárisan, oktettenként (8 bite kiegészítve).

```
IP-cím: 192.168.10.1
11000000.10101000.00001010.00000001
```

- ☐ Működik a 192.168.10.1-re?
- ☐ Mi a 255.255.255.0 binárisan? _____

P2 · Maszk a prefixből (nehézség: 2/5, kb. 15 perc)

Mire jó? A Packet Tracerben a maszkot így kell beírni: 255.255.255.0. A tervekben viszont gyakran így szerepel: /24. Írj programot, amely átváltja!

A program kérjen be egy prefixet (pl. 26), és írja ki:

```
Prefix (pl. 24): 26
Binárisan: 11111111.11111111.11111111.11000000
Maszk:      255.255.255.192
Gépek száma: 62
```

- ☐ /24 → _____ /28 → _____ /30 → _____ (hány gép? _____)
- ☐ Miért pont /30-at használnak két router összekötésére? _____

P3 · Alhálózat-kalkulátor (nehézség: 2/5, kb. 15 perc)

Mire jó? Megtudod, hogy két gép **ugyanabban a hálózatban** van-e. Ezen múlik, hogy közvetlenül eléri-e egymást, vagy router kell. Emlékszel az L3-ra? Ezért kellett a kapcsolónak ip default-gateway.

Használd a Python beépített **ipaddress** modulját (`import ipaddress`).

Írj programot, amely:

1. bekér egy címet prefixszel (pl. 192.168.10.2/24), és kiírja: hálózat címe, maszk, szórási cím, első és utolsó kiosztható cím, gépek száma;
2. bekér egy **másik** IP-címet, és megmondja, hogy ugyanabban a hálózatban van-e.

```
Cím prefixszel: 192.168.10.2/24
Hálózat:      192.168.10.0
...
Másik IP-cím: 192.168.20.10
Más hálózatban van: router (alapértelmezett átjáró) kell hozzá.
```

- ☐ Próbáld ki az L5 topológiájának címeivel: PC-A és S1? _____ PC-A és PC-B? _____
- ☐ 172.16.5.130/26 → hálózat: _____ szórási cím: _____

P4 · Konfiguráció-generátor (nehézség: 3/5, kb. 20 perc)

Mire jó? Ez az igazi hálózatautomatizálás: a program **megírja helyetted** a kapcsolók konfigurációját, te csak bemásolod.

Az adatok egy listában vannak, eszközönként egy **szótárral** (dictionary):

```
eszkozok = [
    {"nev": "S1", "ip": "192.168.10.2", "atjaro": "192.168.10.1"},
    {"nev": "S2", "ip": "192.168.20.2", "atjaro": "192.168.20.1"},
]
```

Egy szótár elemét így éred el: `e["nev"]` → "S1".

Írj programot, amely **minden eszközhöz** készít egy szövegfájlt (S1.txt, S2.txt) a kapcsoló teljes alapkonzfigurációjával: hostname, no ip domain-lookup, enable secret, konzoljelszó, banner motd, interface vlan 1 az IP-címmel és no shutdown, ip default-gateway, mentés. (Az L2 és az L3 lapon megtalálod a parancsokat.)

Kipróbálás a Packet Tracerben:

- ☐ Építs egy új, üres hálózatot: két 2960-as kapcsoló, gyári állapotban.
- ☐ Nyisd meg az S1.txt-t, jelölj ki mindent, másold (**Ctrl + C**), és illeszd be az S1 **CLI** fölére (jobb klikk → Paste, vagy **Ctrl + V**).
- ☐ show running-config → minden benne van? _____ Működött elsőre? Ha nem, mi volt a hiba? _____
- ☐ Adj hozzá egy harmadik kapcsolót (S3, 192.168.30.2) **egyetlen sorral** a listában. Hány sort kellett változtatnod a programban? _____

P5 · Hibakereső: a show ip interface brief elemzése (nehézség: 4/5, kb. 20 perc)

Mire jó? Egy nagy hálózatban senki nem olvas el 200 kimenetet. A program kikeresi a hibás portokat.

Másold be ezt a kimenetet a programodba (ez egy routeré):

```

kimenet = ""Interface
Protocol
GigabitEthernet0/0    192.168.10.1    YES manual up
GigabitEthernet0/1    192.168.20.1    YES manual administratively down down
GigabitEthernet0/2    unassigned      YES unset  administratively down down
Vlan1                  unassigned      YES unset  up
down""

```

Írj programot, amely soronként (az első, fejléc sort kihagyva) kiírja:

- OK, ha a Status és a Protocol is up;
- HIBA ... no shutdown kell, ha **van IP-címe**, de administratively down;
- NEM HASZNÁLT, ha nincs IP-címe és le van kapcsolva;
- FIGYELEM ... nincs kapcsolat, ha a Status up, de a Protocol down.
- ☐ Mit ír ki a programod a 4 sorra?

- ☐ Próbáld ki a saját L5-ös routered igazi kimenetével: a Packet Tracerben add ki a parancsot, másold ki a kimenetet, és tedd a kimenet helyére.

P6 · Kihívás: IP-címterv generátor (nehézség: 5/5)

Írj programot, amely bekér egy hálózatot (pl. 192.168.30.0/24) és a PC-k számát, majd kiír egy **címzési táblázatot**, olyat, mint az L5 elején:

Eszköz	IP-cím	Masz	Átjáró
R1	192.168.30.1	255.255.255.0	–
S1	192.168.30.2	255.255.255.0	192.168.30.1
PC1	192.168.30.10	255.255.255.0	192.168.30.1
PC2	192.168.30.11	255.255.255.0	192.168.30.1

Szabályok:

- a router kapja az első címet, a kapcsoló a másodikat, a PC-k a .10-től;
- ha a hálózat kicsi (pl. /29), a PC-k közvetlenül a kapcsoló után jönnek;
- ha nem fér el ennyi gép, írja ki: Ennyi gép nem fér el ebben a hálózatban!
- ☐ Próbáld ki: 192.168.30.0/24, 3 PC · 10.0.0.0/29, 3 PC · 10.0.0.0/30, 3 PC.
- ☐ **Szuper kihívás:** kösd össze a P4-gyel: a program a címterv alapján a kapcsoló konfigurációját is készítse el!