

```

public static void main (String[] args)
{
    hplanszaGraf.DrawRectangle(ps_Dlugopis, PS_X -
        PS_Margines, PS_X - PS_Margines, PS_Y / 2, PS_X /
        2);
}

public static void ps_main(String[] args)
{
    ps_Dlugopis.Dispose();
    BufferedReader file_reader = new BufferedReader (new InputStreamReader
        (System.in));
    String text;
    while (!text.equals(""))
    {
        public override void ps_wymaz()
        {
            int a;
            for (int i=0; i<PS_widoczny; i++)
            {
                for (int j=0; j<PS_wysokosc; j++)
                {
                    x[j] = '\0';
                    z[a+j]=x[j];
                }
            }
        }
    }
}

```

# Grundlagen Programmierung 2

Datensourcen, InfluxDB und SQL

# Daten und Datenquellen

## Was sind Daten?

Einführung in Datenquellen

Häufige Datenquellen

Open Data

# Was genau sind "Daten"?

- / Daten sind ein allgemeiner Begriff für numerische Werte, die durch Beobachtungen, Messungen usw. gewonnen werden.
- / Umgangssprachlich werden Daten auch als Sachverhalte, Fakten und Ereignisse bezeichnet. Für die Datenverarbeitung sind Daten als Zeichen oder Symbole definiert, die Informationen repräsentieren und die dem Zweck der Verarbeitung dienen

<https://de.wikipedia.org/wiki/Daten>

# Kategorisierung der Daten

## / Nach Herkunft

- / Primärdaten: Originaldaten aus "1. Hand", z. B. aus Messungen, Interviews, Sensoren
- / Sekundärdaten: von Primärdaten abgeleitete Daten; oft bereinigt, strukturiert, aggregiert, bereits veröffentlicht

## / Nach Struktur

- / Strukturierte Daten: z. B. eine Tabelle in einer Datenbank- halbstrukturierte Daten: z. B. Extensible Markup Language (XML)
- / Unstrukturierte Daten: z. B. E-Mails, Dokumente, Texte, PDFs, Bilder, Video- und Audiodateien

# Was ist eine Datenquelle? (Datasource)

**A data source is the location where data that is being used originates from.**

**Eine Datenquelle ist der Ort, woher die verwendeten Daten stammen**

# Physische Geräte

Alle möglichen physischen Geräte, welche Daten sammeln.  
Beispielsweise für Euch interessant:

- Sensoren
- Kameras
- Computer
  
- Wie werden diese Daten angezogen?

# Webseiten

- / Daten können auch aus dem Web bezogen werden.
- / Unternehmen wie Google nutzen das sogenannte „Crawling“
- / Eine Technik wird als Web Scraping bezeichnet. Das Ziel ist es, gesammelte Daten in Strukturierte Daten zu verwandelt.

# „Flat Files“

Die Datensätze in einer „flachen Datei“ folgen einem einheitlichen Format, und es gibt keine Strukturen zur Indizierung oder Erkennung von Beziehungen zwischen den Datensätzen. Eine flache Datei kann eine reine Textdatei oder eine Binärdatei sein. Beziehungen müssen aus den Daten abgeleitet werden.

## Flat-File (one table)

Patient Id	Name	D.o.B	Gender	Phone	Doctor Id	Doctor	Room
134	Jeff	4-Jul-1993	Male	7876453	01	Dr Hyde	03
178	David	8-Feb-1987	Male	8635467	02	Dr Jekyll	06
198	Lisa	18-Dec-1979	Female	7498735	01	Dr Hyde	03
210	Frank	29-Apr-1983	Male	7943521	01	Dr Hyde	03
258	Rachel	8-Feb-1987	Female	8367242	02	Dr Jekyll	06



# Relationale Datenbanken

Eine relationale Datenbank ist ein System zur elektronischen Datenverwaltung. Die wesentliche Aufgabe einer relationalen Datenbank ist es, große Datenmengen effizient, widerspruchsfrei und dauerhaft zu speichern und die benötigten Teilmengen in unterschiedlichen, bedarfsgerechten Darstellungsformen für Benutzer und Anwendungsprogramme bereitzustellen.

Eine relationale Datenbank besteht aus zwei Teilen:

- der Verwaltungssoftware (= Datenbankmanagementsystem, DBMS)
- der Menge der zu verwaltenden Daten

Die Struktur der Daten wird durch ein Datenbankmodell festgelegt.

Die Abfragesprache ist **SQL (Structured Query Language)**.

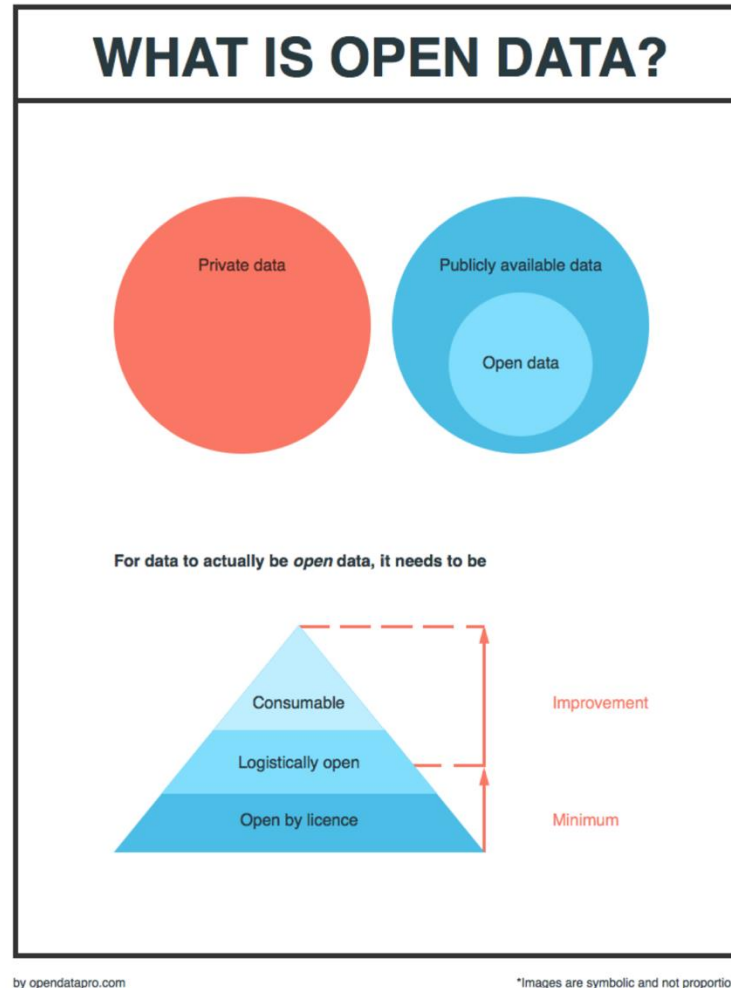
# Open Data

## Warum Open Data?

Was ist Open Data?

Was ist mein nutzen davon?

# Open Data



# Open Data

- / Wikipedia beschreibt «Open Data» als Daten, welche genutzt, verteilt und wiederverwendet werden können, von jedem und zu jedem Zweck.
- / Interessant bei Gebäuden, wenn man herausfinden möchte, was z.B. durchschnittliche Temperatur im Bezirk ist, oder das Haus steuern möchte anhand von Daten, die irgendwo gesammelt werden.
- / Beispiele hier: <http://opendata-showroom.org/de/>

# Datentyp, Datenstruktur

- / Welche Datentypen kennen sie noch aus GP1?
- / Was ist die Aufgabe des Datentyps?
- / Eine Datenstruktur ist eine Datenorganisation, -management oder ein Speicherformat
- / In Python beispielsweise Dictionary und Tupel

# Open Data

/ Ein Fileformat definiert die Syntax und Semantik der Daten im File.

/ Häufige Fileformate:

- CSV
- TXT
- JSON
- XML
- HTML
- ZIP
- XLSX
- PDF
- Bilder files

# Datensammlungsmethoden

## Wie komme ich an Daten?

Einführung  
Häufige Datensammlungsmethoden

# Direkt Download (DDL)

/ Beschreibt einen Hyperlink der dazu genutzt wird, ein File aus dem Internet herunterzuladen

```
import requests

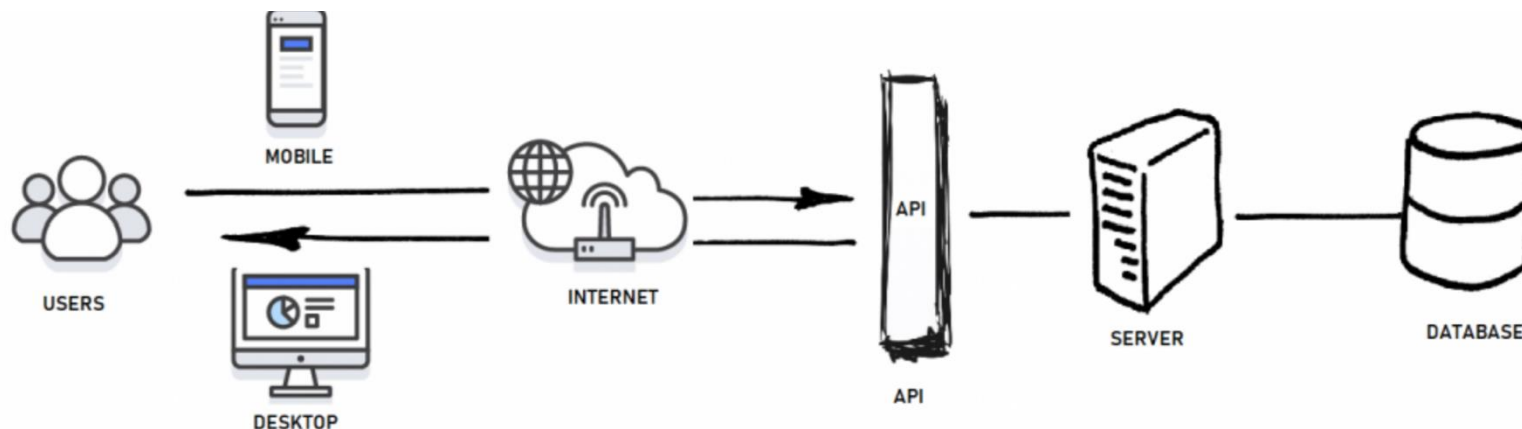
URL = "https://instagram.com/favicon.ico"
response = requests.get(URL)
open("instagram.ico", "wb").write(response.content)
```

Quelle: <https://www.codingem.com/python-download-file-from-url/>



# Web API

- / Die Abkürzung API steht für „Application Programming Interface“
- / Eine Programmierschnittstelle bei der zwei Anwendungen, die voneinander unabhängig sind, problemlos interagieren und Daten austauschen können.

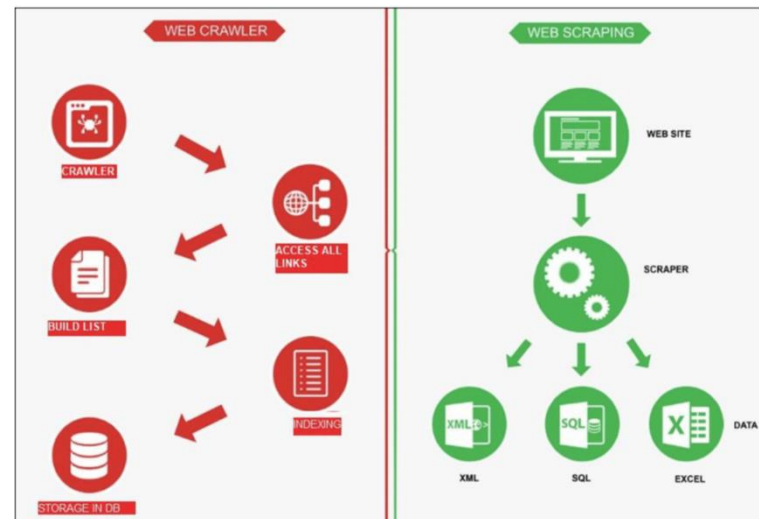


Quelle: <https://levelup.gitconnected.com/how-to-avoid-fat-controllers-in-asp-net-web-api-net-framework-45acb45ba6f0>

# Web Crawler vs. Web Scraping

- / Iterativer Prozess
- / Indexierung der Inhalte und Webseiten

- / Fokus auf den Inhalt  
(Produktinformationen, oder  
allgemeine Informationen)





# Datentypen und Strukturen, Fileformate

## Woher kommen die Daten?

Unterschiede zwischen Datentypen, Datenstrukturen und File  
Formaten  
Häufige Fileformate

# Inhaltsverzeichnis

- / Was ist InfluxDB?
- / Einsatz von InfluxDB
- / Vorteile / Nachteile
- / Time-Series-Konzepte
- / InfluxDB aufsetzen
- / Python-Anbindung
- / Befehle Python

# InfluxDB

- / Datenbankmanagementsystem (DBMS)
- / Open Source
- / Influx 2.0 Cloudbasiert inkl. Benutzeroberfläche
- / Programmiersprache Flux
- / Zentrales Merkmal: Unterstützung von versch. Datenquellen

# Einsatz von InfluxDB

- / Time-Series-Datenbank (TSDB)
- / Speichert Zeitreihen
- / Ideal für Sensordaten oder Protokolle mit Zeitstempeln
- / Kann mit Millionen zeitgleicher Datensätze umgehen

Sensor	Wert	Zeit
Sensor 1	100,00	01/02/2024 @ 10:00
Sensor 2	104,00	01/02/2024 @ 10:00
Sensor 1	100,02	01/02/2024 @ 10:05

# Konzept: Time-Series

- / Zeitstempel → Zeitpunkt der Erfassung
- / Messwert und Felder → 1-n Messwerte (Temperatur, Druck, etc.)
- / Schemalos → Neue Felder können problemlos zu bestehender Struktur hinzugefügt werden
- / Aggregation und Downsampling → Rohdatenmenge reduzieren und gleichzeitig wichtige Informationen beibehalten
- / Retention Policies → Bereinigung der Daten automatisch (nach Ablauf einer zeitlichen Frist)



## Vorteile / Nachteile

- / Geschwindigkeit (+) (Traditionelle Datenbanken brauchen für das Indexing länger)
- / Serverless-Infrastruktur (+)
- / Schema-less-Design (+)
- / Limitierte Secondary-Indexing (-)
- / Ressourcen intensiv (-)

# Datenpunkt in InfluxDB

/ Ein Datenpunkt besteht aus:

/ **Messung:**

/ Logische Sammlung von Datenpunkten, die thematisch zusammengehören, ähnlich wie eine Tabelle in einer relationalen Datenbank. Jede Messung besteht aus einem Namen und einer Sammlung von Feldern und Tags.

/ **Tags:**

/ Schlüssel-Wert-Paare, die Metadaten zu den Datenpunkten enthalten und für effizientes Filtern und Gruppieren verwendet werden. Tags werden indexiert, was schnelle Abfragen ermöglicht.

/ **Felder:**

/ Die eigentlichen Datenwerte. Jedes Feld hat einen Namen und einen Wert. Im Gegensatz zu Tags werden Felder nicht indexiert und sind daher flexibler in Bezug auf die Art der gespeicherten Daten.

/ **Zeitstempel:**

/ Der Zeitpunkt, zu dem der Datenpunkt erfasst wurde.

## Einzelaufgabe (20 Min)

/ Installiere InfluxDB:

/ <https://docs.influxdata.com/influxdb/v2/get-started/setup/>

/ <https://docs.influxdata.com/influxdb/cloud/api-guide/client-libraries/python/>



[https://miro.medium.com/v2/resize:fit:1100/format:webp/1\\*MaEmNa5X1tR4XQP6VyKZ7g.png](https://miro.medium.com/v2/resize:fit:1100/format:webp/1*MaEmNa5X1tR4XQP6VyKZ7g.png)

# Inhaltsverzeichnis

- / - Datenbanken Einführung
- / - SQL
- / - SQL vs NoSQL
- / - NoSQL

# SQL Erstelle neue Datenbank

/ Öffne -> InfluxDB CLI (Starte Terminal/Eingabeaufforderung, Befehl: influx)

/ Zeige alle Datenbanken:

```
SHOW DATABASES
```

/ Erstelle eine neue Datenbank:

```
CREATE DATABASE mydb
```

/ Überprüfe ob die Datenbank existiert

```
SHOW DATABASES
```

# SQL Daten einfügen & Lesen

## / Daten in Datenbank einfügen

```
INSERT weather,location=us-midwest temperature=82 1465839830100400200;
```

/ Notiz: Hierbei handelt es sich um einen Datenpunkt mit folgender Struktur:

/ Messung: 'weather'

/ Tags: 'location=us-midwest'

/ Felder: 'temperature=82'

/ Zeitstempel '1465839830100400200' (Unix-Nanosekunden-Timestamp)

## / Daten aus Datenbank lesen

```
SELECT * FROM mydb
```

## Einzelaufgabe (20 Min)

- / Erstelle eine neue Datenbank mittels InfluxDB CLI
- / Schreibe Beispieldaten zum Wetter in Luzern
- / Fragen diese Daten von der Datenbank ab



## Einzel Aufgabe (30 Min)

- / Nutze das bereitgestellte File «einzel Aufgabe\_influxdb.py»
- / Erstelle eine SQL-ähnliche Abfrage zur Analyse der Daten
  - / Durchschnittliche Temperatur bspw.
- / Nutze Tags und Felder für spezifische Abfragen aus Python
  - / Spezifische Daten für Standort
  - / Abfragen von Daten basierend auf einem Temperaturbereich