# IOT KERETRENDSZEREK ÉS IPARI ALKALMAZÁSAIK

SmartComLab

Gyors prototípus építés II.

Raspbery Pi, Linux, Python

# Raspberry Pi 3B image

The image is for a 4GB microSD card (hopefully class 10 or above).

Preconfigured:

- Arrowhead core framework pre-configured
- Git, Maven, JDK 1.8, MySQL (MariaDB)
- PhpMyAdmin: *http://ip/phpmyadmin*
- Devtools (MCEdit, iptables…)
- Mosquitto MQTT broker
- The Raspberry hosts a Wi-Fi network and tunnels Internet if connected to eth0

Wi-Fi access point details:

- WiFi SSID: Arrowhead-Raspi-IoT#, password: arrowhead
- Raspi is DHCP master on the 192.168.42.10-50 range
- Raspi is the 192.168.42.1

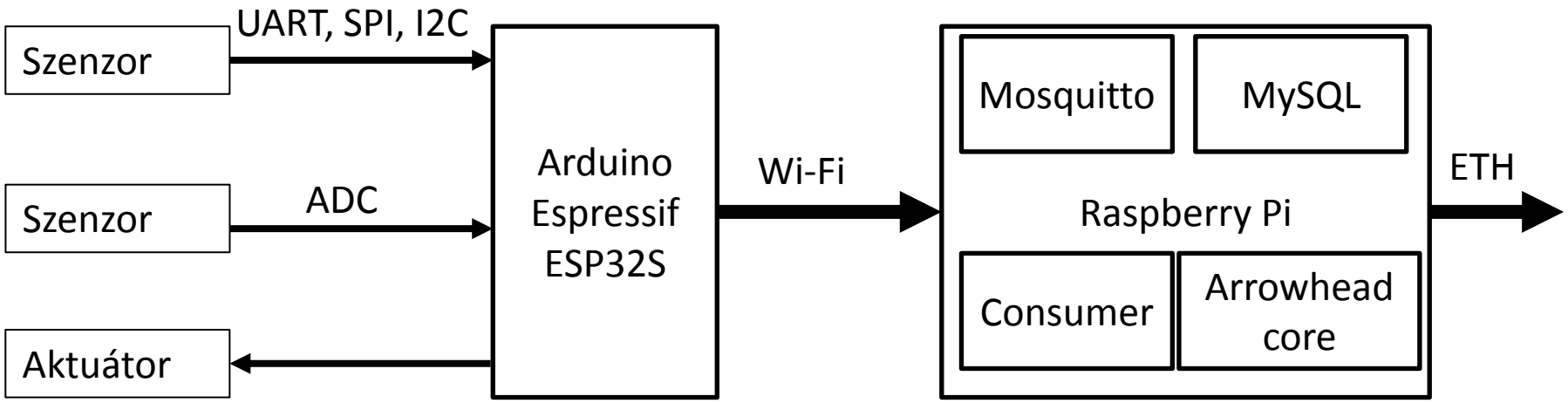
https://learn.adafruit.com/setting-up-a-raspberry-pi-as-a-wifi-access-point/install-software

Raspbian: Debian alapú Linux distro

- Csomagkezelő: apt-get

4 x ARM 1.6 GHz mag, 512 MB RAM, SD kártya

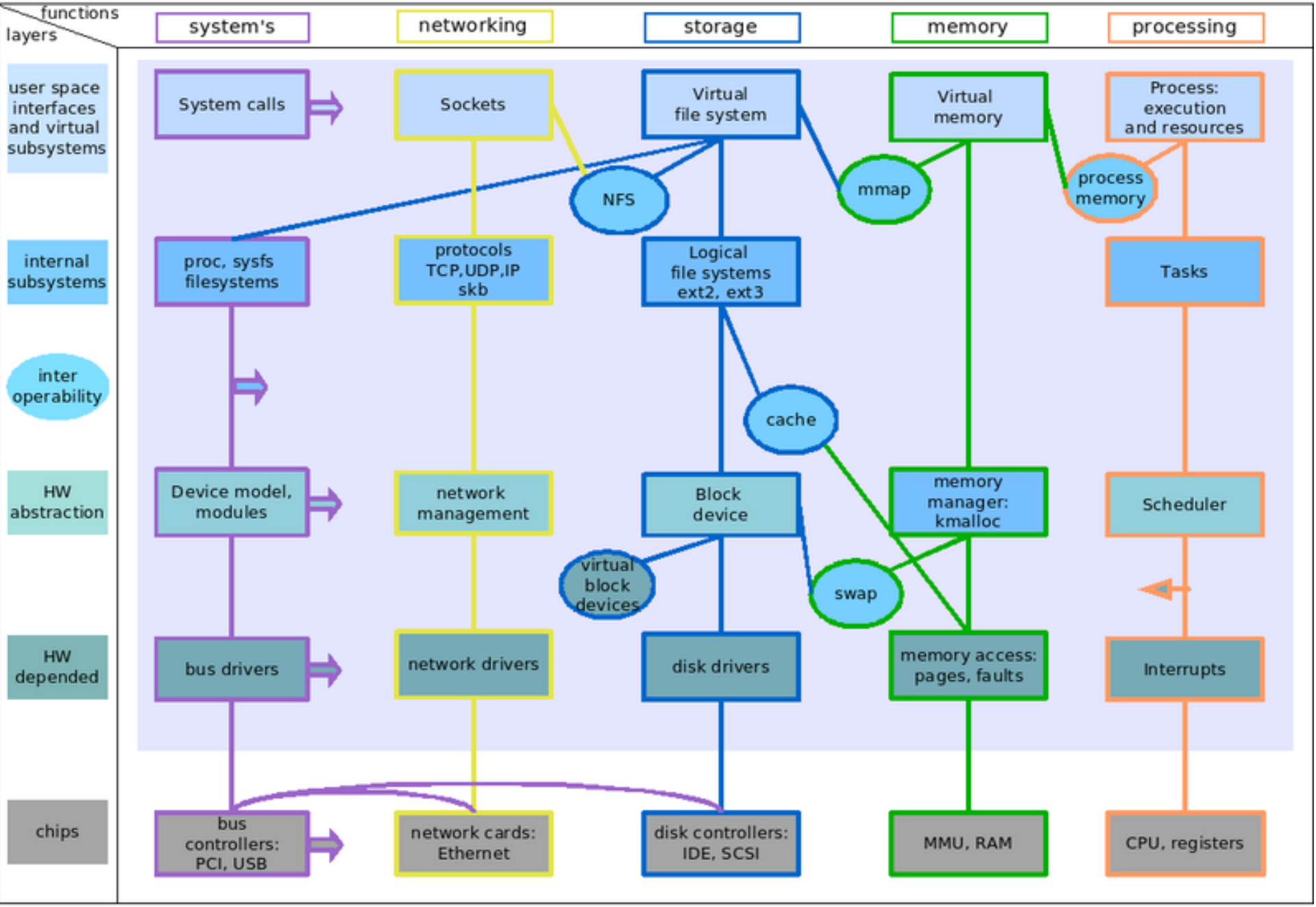
# Házi feladat áttekintés



# Tematika

- Linux bevezető
  - Könyvtárszerkezet
  - Alapvető bash parancsok
  - Futtatás indításkor, „service” létrehozása
- Python
  - Nyelv alapismerete
  - Néhány hasznos könyvtár

# Simplified Linux kernel diagram in form of a matrix map



# File System

- „device” vagy „partition” ami file-ok tárolására van formázva
- Sok formátum – a felhasználó előtt ugyanolyan megjelenés: könyvtárszerkezetben
- /proc/filesystems - pl. ext2 ext3 vfat NFS ntfs

- A hozzáféréshez *mount*-olni kell – így könyvtárként látszik/elérhető

  *mount -t vfat /dev/hda1 /windows/cdrive*
- mount / umount /eject

# File System (cont)

- Applikáció is
  - Az adathozzáférés vezérlésére
  - A könyvtárak kezelésére és hozzáféréséhez
  - Más applikációk is használják a szolgáltatásait
    - állományokat és könyvtárakat készít
    - megnyit és módosít létezőket
    - törlés
    - hozzáférés-szabályozás

# Néhány főbb alkönyvtár

| | |
|---|---|
| /bin | common programs, shared by admins, users |
| /boot | startup files and kernel. Also GRUB data |
| /dev | references to all hardware as special files |
| /etc | important system configuration files. Sort of like Control Panel |
| /home | containes home directories for most users |
| /initrd | information contained for booting.. Don't mess with it! |
| /lib | library files, common files needed by many programs |
| /lost+found | files that were recovered after a system failure |
| /mnt | mount point for all external devices |
| /net | mount point for remote filesystems |
| /opt | extra third party software |
| /proc | info about system resources, man proc |
| /root | home dir. for the the admin |
| /tmp | temporary space |
| /usr | program, libraries, docs for most user programs |
| /var | other temporary files, such as logs, downloaded files, mail queue |

# Alapvető műveletek

- cd — könyvtárak közötti navigációhoz
- mkdir — könyvtár létrehozása
- rmdir — könyvtár törlése
- ls — könyvtár tartalma
- echo $PATH — végrehajtható parancsok helyei
  *pl.: /usr/local/bin/:/usr/bin:/bin/*
- export — környezeti változó beállítása
  *pl. PATH=$PATH:/new/directory/path*
- man — help a parancsok használatához
- history — eddig használt parancsok listája
- & — parancs után írva: fusson a háttérben *ls & -> [1] 23142*
- fg — futó job visszahozása előtérbe *fg 23142*

# Szöveges manipuláció

- cat teljes file szövegének összeállítása (kiírás)
- tac utolsó sor legelöl (hasznos pl. log file esetén)
- head a file kezdő sorai (default:10)
- tail a file záró sorai (default:10)
- more szöveg kiírása oldalanként a terminálra
- less szöveg kiírása: a felhasználó mászkálhat benne

regxp – regular expressions

- grepszövegben keresés szabályok szerint
- (g)awk szövegben keresés/manipuláció
- sed szöveg egyszeri futás alatti szerkesztése
- vi klasszik szövegszerkesztő
- emacs legendás szövegszerkesztő

# Accounting

- Multiuser operációs rendszer
- Account:
  - felhasználónév azonosítja,
  - jelszó védi
- Kétféle account:
  - Root/superuser: bármit megtehet
  - User: saját userspace-én garázdálkodhat csak
- Sudo: superuser do

# Accounting (cont)

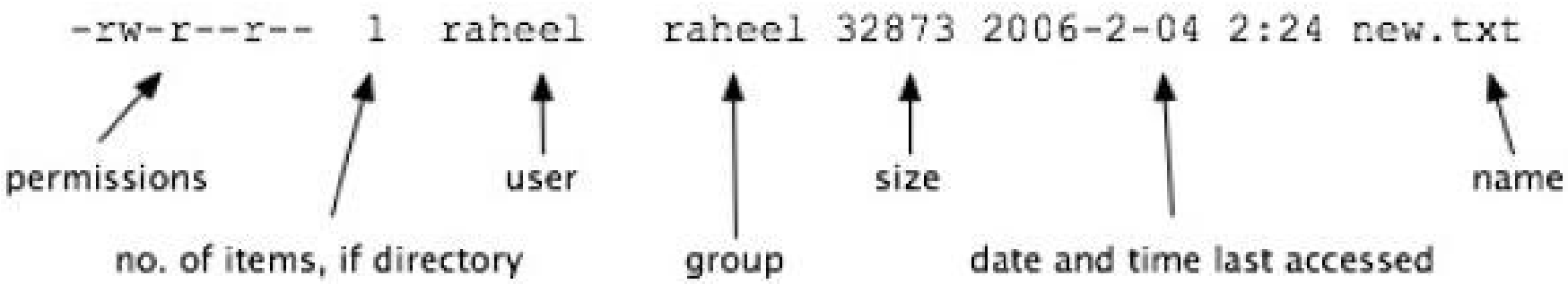
- su — root átvált az adott userre/ré
- Sudo — root jogokkal futtatni egy adott parancsot
- adduser — felhasználó hozzáadása
- passwd — jelszó megváltoztatása
- userdel — felhasználói account törlése
- /etc/passwd — felhasználói account információk
- /etc/shadow — felhasználói jelszó titkosítva; valamint account- és jelszó-lejárati infók

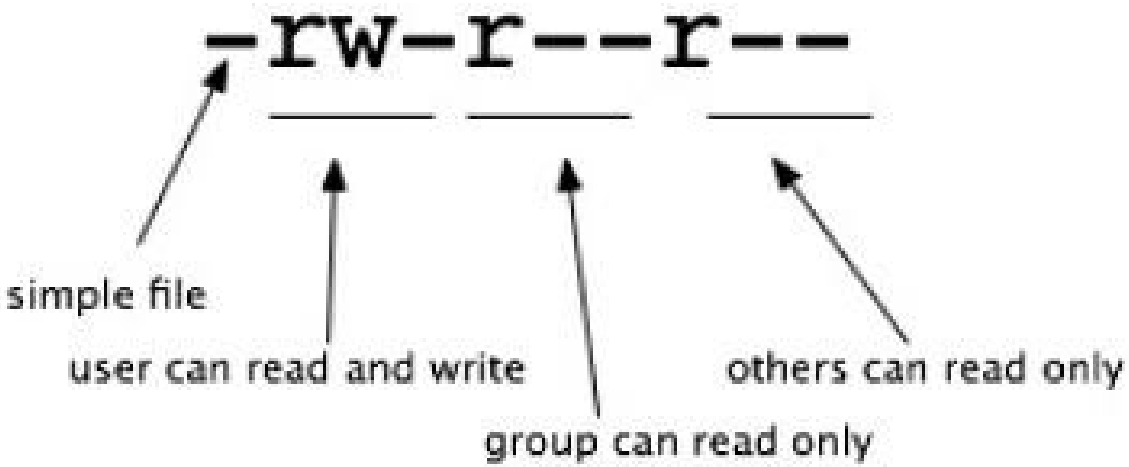
# Authorization: ownership & access

- Hozzáférés
  - Multi-user környezet!
  - Felhasználók nem férnek hozzá egymás file-jaihoz
  - Speciális file-okhoz csak a root fér hozzá
- Csoportok – groups
  - a felhasználók több csoporthoz tartozhatnak
  - könnyebb/rugalmasabb hozzáférés-kezelés
- Hozzáférés: felhasználók / csoportok / egyéb

# Hozzáférés - példa

- „ls – l”  - parancs példa kimenete



# Hozzáférés változtatás

- *chmod* parancs

- *r, w, x: read, write, execute*
- *Root: megváltoztathatja bármely file/directory hozzáférés-szabályait*
- *A root mellett csak a birtokos (user) tudja változtatni*

*pl. chmod 755 myfile*

# Ownership változtatás

- Ownership (birtoklás) változtatás: chown
  *chown username file_or_dir*

- Csoport-birtoklás változtatás: chgrp
  *chgrp groupname file_or_dir*

- Kombinált csoport és felhasználónév:
  *chgrp name.name file_or_dir*

# Automatikus indítás

- „Service” létrehozásával a kódod automatikusan fog elindulni boot után
- Tutorial:

https://www.raspberrypi.org/documentation/linux/usage/systemd.md

# Python 3 bevezető

# Áttekintés

- Miért épp a Python?
- Python referencia & tutorial @ python.org
  - https://docs.python.org/3/tutorial/
  - https://docs.python.org/3/reference/index.html
- Python áttekintés: Cheat Sheets
- Gyakorlatok
- Best Practices
  - Általában a szkripteléshez
  - Python-hoz - https://docs.python-guide.org/
- Python Standard Library - https://docs.python.org/3/library/

# Miért épp a Piton?

- Általános célú nyelv, amin könnyű írni és olvasni
- Nagyon sok lehetőséget kínál, különféle területeken vannak kiváló „library”-jai
  - Machine Learning
  - Big Data
  - Data Science
  - Web Design
  - System Administration
- Egészséges, aktív közösség építi, szépen fejlődik
- Nagyvállalatok (is) szponzorálják

# Python referenciák, tutorial-ok

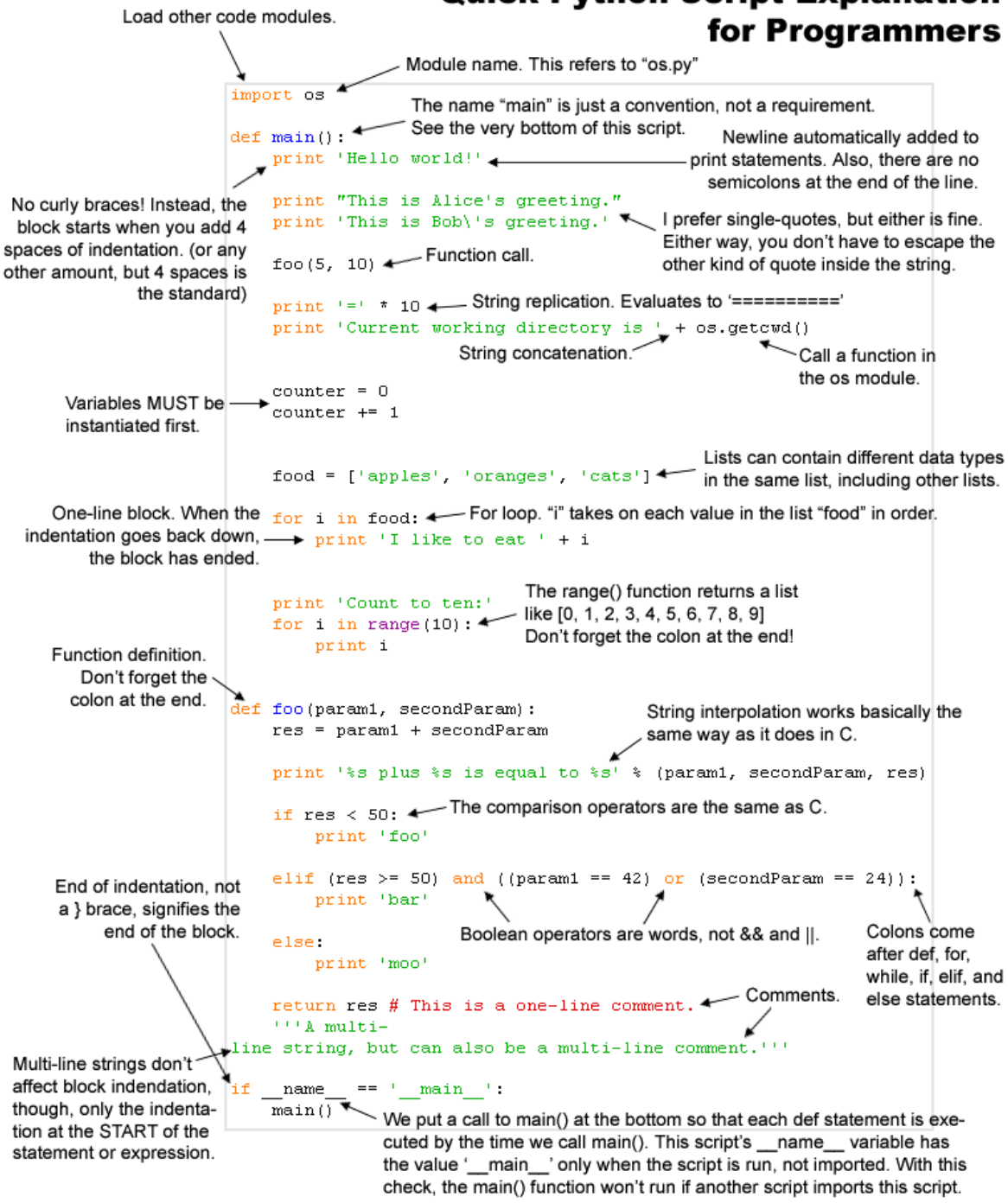
- https://www.sololearn.com/Course/Python/
- https://docs.python.org/3/tutorial/
- https://developers.google.com/edu/python/
- https://docs.python.org/3/reference/index.html

# Cheat Sheets

- https://perso.limsi.fr/pointal/_media/python:cours:mementopython3-english.pdf
- http://www.webpages.uidaho.edu/~stevel/504/Python%20Notes.pdf
- https://ugoproto.github.io/ugo_py_doc/Python-Cheat-Sheet-by-CodeConquestDOTcom.pdf
- https://github.com/ehmatthes/pcc/releases/download/v1.0.0/beginners_python_cheat_sheet_pcc_all.pdf
- http://coffeeghost.net/pybat/python_cheatsheet.png
- https://docs.python-guide.org/ → Hitchhiker's Guide to Python
- …

http://coffeeghost.net

# Quick Python Script Explanation for Programmers



# Példakód

```
import pymysql.cursors
import paho.mqtt.client as mqtt
from datetime import datetime

print("debug1")

def onconnect(client, userdata, flags, rc):
        if rc==0:
                print("connected OK Returned code=", rc)
        else:
                print("Bad connection Returned code=", rc)

client = mqtt.Client("python1")
client.on_connect=onconnect
client.connect("192.168.1.148")
client.subscribe("esp/both")

print("debug2")

def onmessage (client, userdata, message):
        print("received msg;", str(message.payload.decode("utf-8")))
        print("topic:",message.topic)
        if message.topic == "esp/both":
                print("Saving to database")
                connection=pymysql.connect( host="localhost", user="root", password ="root", db="arrowhead", cursorclass=pymysql.cursors.DictCursor)
                try:
                        with connection.cursors() as cursor:
                                now = datetime.now()
                                formDate = now.strftime ("%Y-%m-%d %H:%M:%S")
                                sql = "INSERT INTO tempData (humidity, temperature, time) VALUES (" + str(message.payload.decode("utf-8"))+ ", '" + formDate$
                                print(sql)
                                cursor.execute(sql)
                                connection.commit()
                                print("MySQL commit successful.")
                finally:
                        connection.close()
                print("MySQL connection closed.")
```