What is Data?

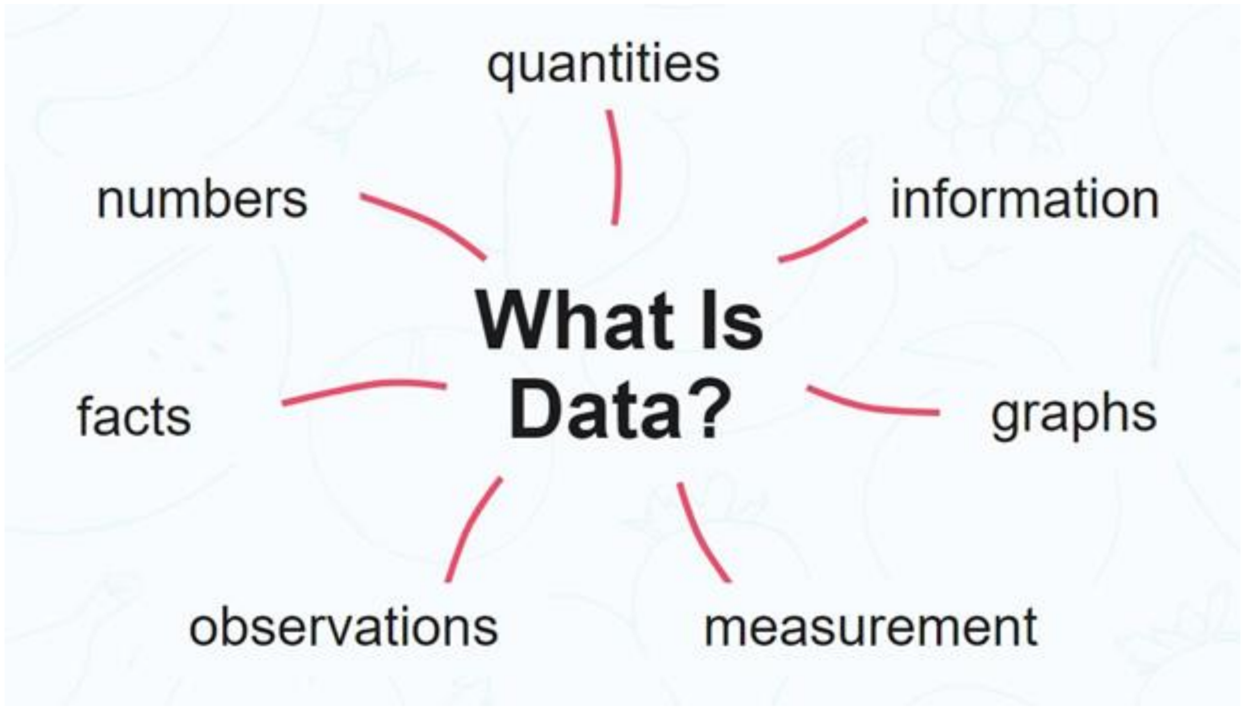
Topics we will cover

1. Types of Data
   a. Broader Categories of Data
2. Data Formats
3. How to get Data?

How to structure and represent your data efficiently is crucial for optimal performance and accurate results?

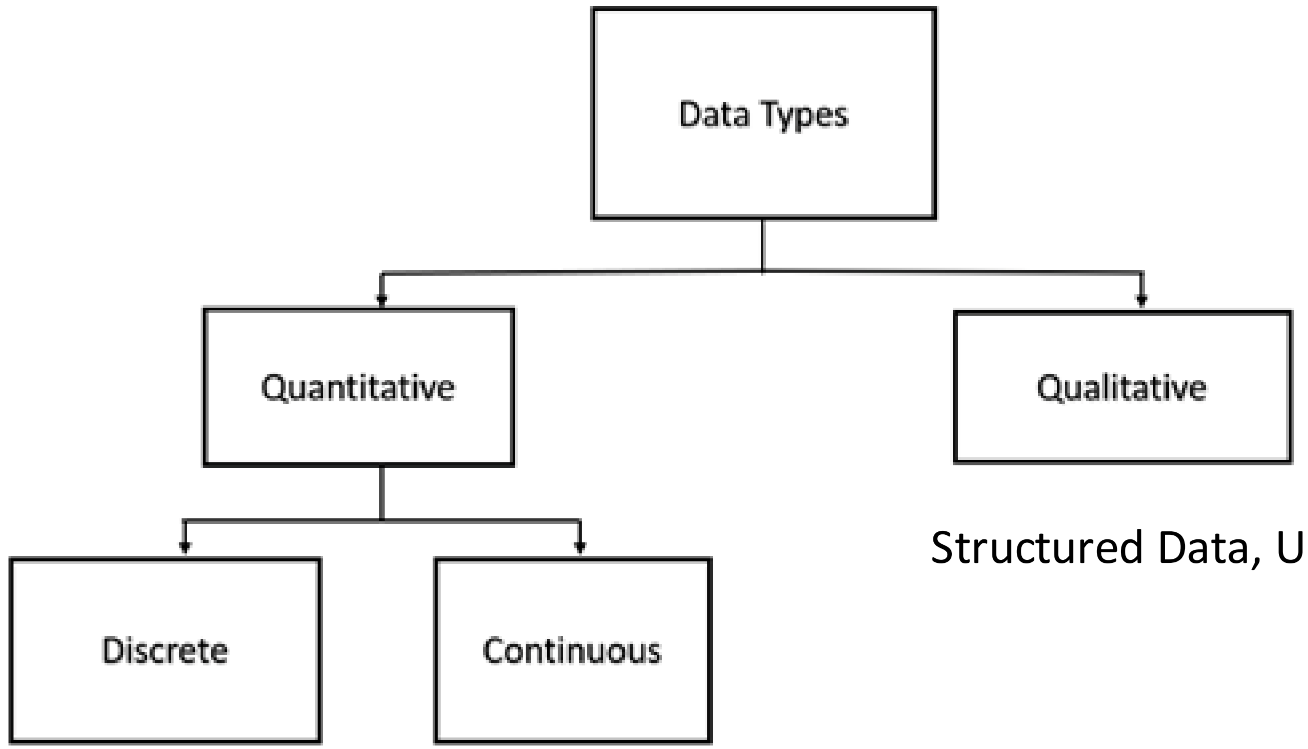
Data

Data is raw information, facts, or statistics that can be in various forms such as numbers, text, images, or more.

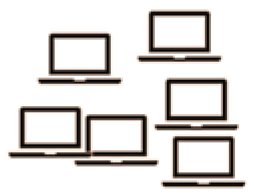


Data Types

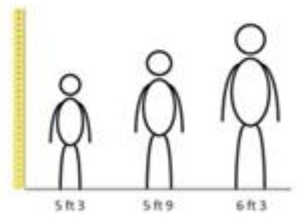
Broad Category of Data



Structured Data, Unstructured Data


No. of Laptops

No. of Cars



Height

Time

Types of Data

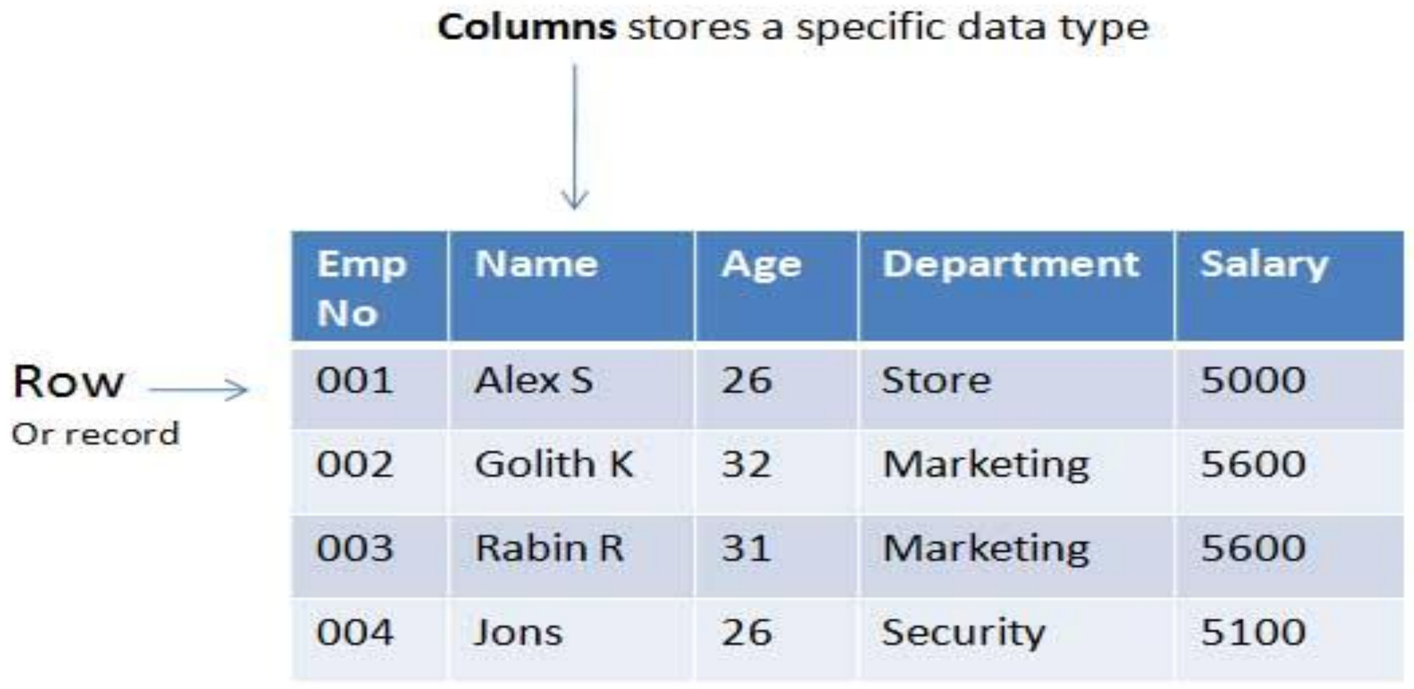
1. **Structured Data**
    a) Tabular Data
    b) Time-Based Data
2. **Graph**
3. **Unstructured Data**
    a) Text Data
    b) Image Data, Video Data
4. **Many more**

Structure Data

Tabular Data–(Things that are in tables): Structured data organized into rows and columns, often resembling a spreadsheet or database table.

Example:

- Demographic info
- Grades
- Many more….


Columns stores a specific data type
Row
Or record

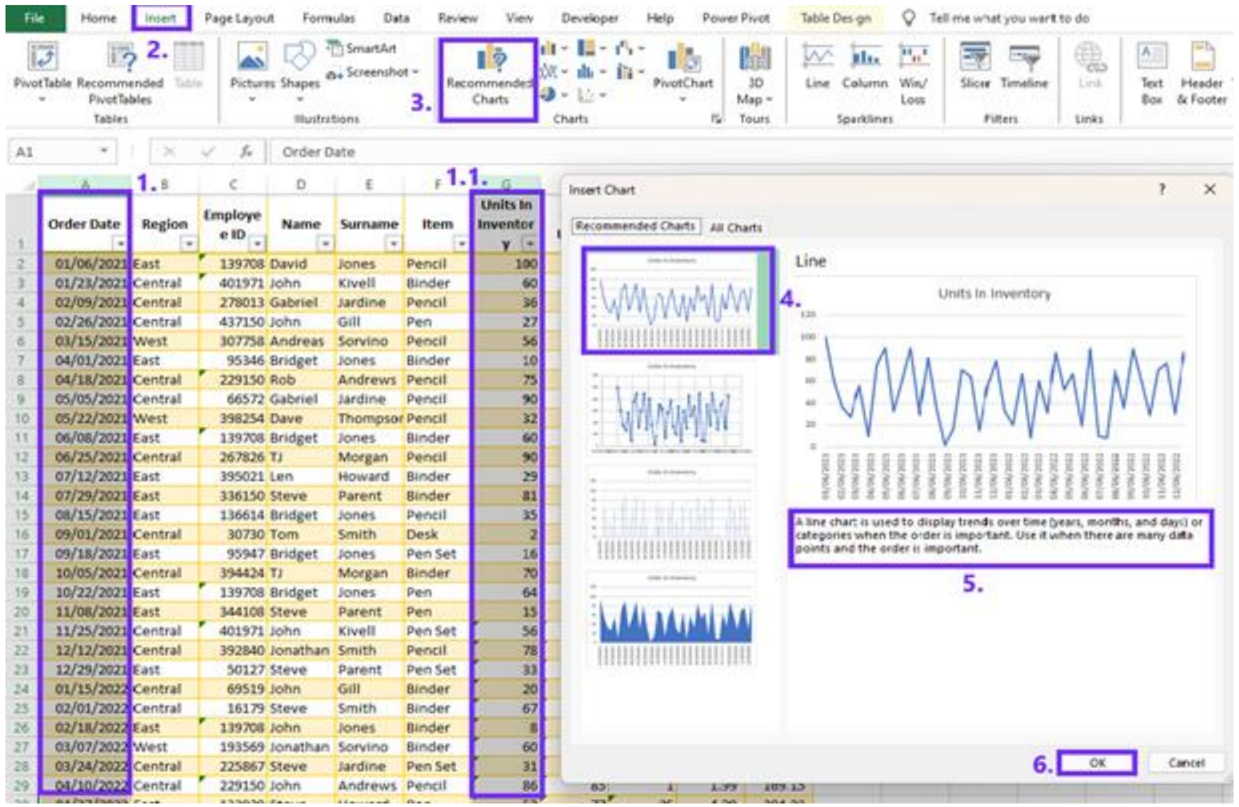
Emp No	Name	Age	Department	Salary
001	Alex S	26	Store	5000
002	Golith K	32	Marketing	5600
003	Rabin R	31	Marketing	5600
004	Jons	26	Security	5100

Time-Based Data

Also known as temporal data

Data that is recorded or organized in relation to **specific timestamps or time intervals.**

- Track changes, trends, and patterns over time
  - Finance, weather forecasting, business analytics, and scientific research.



	Order Date	Region	Employee ID	Name	Surname	Item	Units In Inventory
2	01/06/2021	East	139708	David	Jones	Pencil	100
3	01/23/2021	Central	401971	John	Kivell	Binder	60
4	02/09/2021	Central	278013	Gabriel	Jardine	Pencil	36
5	02/26/2021	Central	437150	John	Gill	Pen	27
6	03/15/2021	West	307758	Andreas	Sorvino	Pencil	56
7	04/01/2021	East	95346	Bridget	Jones	Binder	10
8	04/18/2021	Central	229150	Rob	Andrews	Pencil	75
9	05/05/2021	Central	66572	Gabriel	Jardine	Pencil	90
10	05/22/2021	West	398254	Dave	Thompson	Pencil	32
11	06/08/2021	East	139708	Bridget	Jones	Binder	60
12	06/25/2021	Central	267826	TJ	Morgan	Pencil	90
13	07/12/2021	East	395021	Len	Howard	Binder	29
14	07/29/2021	East	336150	Steve	Parent	Binder	81
15	08/15/2021	East	136614	Bridget	Jones	Pencil	35
16	09/01/2021	Central	30730	Tom	Smith	Desk	2
17	09/18/2021	East	95947	Bridget	Jones	Pen Set	16
18	10/05/2021	Central	394424	TJ	Morgan	Binder	70
19	10/22/2021	East	139708	Bridget	Jones	Pen	64
20	11/08/2021	East	344108	Steve	Parent	Pen	15
21	11/25/2021	Central	401971	John	Kivell	Pen Set	56
22	12/12/2021	Central	392840	Jonathan	Smith	Pencil	78
23	12/29/2021	East	50127	Steve	Parent	Pen Set	33
24	01/15/2022	Central	69519	John	Gill	Binder	20
25	02/01/2022	Central	16179	Steve	Smith	Binder	67
26	02/18/2022	East	139708	John	Jones	Binder	8
27	03/07/2022	West	193569	Jonathan	Sorvino	Binder	60
28	03/24/2022	Central	225867	Steve	Jardine	Pen Set	31
29	04/10/2022	Central	229150	John	Andrews	Pencil	86

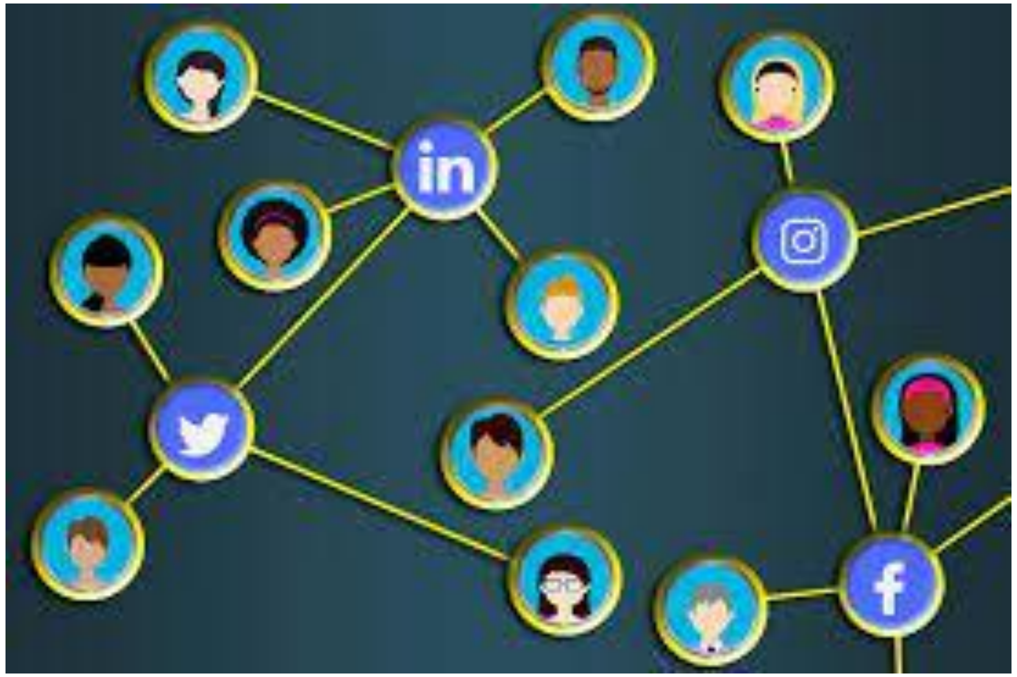
A line chart is used to display trends over time (years, months, and days) or categories when the order is important. Use it when there are many data points and the order is important.

Types of Data cont.

Graph: Represents relationships between entities using nodes (vertices) and edges.

Examples:

- Social connections
- Websites
- Network traffic
- Roads



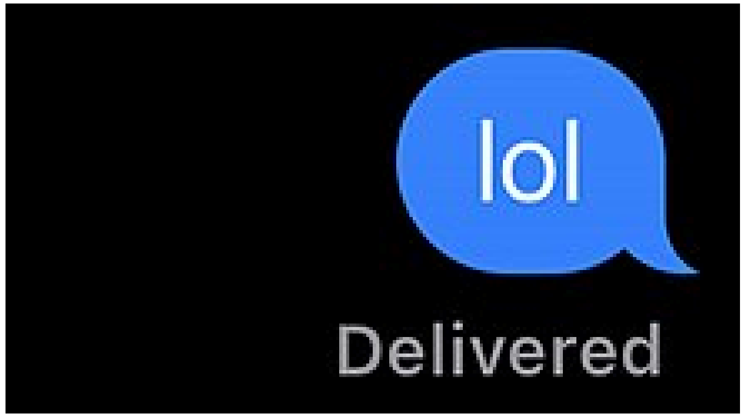
Types of Data cont.

Unstructured Data: lacks a predefined **structure** or **format**, challenging to analyze and process.

- **Videos**
  - Tik Tok
- **Images**
  - James Webb
  - Faces
  - Handwriting
  - Road signs
- **Audio**
  - Alexa
  - Real-time translation
  - Music
- **Biometrics**
  - Fingerprints
  - Facial recognition
- **Haptics**
  - Phone vibrates to notify you of a message,

Unstructured Data: Text

Text: human-readable text


lol

Delivered

Examples:

- Reviews, Books, Articles, Emails
- Translation
- ChatGPT → generate human-like text responses in a conversational manner.
- Social Media Post

Data Formats in Data Science

Determine how data is organized and how efficiently it can be read, written, and processed.

Data Formats

- **CSV / TSV**
- **Image**
    - .jpg
    - .png
- **Audio**
    - .wav
    - .mpg
- **JSON**
- **XML / HTML**

- **No SQL Database**
    - Bigtable
    - Accumulo

- **SQL Database**
    - mySQL
    - Postgres
    - etc...

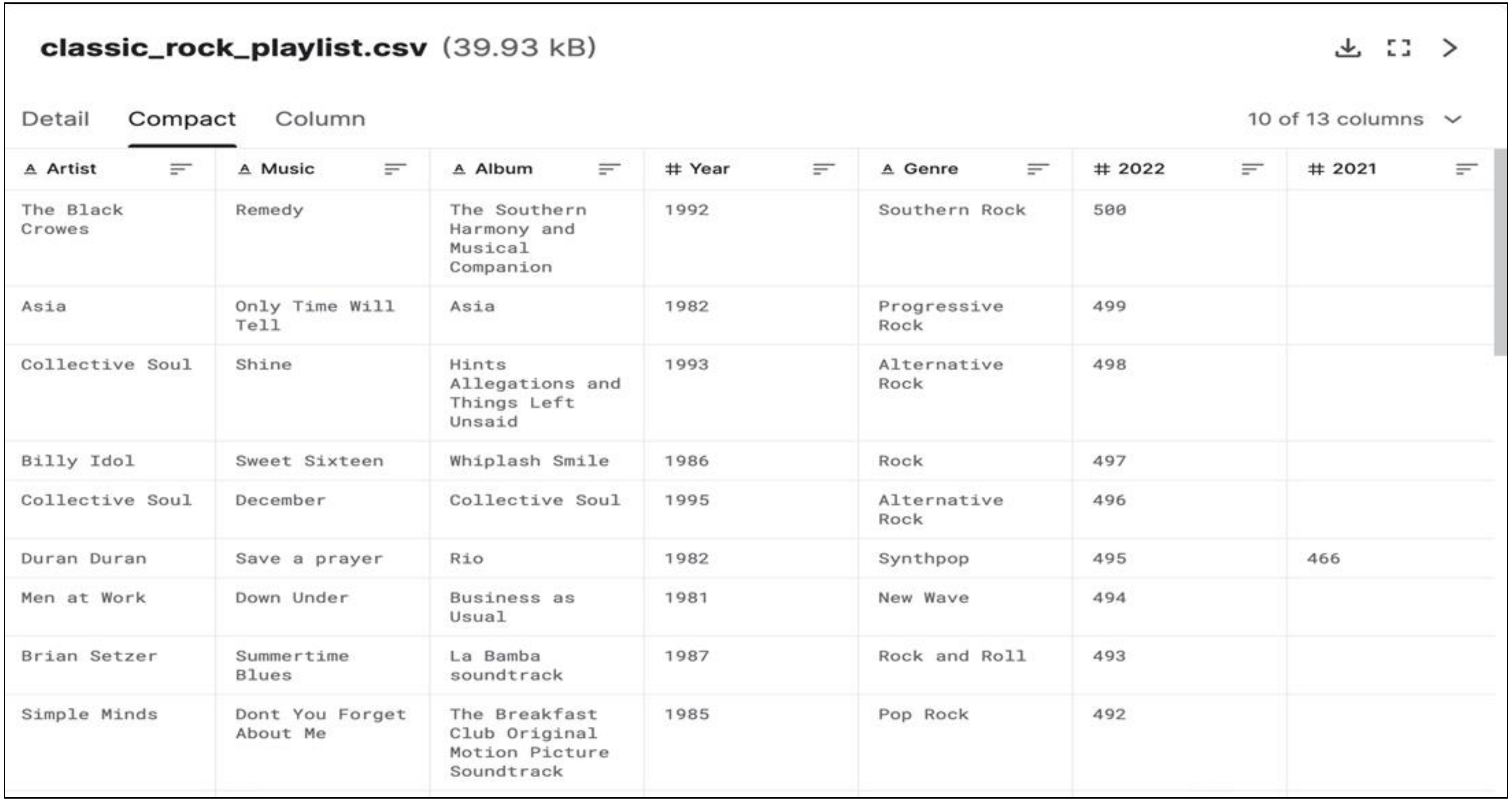
CSV/ TSV to Store Tabular Data

- CSV (Comma-Separated Values)
- TSV (Tab-Separated Values)

Any CSV reader worth anything can parse files with any delimiter, not just a comma.

Delimiter: The separator character : the comm (,), the tab (\t), colon (:) and semi-colon (;) characters.

Tabular Data: Example



classic_rock_playlist.csv (39.93 kB)

Detail | Compact | Column — 10 of 13 columns

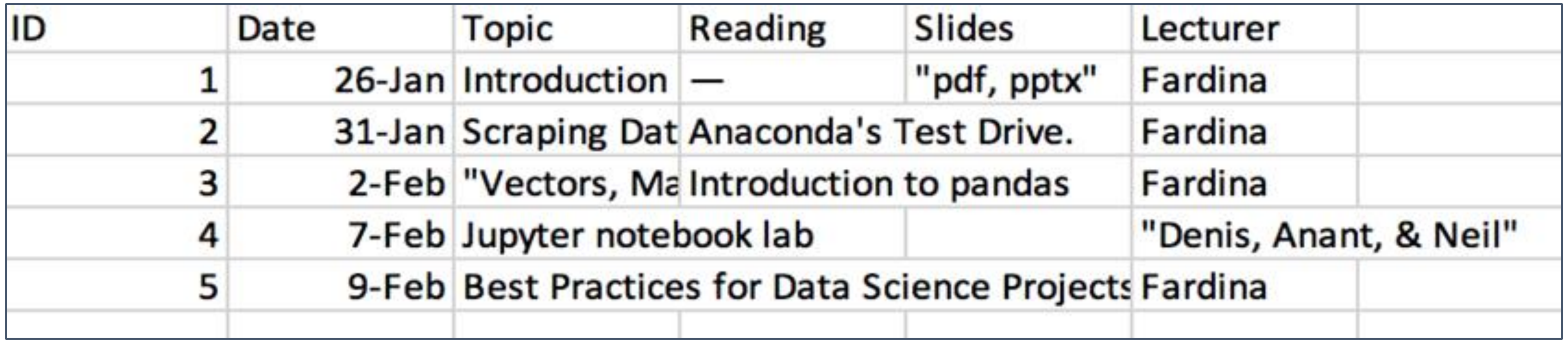
Artist	Music	Album	Year	Genre	2022	2021
The Black Crowes	Remedy	The Southern Harmony and Musical Companion	1992	Southern Rock	500	
Asia	Only Time Will Tell	Asia	1982	Progressive Rock	499	
Collective Soul	Shine	Hints Allegations and Things Left Unsaid	1993	Alternative Rock	498	
Billy Idol	Sweet Sixteen	Whiplash Smile	1986	Rock	497	
Collective Soul	December	Collective Soul	1995	Alternative Rock	496	
Duran Duran	Save a prayer	Rio	1982	Synthpop	495	466
Men at Work	Down Under	Business as Usual	1981	New Wave	494	
Brian Setzer	Summertime Blues	La Bamba soundtrack	1987	Rock and Roll	493	
Simple Minds	Dont You Forget About Me	The Breakfast Club Original Motion Picture Soundtrack	1985	Pop Rock	492	

Representation of tabular data: how how data might be structured in CSV Files

```
Artist,Music,Album,Year,Genre,2022,2021,2020,2019,2018,2017,2016,2015

The Black Crowes,Remedy,The Southern Harmony and Musical
Companion,1992,Southern Rock,500,,324,290,132,64,36,

Asia,Only Time Will Tell,Asia,1982,Progressive Rock,499,,,,,,,

Collective Soul,Shine,Hints Allegations and Things Left Unsaid,1993,Alternative
Rock,498,,,419,485,403,,
```

Python's **pandas library** makes it easy to load (`df = pd.read_csv('data.csv')`) and manipulate CSV data.

CSV Files in Python



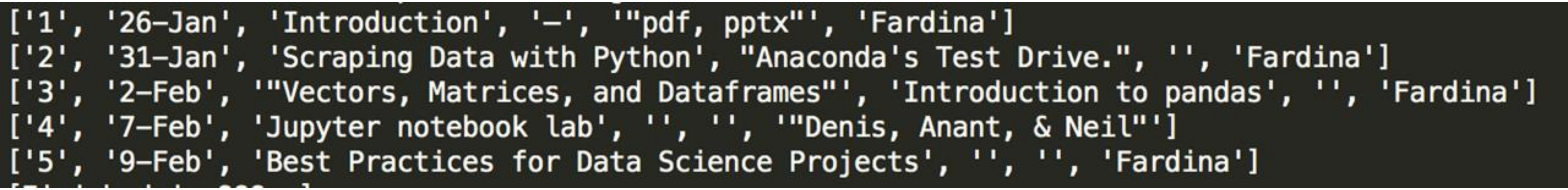
ID	Date	Topic	Reading	Slides	Lecturer	
1	26-Jan	Introduction	—	"pdf, pptx"	Fardina	
2	31-Jan	Scraping Dat	Anaconda's Test Drive.		Fardina	
3	2-Feb	"Vectors, Ma	Introduction to pandas		Fardina	
4	7-Feb	Jupyter notebook lab			"Denis, Anant, & Neil"	
5	9-Feb	Best Practices for Data Science Projects			Fardina	

Input file: schedule.csv

Don't write your own CSV or JSON parser

```python
import csv
with open("schedule.csv", "r") as f:
    reader = csv.reader(f, delimiter= ",", quotechar='"')
    next(reader)
    for row in reader:
        print(row)
```

Output:

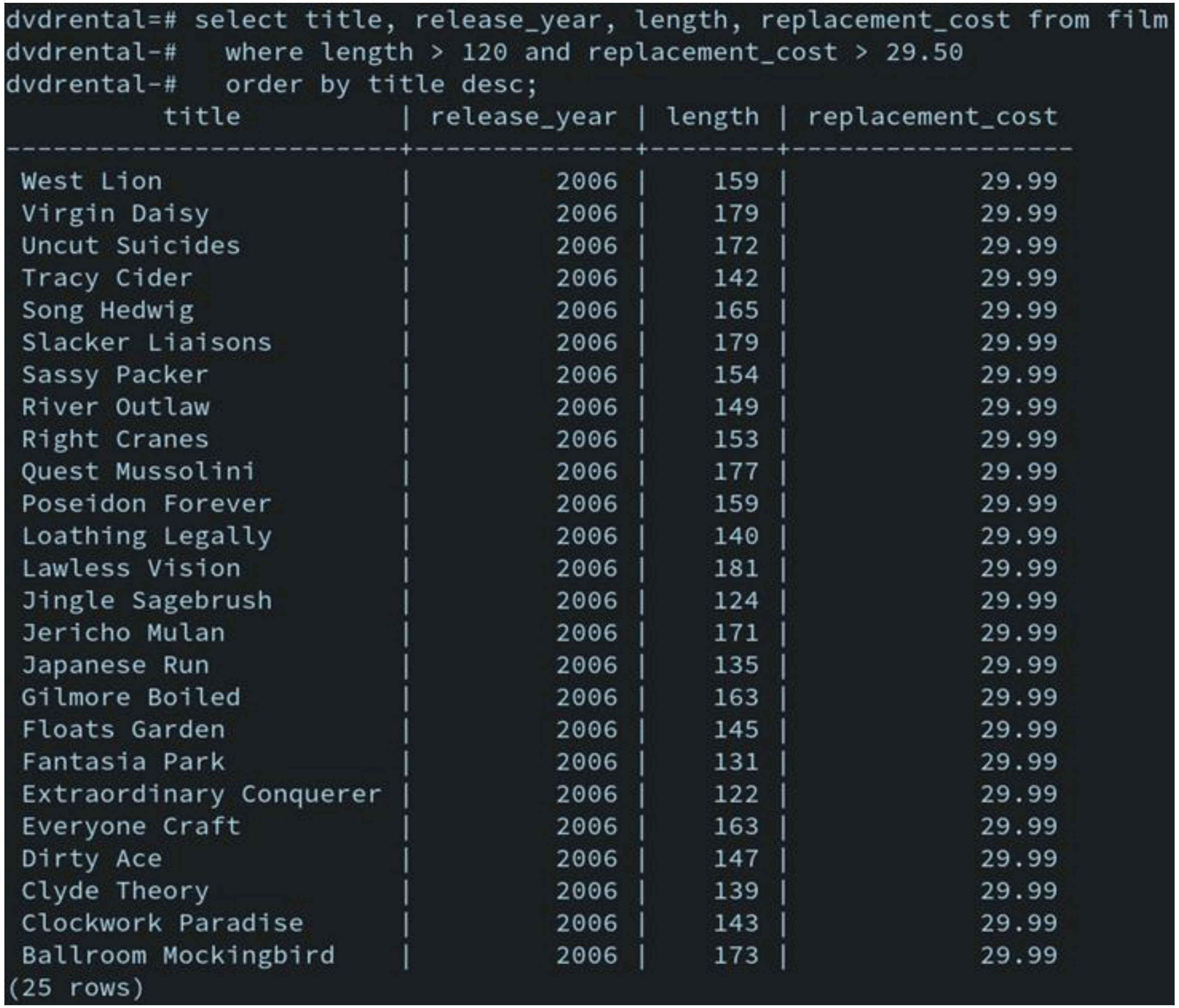


```
['1', '26-Jan', 'Introduction', '—', '"pdf, pptx"', 'Fardina']
['2', '31-Jan', 'Scraping Data with Python', "Anaconda's Test Drive.", '', 'Fardina']
['3', '2-Feb', '"Vectors, Matrices, and Dataframes"', 'Introduction to pandas', '', 'Fardina']
['4', '7-Feb', 'Jupyter notebook lab', '', '', '"Denis, Anant, & Neil"']
['5', '9-Feb', 'Best Practices for Data Science Projects', '', '', 'Fardina']
```

(We'll use pandas to do this much more easily and efficiently)

Databases

A database is an organized collection of structured information, or data that handle more complex data relationships, often organized in tables.

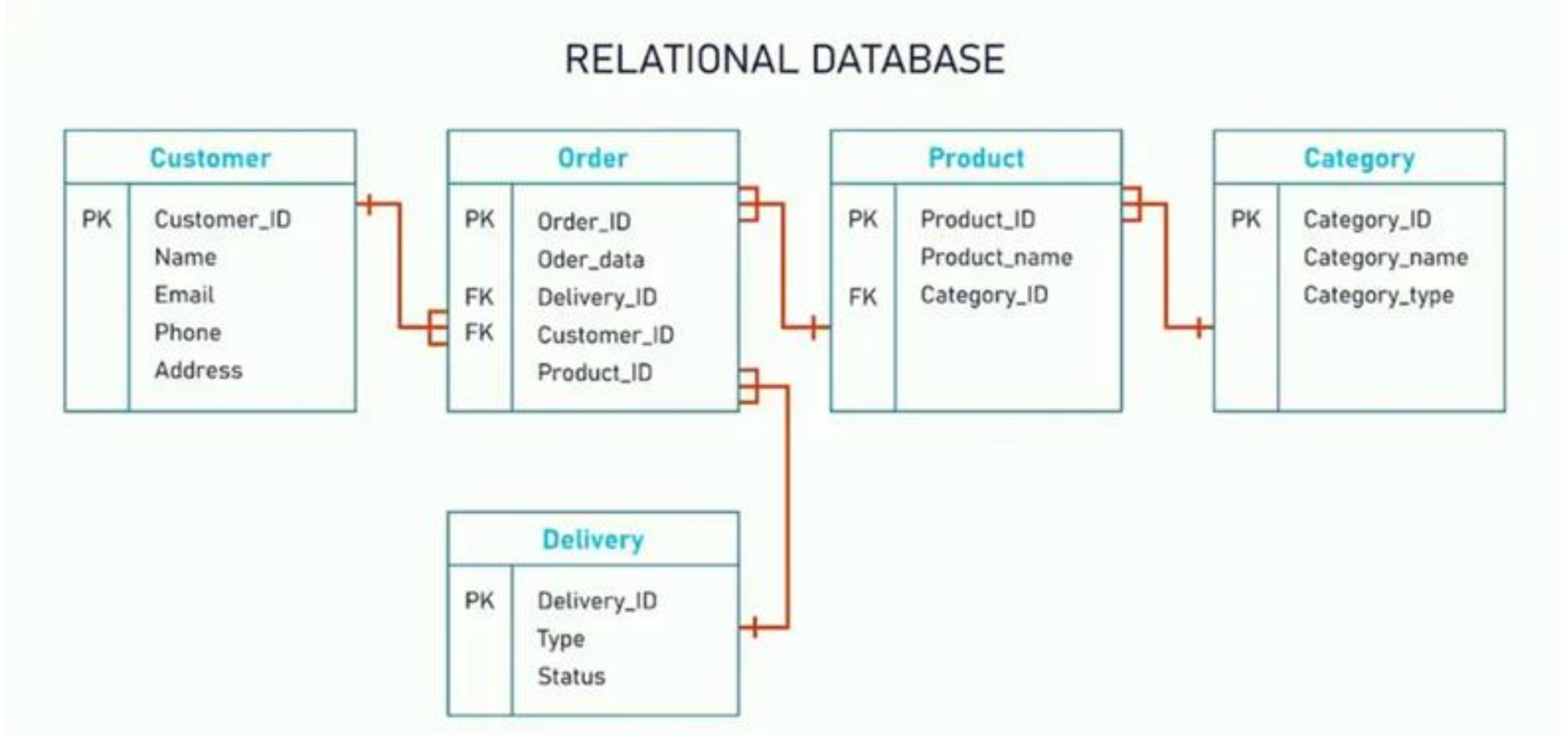


```
dvdrental=# select title, release_year, length, replacement_cost from film
dvdrental-#   where length > 120 and replacement_cost > 29.50
dvdrental-#   order by title desc;
          title          | release_year | length | replacement_cost
-------------------------+--------------+--------+------------------
 West Lion               |         2006 |    159 |            29.99
 Virgin Daisy            |         2006 |    179 |            29.99
 Uncut Suicides          |         2006 |    172 |            29.99
 Tracy Cider             |         2006 |    142 |            29.99
 Song Hedwig             |         2006 |    165 |            29.99
 Slacker Liaisons        |         2006 |    179 |            29.99
 Sassy Packer            |         2006 |    154 |            29.99
 River Outlaw            |         2006 |    149 |            29.99
 Right Cranes            |         2006 |    153 |            29.99
 Quest Mussolini         |         2006 |    177 |            29.99
 Poseidon Forever        |         2006 |    159 |            29.99
 Loathing Legally        |         2006 |    140 |            29.99
 Lawless Vision          |         2006 |    181 |            29.99
 Jingle Sagebrush        |         2006 |    124 |            29.99
 Jericho Mulan           |         2006 |    171 |            29.99
 Japanese Run            |         2006 |    135 |            29.99
 Gilmore Boiled          |         2006 |    163 |            29.99
 Floats Garden           |         2006 |    145 |            29.99
 Fantasia Park           |         2006 |    131 |            29.99
 Extraordinary Conquerer |         2006 |    122 |            29.99
 Everyone Craft          |         2006 |    163 |            29.99
 Dirty Ace               |         2006 |    147 |            29.99
 Clyde Theory            |         2006 |    139 |            29.99
 Clockwork Paradise      |         2006 |    143 |            29.99
 Ballroom Mockingbird    |         2006 |    173 |            29.99
(25 rows)
```

Relational Database

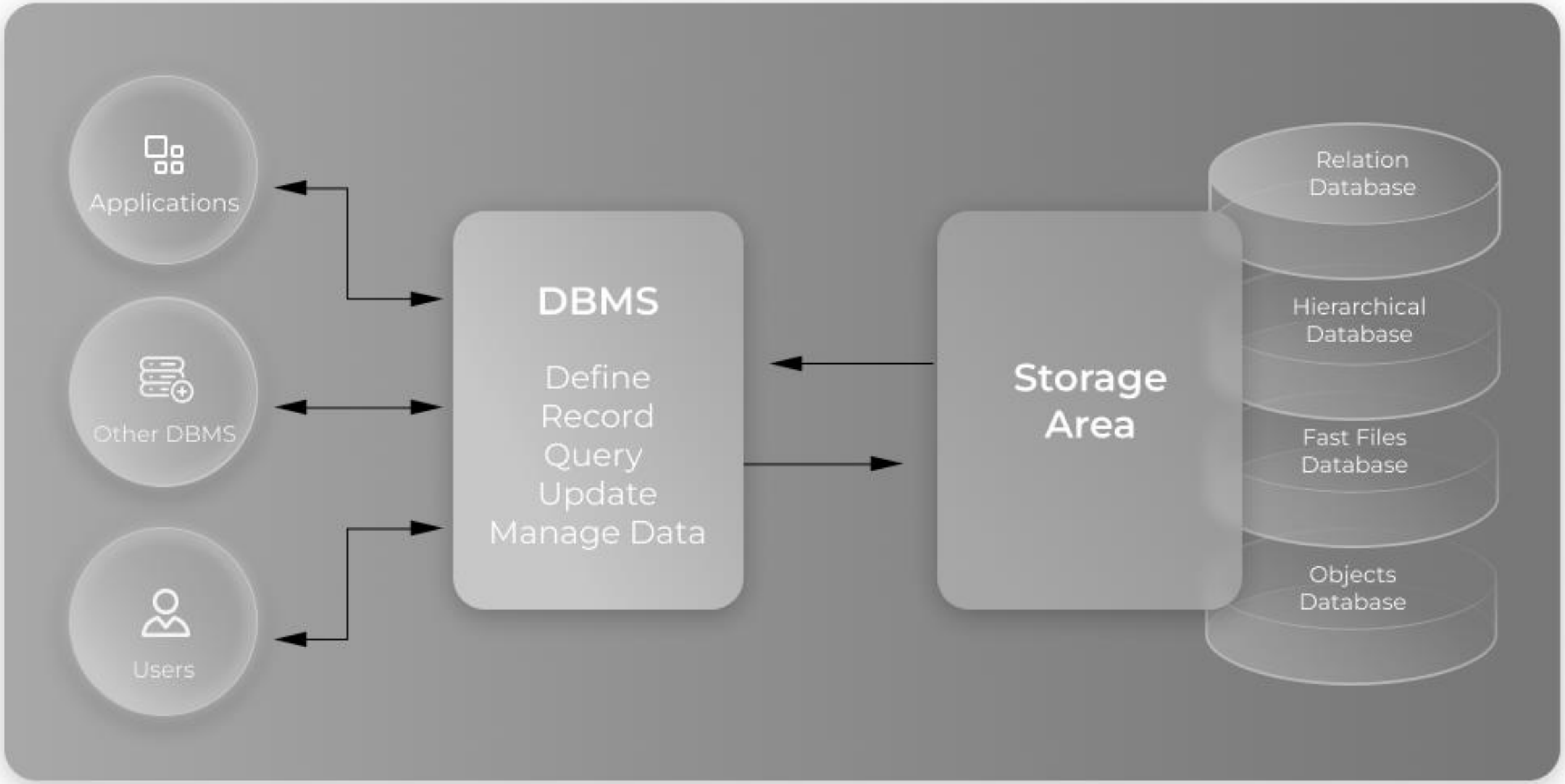
A type of database that organizes data into tables (also called relations) which are linked together by relationships (such as foreign keys).

- Typically queried using SQL (Structured Query Language) to retrieve, insert, update, and delete data.



DBMS (Database Management System)

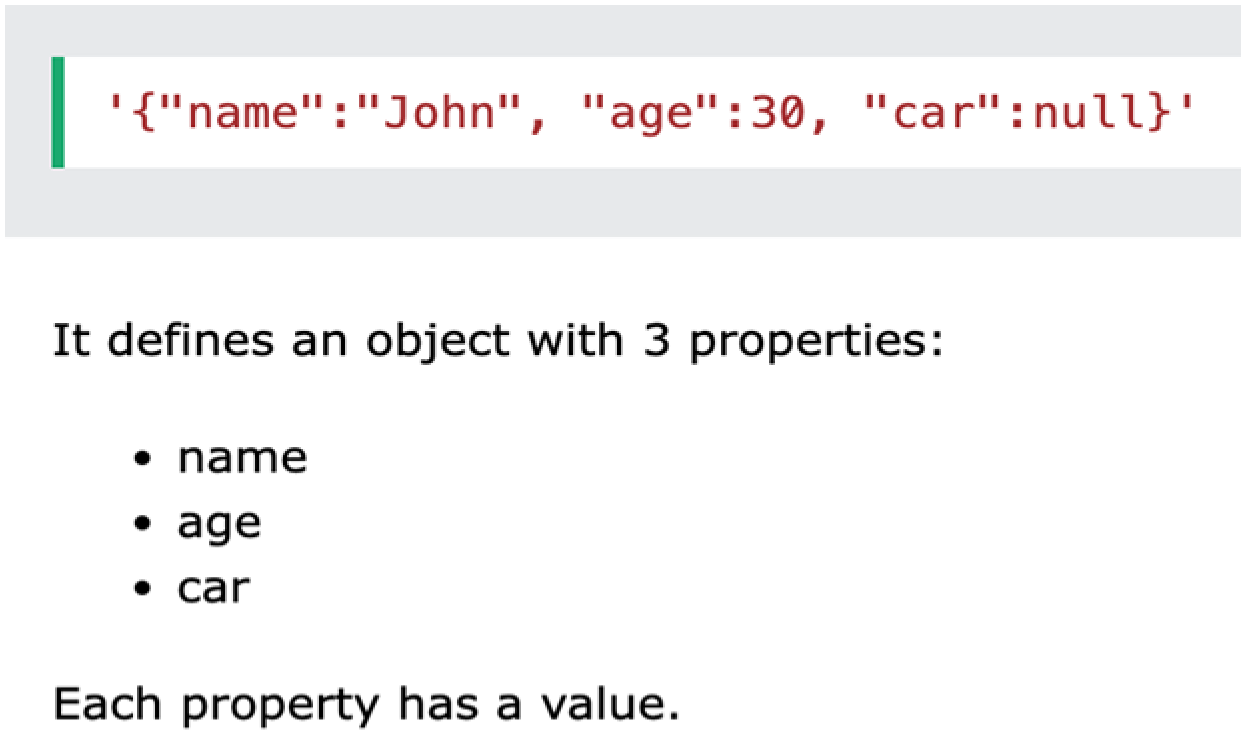
Software that manages databases and provides an interface for interacting with the stored data.



JSON (JavaScript Object Notation)

A lightweight, text-based format used for **representing structured data in key-value pairs.**

- Supports hierarchical data, which makes it suitable for more complex data.

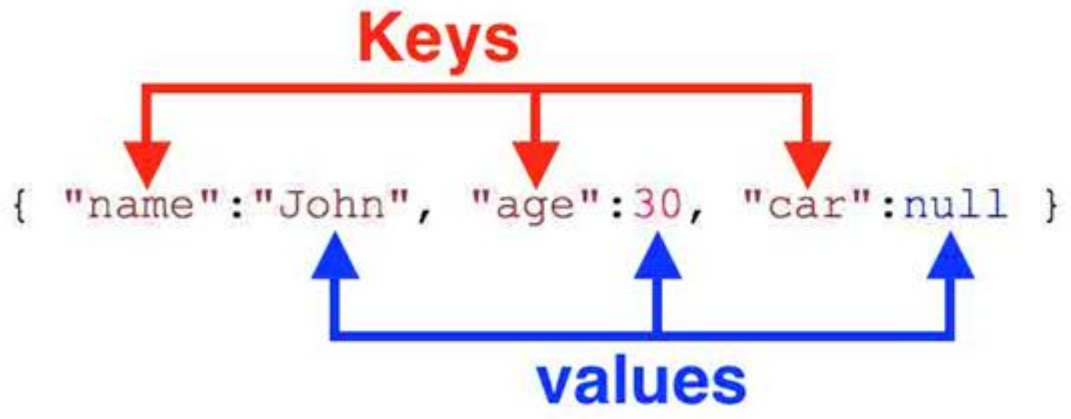


```
'{"name":"John", "age":30, "car":null}'
```

It defines an object with 3 properties:

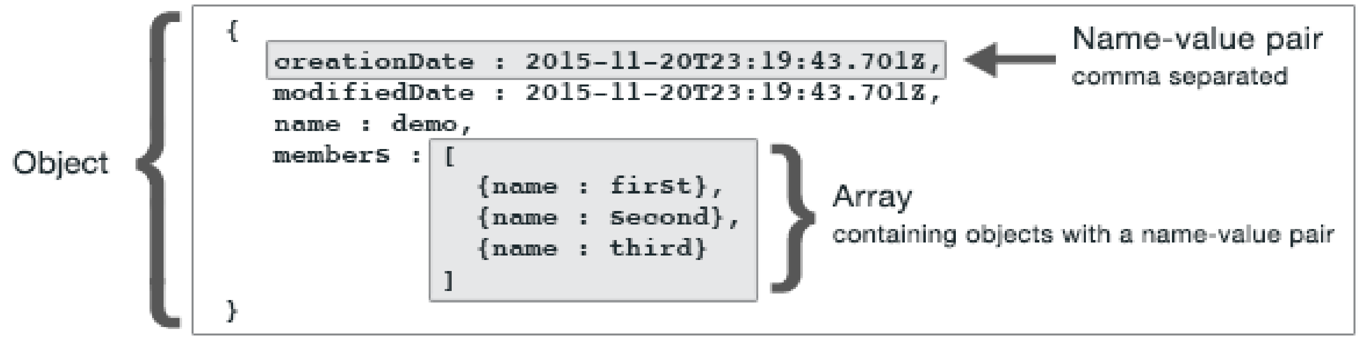
- name
- age
- car

Each property has a value.

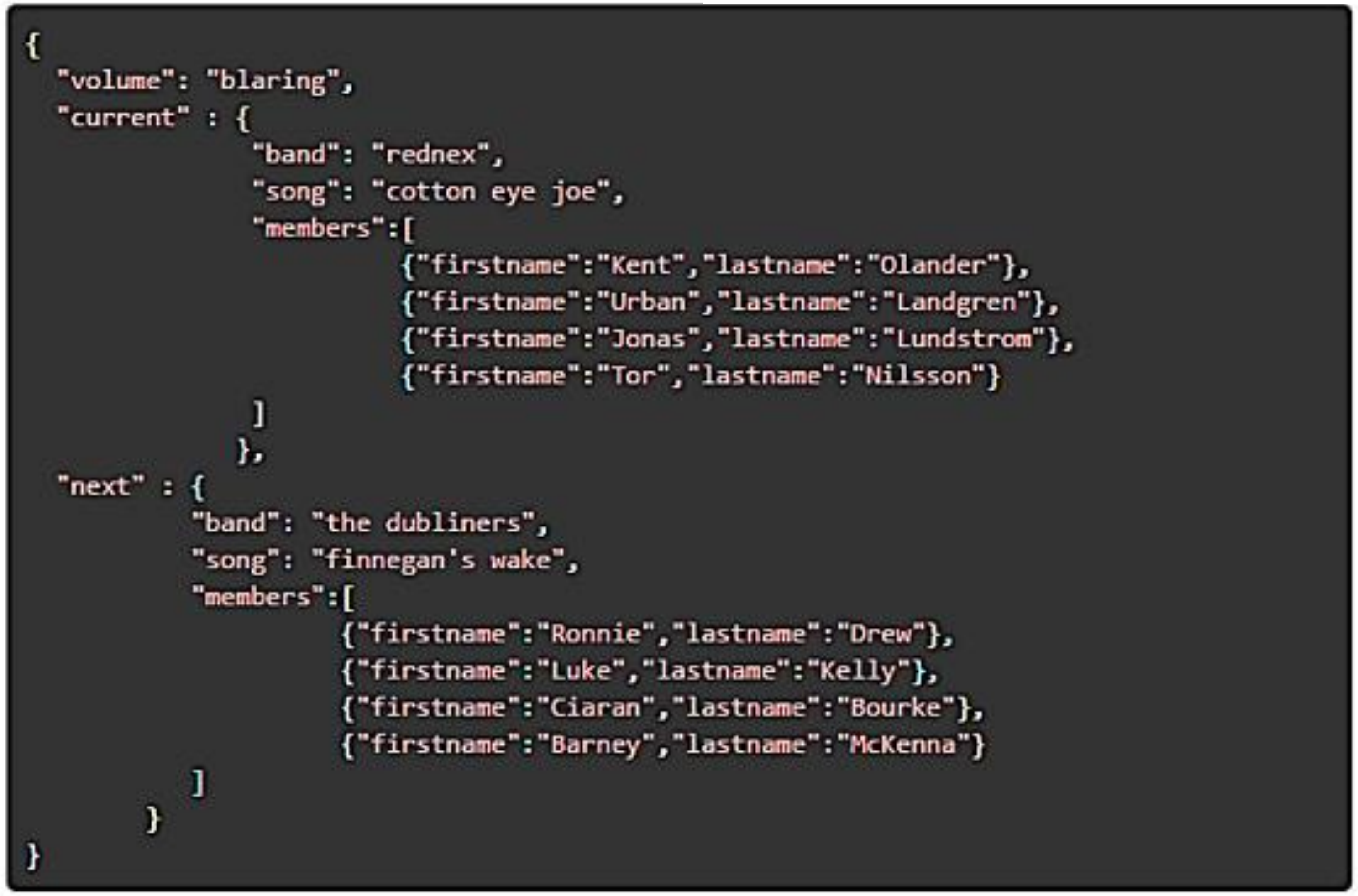


```
{ "name":"John", "age":30, "car":null }
```

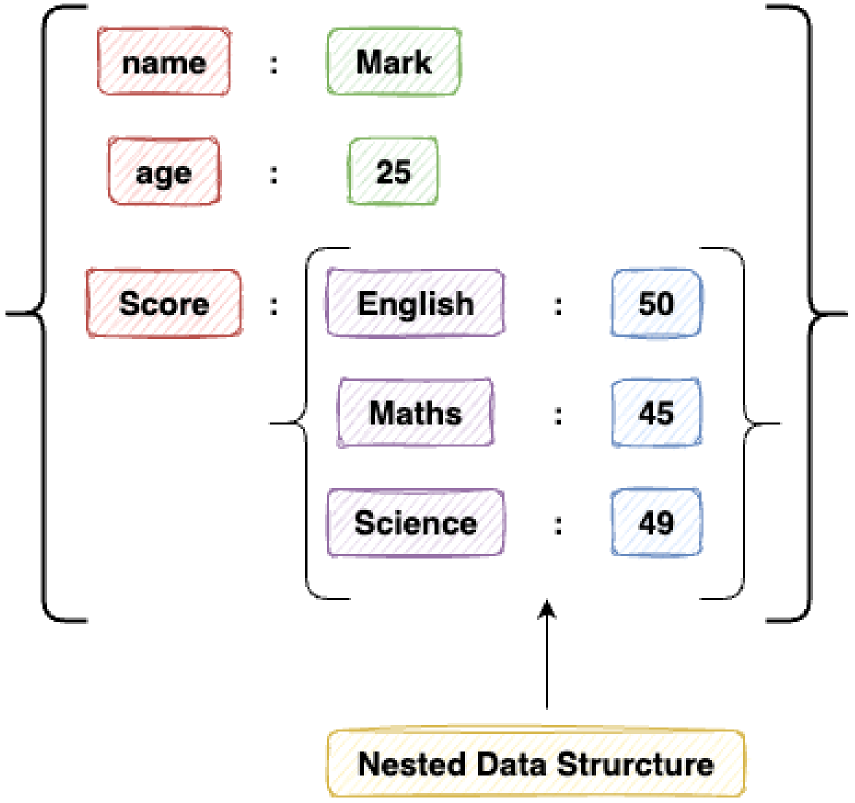
JSON: Example



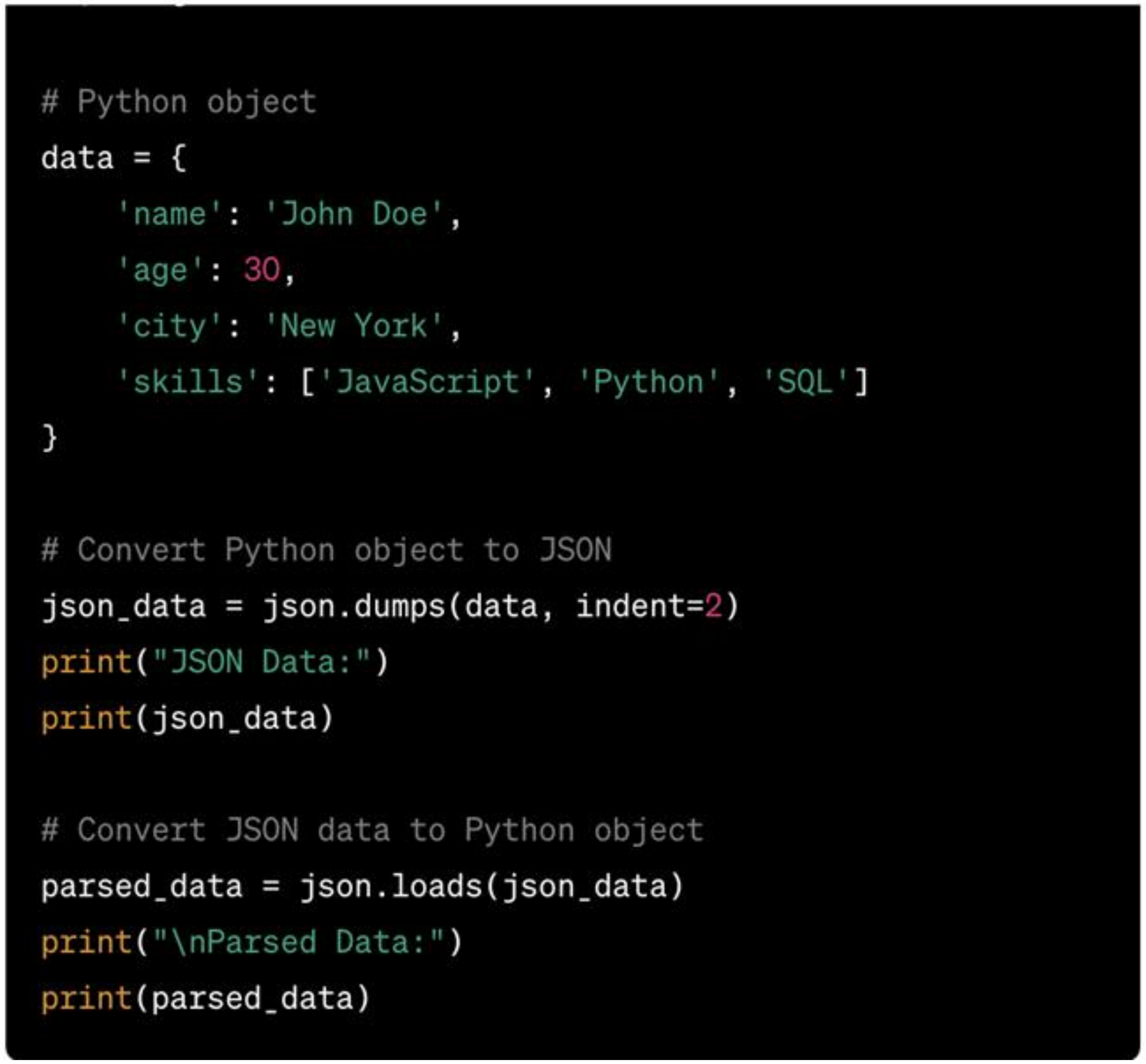
```
{
    creationDate : 2015-11-20T23:19:43.701Z,
    modifiedDate : 2015-11-20T23:19:43.701Z,
    name : demo,
    members : [
        {name : first},
        {name : second},
        {name : third}
    ]
}
```



```
{
  "volume": "blaring",
  "current" : {
             "band": "rednex",
             "song": "cotton eye joe",
             "members":[
                       {"firstname":"Kent","lastname":"Olander"},
                       {"firstname":"Urban","lastname":"Landgren"},
                       {"firstname":"Jonas","lastname":"Lundstrom"},
                       {"firstname":"Tor","lastname":"Nilsson"}
             ]
            },
  "next" : {
          "band": "the dubliners",
          "song": "finnegan's wake",
          "members":[
                    {"firstname":"Ronnie","lastname":"Drew"},
                    {"firstname":"Luke","lastname":"Kelly"},
                    {"firstname":"Ciaran","lastname":"Bourke"},
                    {"firstname":"Barney","lastname":"McKenna"}
          ]
        }
}
```



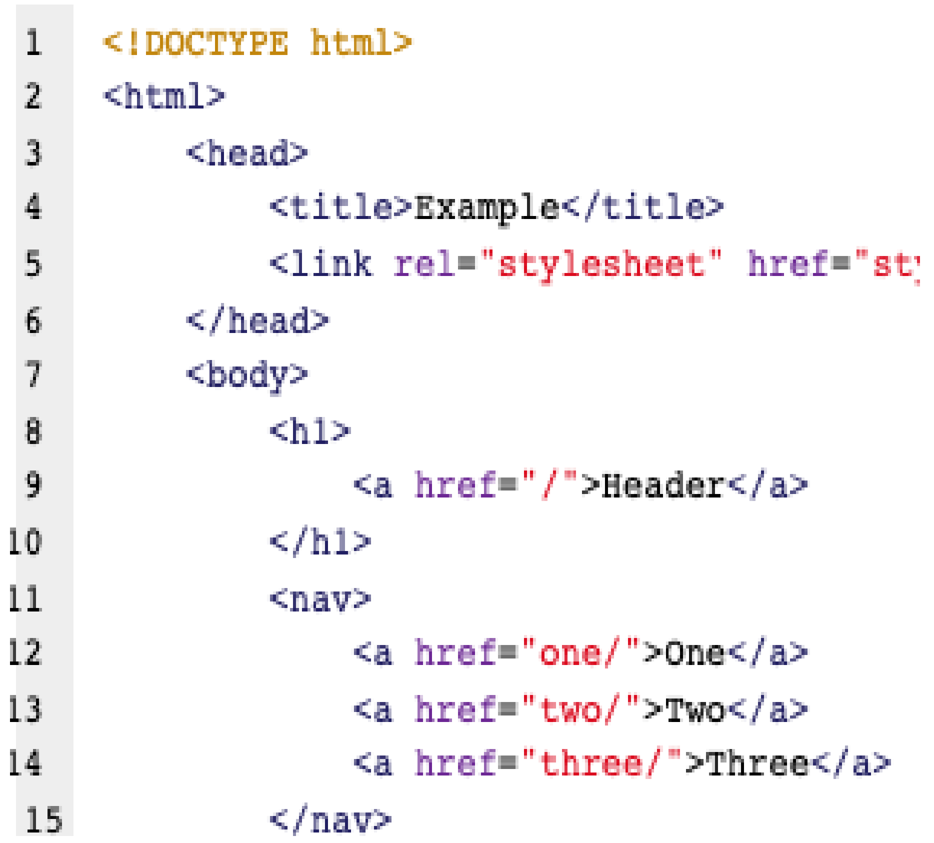
JSON in Python



```python
# Python object
data = {
    'name': 'John Doe',
    'age': 30,
    'city': 'New York',
    'skills': ['JavaScript', 'Python', 'SQL']
}

# Convert Python object to JSON
json_data = json.dumps(data, indent=2)
print("JSON Data:")
print(json_data)

# Convert JSON data to Python object
parsed_data = json.loads(json_data)
print("\nParsed Data:")
print(parsed_data)
```

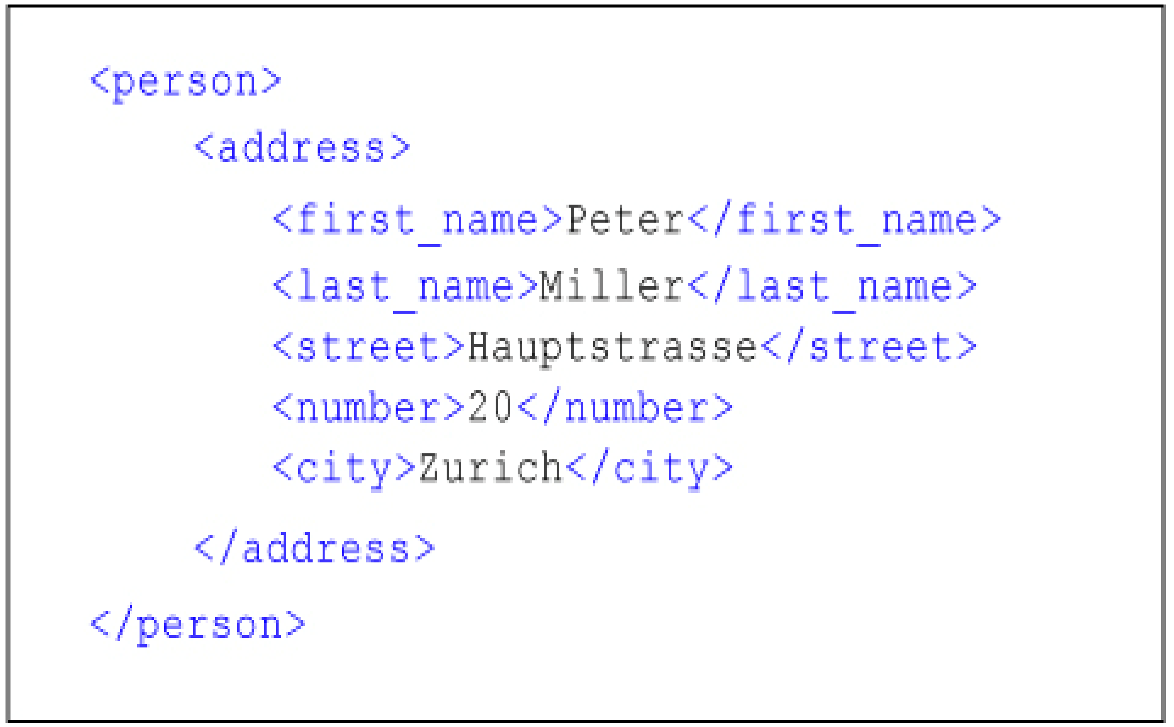
HTML and XML

HTML →Hypertext Markup Language



```
<!DOCTYPE html>
<html>
    <head>
        <title>Example</title>
        <link rel="stylesheet" href="st
    </head>
    <body>
        <h1>
            <a href="/">Header</a>
        </h1>
        <nav>
            <a href="one/">One</a>
            <a href="two/">Two</a>
            <a href="three/">Three</a>
        </nav>
```

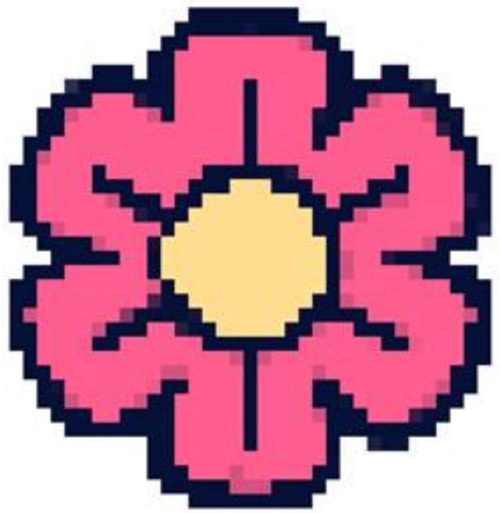
XML →eXtensible Markup Language

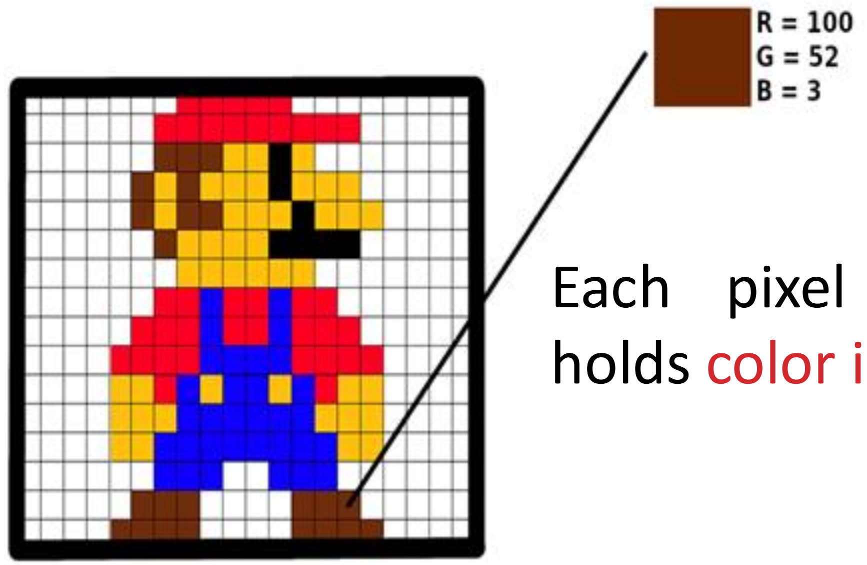


```
<person>
    <address>
        <first_name>Peter</first_name>
        <last_name>Miller</last_name>
        <street>Hauptstrasse</street>
        <number>20</number>
        <city>Zurich</city>
    </address>
</person>
```

Data Format: Images

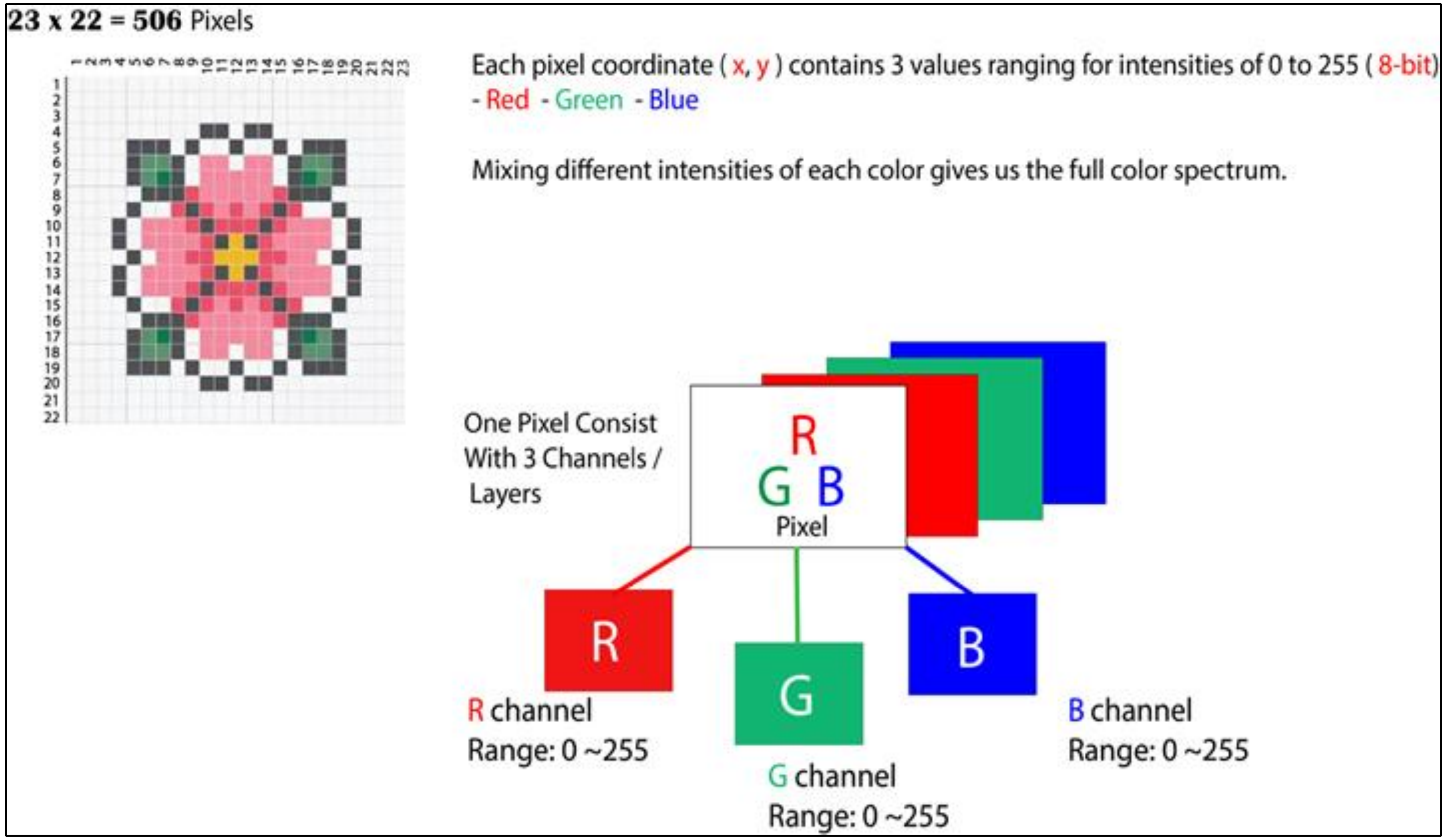
Image data encompasses **the visual content and properties of an image** (visual representation), including details such as **colors, shapes, patterns, and pixel values**.

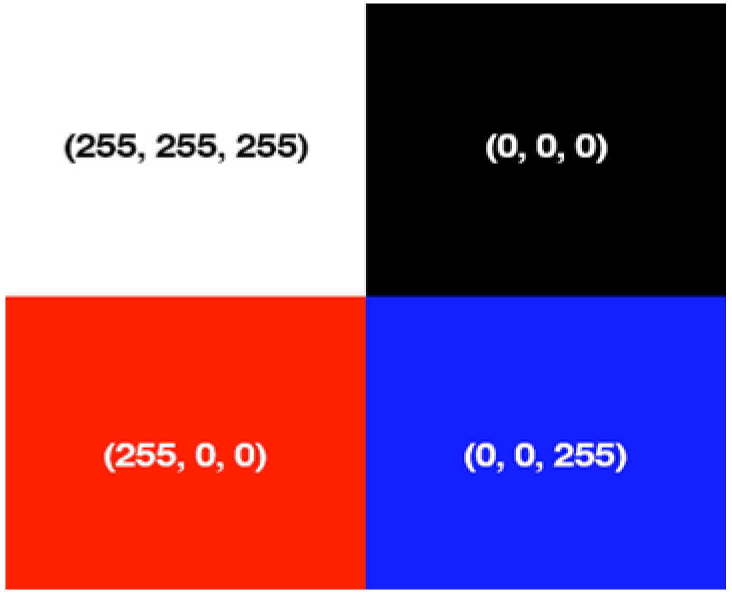




Each pixel (building block) holds color information.

Data Format: Images


23 x 22 = 506 Pixels
Each pixel coordinate (x, y) contains 3 values ranging for intensities of 0 to 255 (8-bit)
- Red - Green - Blue
Mixing different intensities of each color gives us the full color spectrum.
One Pixel Consist With 3 Channels / Layers
R G B
Pixel
R
G
B
R channel
Range: 0 ~255
G channel
Range: 0 ~255
B channel
Range: 0 ~255

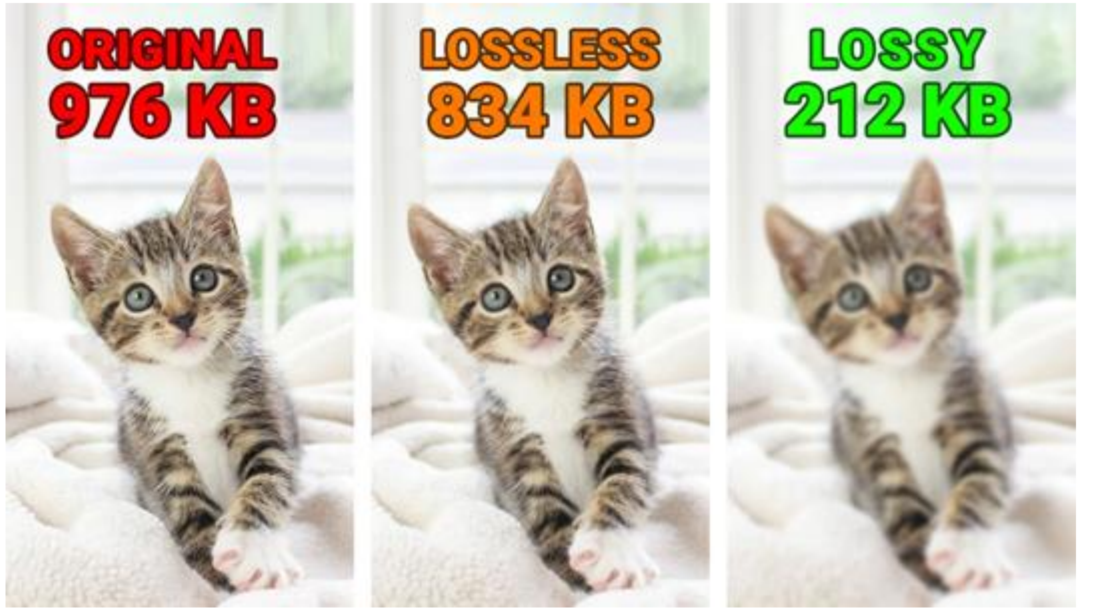


Channel Values: ranges from 0 (no intensity) to 255 (maximum intensity), creating varying shades of the color

Images Data Form and Compression

Why Compress? To save storage space.

- **Lossy Compression:** JPEG
- **Lossless Compression:** PNG, GIF



How to Get Data?

How to Get Data?

- Given to you by your company
- Gathered from databases
- From the internet (for example: web scrapping)
- From a restful API

Web Scraping: involves extracting data directly from web pages. It doesn't rely on APIs; instead, it simulates a web browser to retrieve and parse HTML content from websites.

Web scraping can be used to extract data when an API isn't available or when you need to collect information from web pages that aren't designed for programmatic access (should be done with caution, considering legal, ethical, and access restrictions).

Beautiful Soup and Parsing HTML

Beautiful Soup is a Python library for parsing HTML and XML, making web scraping (extracting data directly from web pages) and data extraction easier.

```
soup = BeautifulSoup(page.content, 'html.parser')
soup.find_all('p')
```

```
[<p>Here is some simple content for this page.</p>]
```

Note that `find_all` returns a list, so we'll have to loop through, or use list indexing, it to extract text:

```
soup.find_all('p')[0].get_text()
```

```
'Here is some simple content for this page.'
```

Notes: Don't write own parser. Install Beautiful soup and Use Beautiful Soup to parse HTML content by creating a Beautiful Soup object.

Restful APIs (Application Programming Interface)

RESTful API (Representational State Transfer API) provide **a structured and documented way** to access data from websites or services.

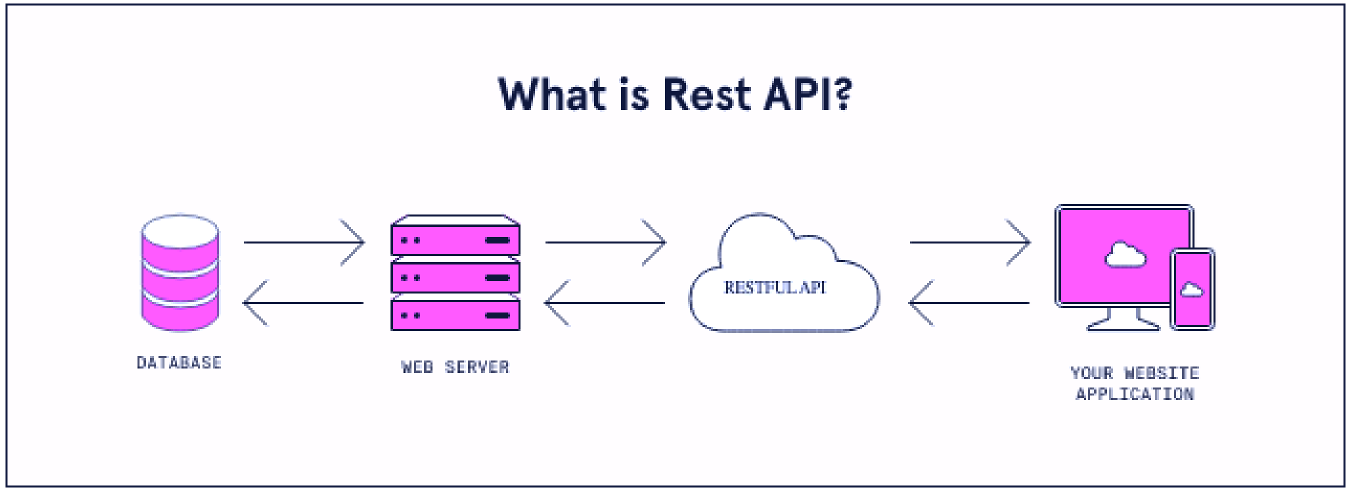
Features:

- Reliable data access.
- Establishes service agreements.
- Enables request-response communication.

API Documentation: Guides usage and data interpretation.

```
import requests
response = requests.get("http://api.open-notify.org/astros.json")
print(response)
```

"If you send me a specific request, I will return some information in a structured and documented format."



Summary

- **Data Types:** Various data types (tabular, text, graph, unstructured) impact data preparation in the data science lifecycle.
- **File Formats:** Knowledge of file formats (e.g., CSV, JSON etc.) aids data ingestion and transformation during data preparation.
- **Databases:** are structured repositories for storing and retrieving data, playing a central role in data management during the data science process.
- **Data Acquisition:** RESTful APIs and web scraping are some key for gathering data at the beginning of the data science lifecycle.

Understanding these elements is crucial for progressing through the data science lifecycle, leading to data-driven insights and solutions.

Some More Examples of Different Types of Data

- **Tabular Data**
  - Text document of the heights of everyone in this class in inches
  - IRS Data for taxpayers
  - Netflix show data
- **Graph Data**
  - Social networks
  - Course prereqs
  - Highway
- **Geo Data**
  - Flight paths
  - Weather patterns
  - All phones on verizon

Some More Examples cont.

- **Raw Data**
  - Image
  - Video
  - Audio
  - Telemetry
- **Hierarchies (Graphy)**
  - Taxonomy for something
  - Family tree
  - File directory
- **Text**
  - All tweets
  - Chat logs
  - Search history
  - Shakespeare
- **Time Series**
  - People in store
  - Stock prices

Additional Reading Slides

JSON Files & Strings

- **Easy for humans to read (and sanity check, edit)**
- **Example:** The JSON object represents a person's information:

```
{
"name": "John",
"age": 25,
"city": "New York"
}
```

$\rightarrow$ **“Key”:“value” pair**

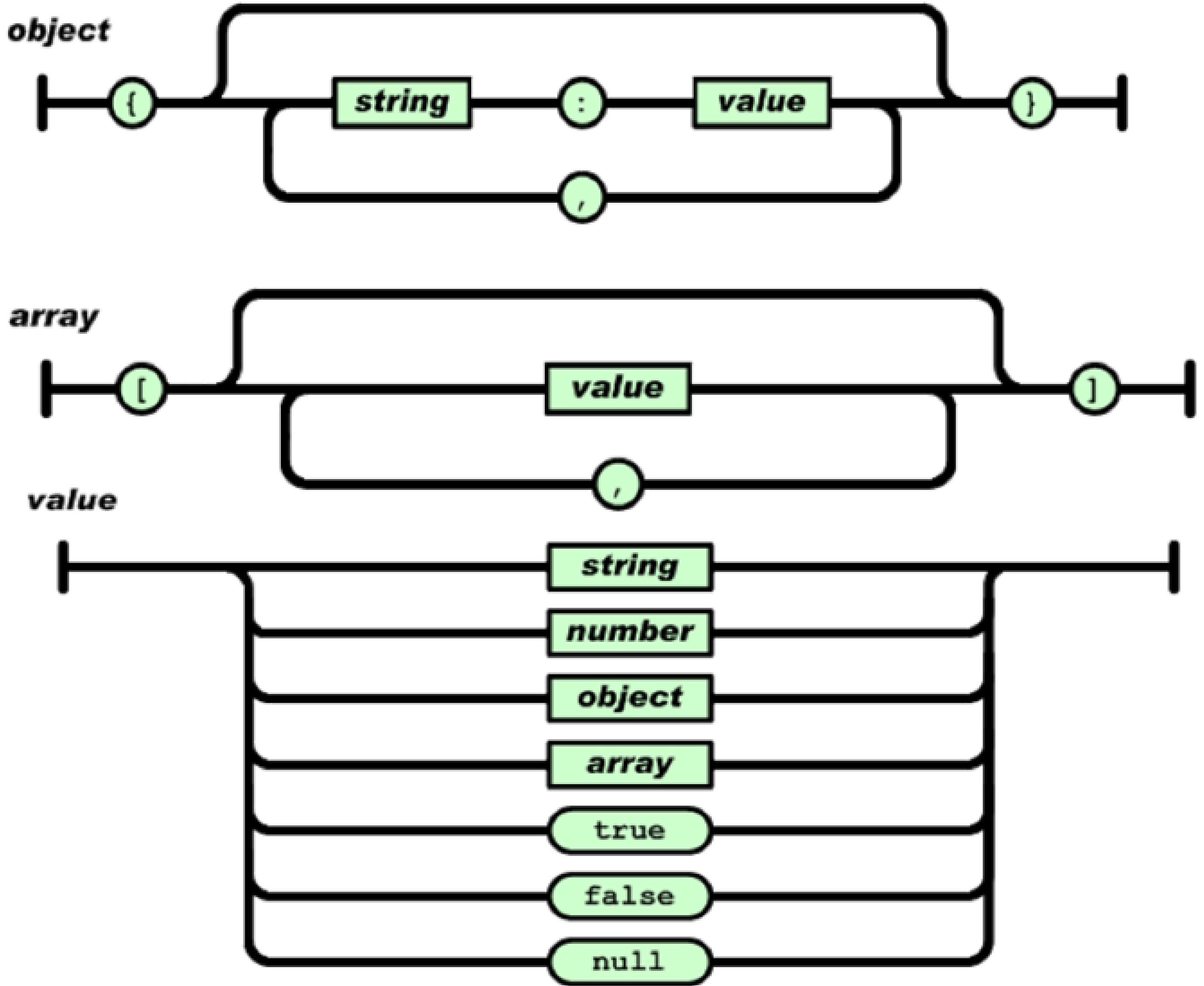
Popular for data interchange and storage due to its simplicity and broad compatibility: Serialization & Deserialization

- **JSON is a method for serializing objects:**
  - Convert an object into a string
  - Deserialization converts a string back to an object

JSON Files & Strings

Defined by three universal data structures

Valid JSON data type for "value": objects, arrays, strings, numbers, booleans, and null



Python dictionary, Java Map, hash table, etc …

```
{
"name": "John",
"age": 25,
}
```

Python list, Java array, vector, etc …

["apple", "banana", "orange"]

Python string, float, int, boolean, JSON object, JSON array, …

"Hello, world!"

Images from: http://www.json.org/

JSON In Python

Some built-in types: “Strings”, 1.0, True, False, None

Lists: [“Goodbye”, “Cruel”, “World”]

Dictionaries: {“hello”: “bonjour”, “goodbye”: “au revoir”}

Dictionaries within lists within dictionaries within lists (Nested Structures):

```
[1, 2, {“Help”:[
                “I’m”, {“trapped”: “in”},
                “CMSC320”
                ]}]
```