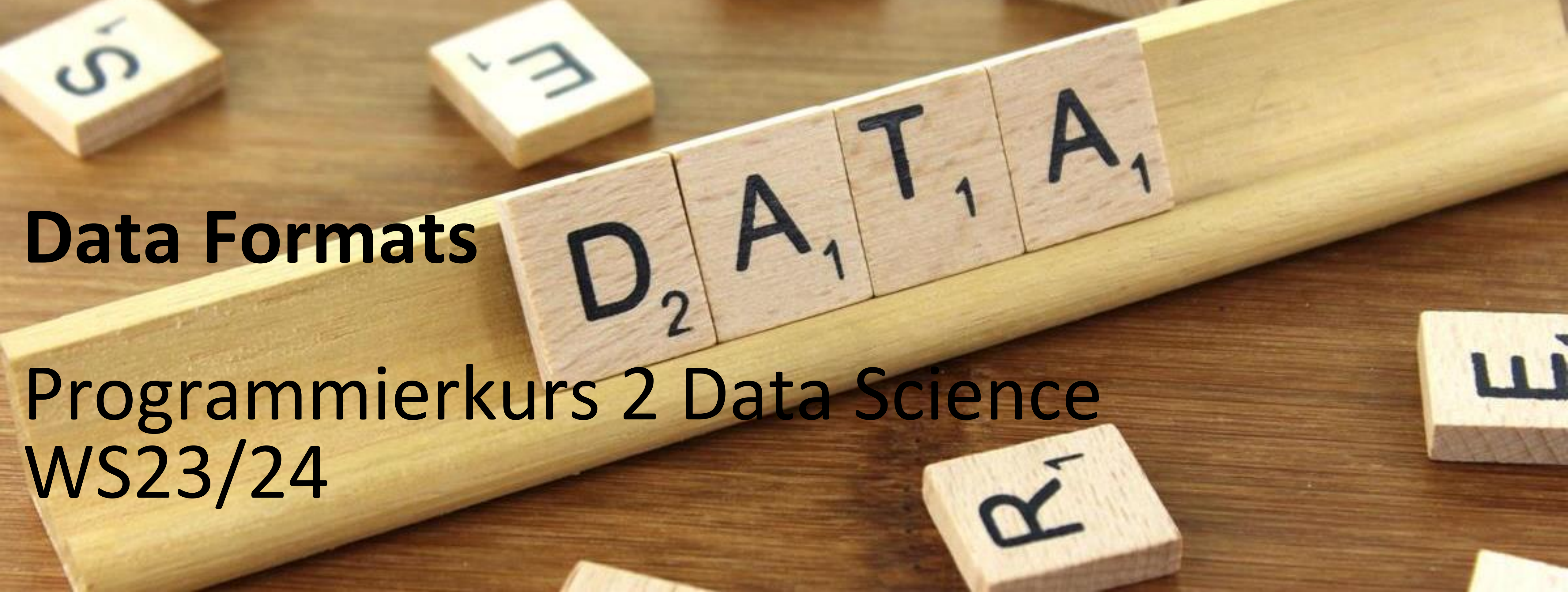


Data Formats

Programmierkurs 2 Data Science
WS23/24

Leonard Traeger
M. Sc. Information Systems
leonard.traeger@fh-dortmund.de

Disclaimer

This lecture part is designed to give you a rough understanding of data in the wild.

If your group aims to gather data via Web-Scraping – this is a good start.

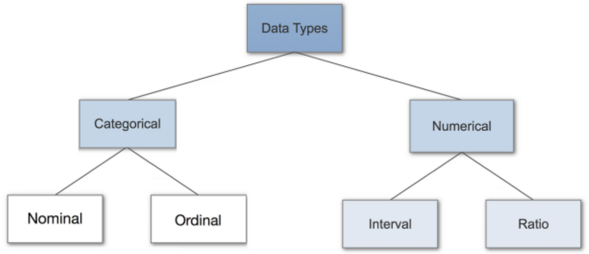
- This lesson is partially based on the Library Carpentry https://librarycarpentry.org/lc-spreadsheets/
- Slides by Philipp Schaer, Technische Hochschule Köln, Cologne, Germany

Learning Goals Data Formats

- **Give** a definition on data.
- **Explain** existing formats to structure data and their core purpose. **Name** advantages and disadvantages.
- **Identify** data formats given CSV, JSON, and XML samples.
- **Describe** XPath and how it can be used in Web-Scraping.

What is data?

DATA...
DATA...EVERYWHERE...
makeameme.org


Data Types
Categorical
Numerical
Nominal
Ordinal
Interval
Ratio

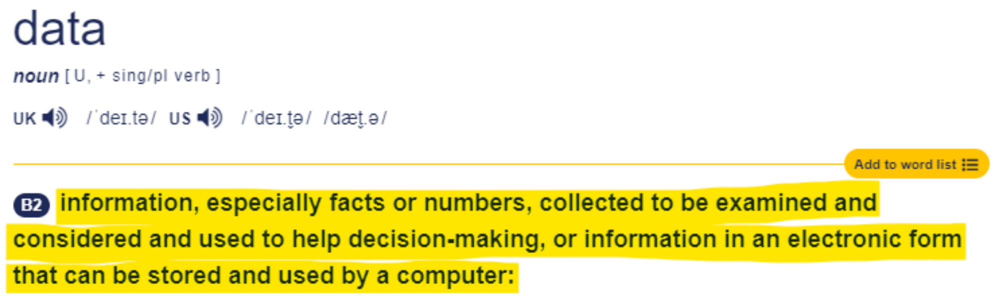
Data

Even though we can consider a multitude of things as “data“: digits, sound, images, text,...

- Common standards exist to store these and have proven themselves in domains.

As data scientists we should be able to

- **Transform raw data** into a machine readable and optimized representation format.
- Recommend **different data format standards**.
- *Map different data formats: because our analytical application should abstract any standard.*



data

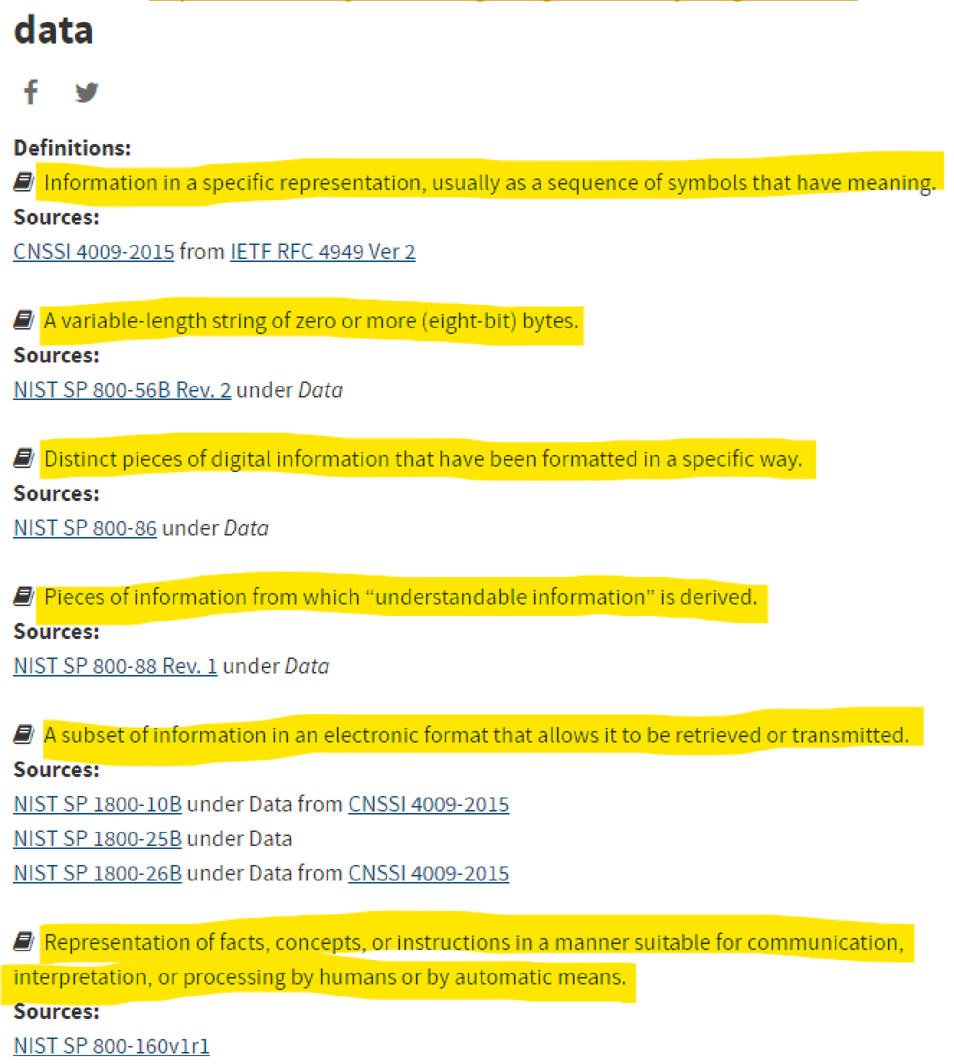
noun [U, + sing/pl verb]

UK /ˈdeɪ.tə/ US /ˈdeɪ.t̬ə/ /ˈdæt̬.ə/

Add to word list

B2 information, especially facts or numbers, collected to be examined and considered and used to help decision-making, or information in an electronic form that can be stored and used by a computer:

https://dictionary.cambridge.org/dictionary/english/data



data

Definitions:

Information in a specific representation, usually as a sequence of symbols that have meaning.
Sources:
CNSSI 4009-2015 from IETF RFC 4949 Ver 2

A variable-length string of zero or more (eight-bit) bytes.
Sources:
NIST SP 800-56B Rev. 2 under *Data*

Distinct pieces of digital information that have been formatted in a specific way.
Sources:
NIST SP 800-86 under *Data*

Pieces of information from which “understandable information” is derived.
Sources:
NIST SP 800-88 Rev. 1 under *Data*

A subset of information in an electronic format that allows it to be retrieved or transmitted.
Sources:
NIST SP 1800-10B under Data from CNSSI 4009-2015
NIST SP 1800-25B under Data
NIST SP 1800-26B under Data from CNSSI 4009-2015

Representation of facts, concepts, or instructions in a manner suitable for communication, interpretation, or processing by humans or by automatic means.
Sources:
NIST SP 800-160v1r1

Data Formats (cont.)

Binary

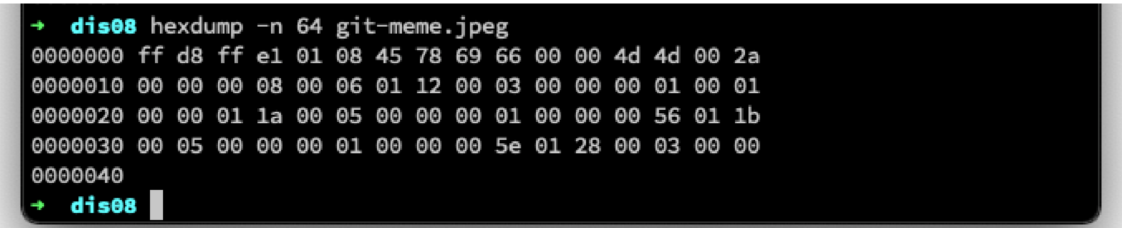
- Not human readable
- Memory efficient and fast to parse
- Platform-dependent (negative aspect)
- Difficult format conversion (e.g., open a Word Document in Open Office…)

Text

- Human readable (mostly)
- Waste more memory and relatively slow to parse
- Platform-independent (positive aspect, but still some encoding problems)
- Easy format conversion.

Binary Data

- Binary files are usually thought of as being a **sequence of bytes**, which means the binary digits (bits) are grouped in eights.
- Binary **files** typically **contain bytes** that are **intended** to be interpreted as something **other than text characters**.
- Some binary files contain **headers to interpret** the data in the file. The header often contains a **signature or magic number** which can identify the format.
- JPEG magic numbers: ff d8 ff e0 or ff d8 ff e1



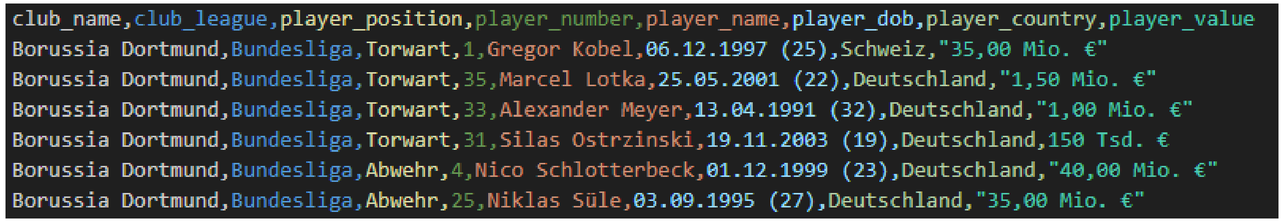
```
➜  dis08 hexdump -n 64 git-meme.jpeg
0000000 ff d8 ff e1 01 08 45 78 69 66 00 00 4d 4d 00 2a
0000010 00 00 00 08 00 06 01 12 00 03 00 00 00 01 00 01
0000020 00 00 01 1a 00 05 00 00 00 01 00 00 00 56 01 1b
0000030 00 05 00 00 00 01 00 00 00 5e 01 28 00 03 00 00
0000040
➜  dis08 
```

Technology
Arts Sciences
TH Köln

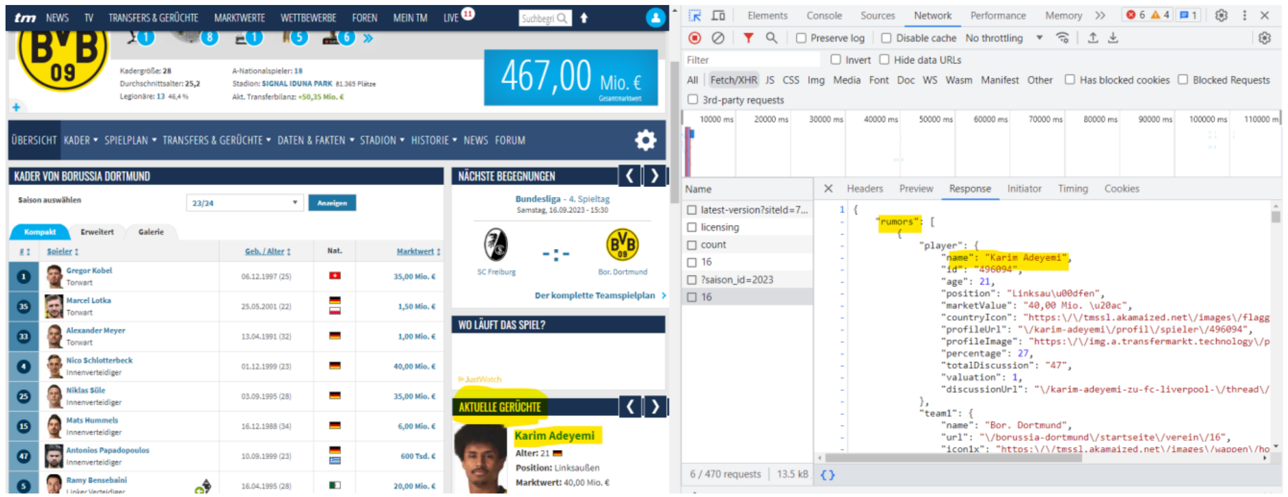
Text Formats

1. **CSV:** strings separated by commas and newlines.
   → Simple and most common in modelling relational data.
2. **JSON**: uses javascript syntax.
   → Tree-based and most common in data exchange.
3. **XML**: is a mark-up language (meta) and builds XHTML (language of the web).
   → Tree-based and most common in (web) structured documents.

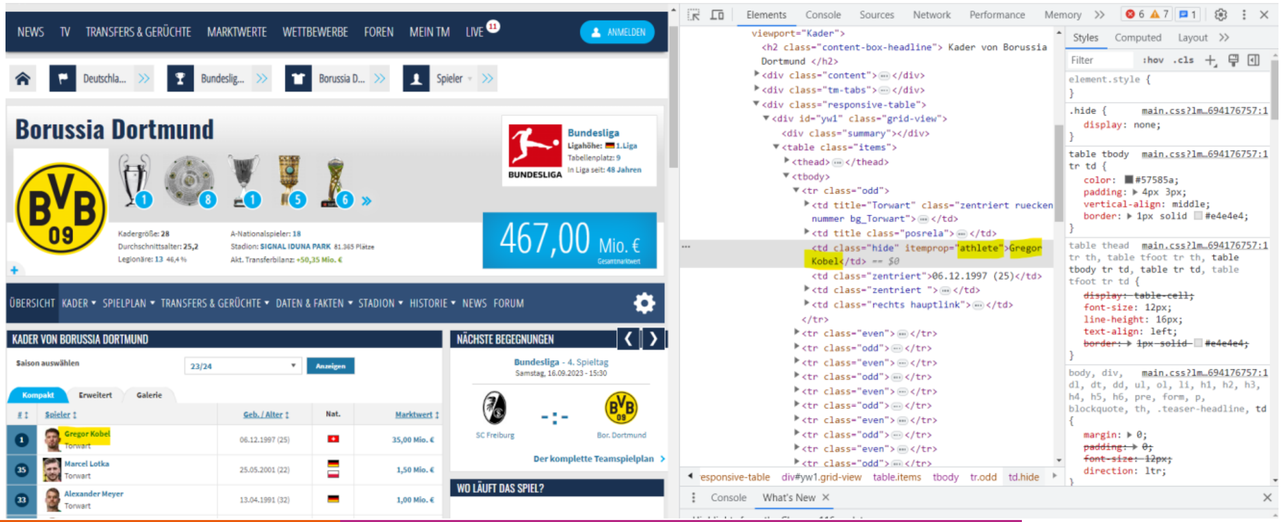
- **RDF**: WWW model to exchange metadata and linkages within graphs
- **OWL**: ontologies in semantic web



```
club_name,club_league,player_position,player_number,player_name,player_dob,player_country,player_value
Borussia Dortmund,Bundesliga,Torwart,1,Gregor Kobel,06.12.1997 (25),Schweiz,"35,00 Mio. €"
Borussia Dortmund,Bundesliga,Torwart,35,Marcel Lotka,25.05.2001 (22),Deutschland,"1,50 Mio. €"
Borussia Dortmund,Bundesliga,Torwart,33,Alexander Meyer,13.04.1991 (32),Deutschland,"1,00 Mio. €"
Borussia Dortmund,Bundesliga,Torwart,31,Silas Ostrzinski,19.11.2003 (19),Deutschland,150 Tsd. €
Borussia Dortmund,Bundesliga,Abwehr,4,Nico Schlotterbeck,01.12.1999 (23),Deutschland,"40,00 Mio. €"
Borussia Dortmund,Bundesliga,Abwehr,25,Niklas Süle,03.09.1995 (27),Deutschland,"35,00 Mio. €"
```



```
{
    "rumors": [
        {
            "player": {
                "name": "Karim Adeyemi",
                "id": "496094",
                "age": 21,
                "position": "Linksau\u00dfen",
                "marketValue": "40,00 Mio. \u20ac",
                "countryIcon": "https:\/\/tmssl.akamaized.net\/images\/flagg
                "profileUrl": "\/karim-adeyemi\/profil\/spieler\/496094",
                "profileImage": "https:\/\/img.a.transfermarkt.technology\/p
                "percentage": 27,
                "totalDiscussion": "47",
                "valuation": 1,
                "discussionUrl": "\/karim-adeyemi-zu-fc-liverpool-\/thread\/
            },
            "team1": {
                "name": "Bor. Dortmund",
                "url": "\/borussia-dortmund\/startseite\/verein\/16",
                "icon1x": "https:\/\/tmssl.akamaized.net\/images\/wappen\/ho
```

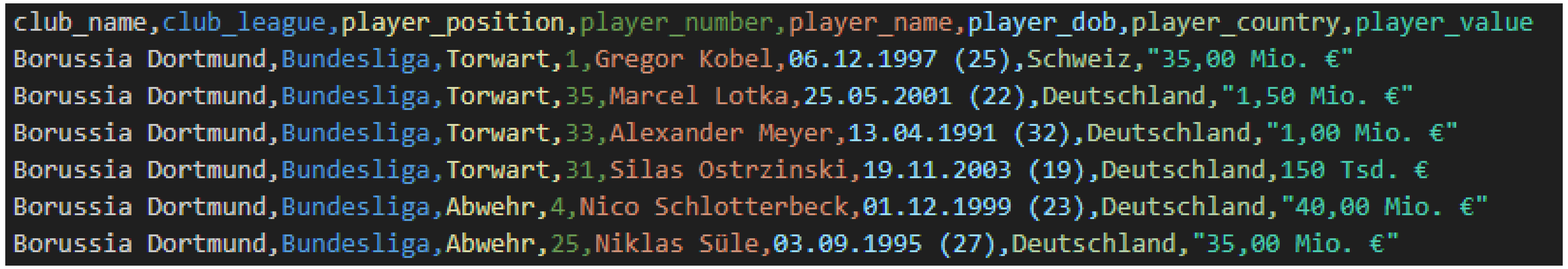


```
viewport="Kader">
  <h2 class="content-box-headline"> Kader von Borussia Dortmund </h2>
  <div class="content">…</div>
  <div class="tm-tabs">…</div>
  <div class="responsive-table">
    <div id="yw1" class="grid-view">
      <div class="summary"></div>
      <table class="items">
        <thead>…</thead>
        <tbody>
          <tr class="odd">
            <td title="Torwart" class="zentriert rueckennummer bg_Torwart">…</td>
            <td title class="posrela">…</td>
            <td class="hide" itemprop="athlete">Gregor Kobel</td> == $0
            <td class="zentriert">06.12.1997 (25)</td>
            <td class="zentriert ">…</td>
            <td class="rechts hauptlink">…</td>
          </tr>
          <tr class="even">…</tr>
          <tr class="odd">…</tr>
          <tr class="even">…</tr>
          <tr class="odd">…</tr>
          <tr class="even">…</tr>
          <tr class="odd">…</tr>
          <tr class="even">…</tr>
          <tr class="odd">…</tr>
          <tr class="even">…</tr>
          <tr class="odd">…</tr>
```

Technology Arts Sciences TH Köln

In pandas library:
pandas.read_csv(file.csv)

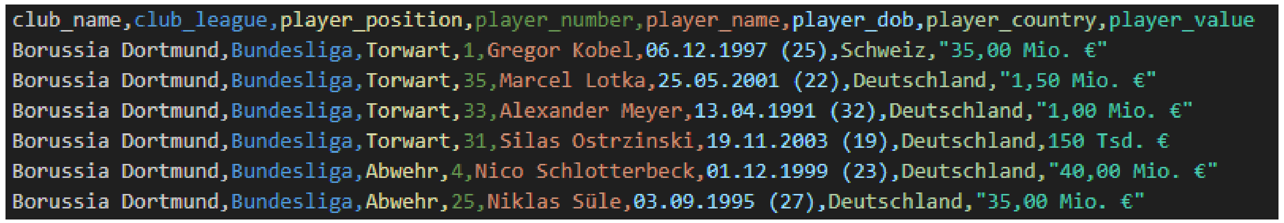
CSV: Comma seperated values



```
club_name,club_league,player_position,player_number,player_name,player_dob,player_country,player_value
Borussia Dortmund,Bundesliga,Torwart,1,Gregor Kobel,06.12.1997 (25),Schweiz,"35,00 Mio. €"
Borussia Dortmund,Bundesliga,Torwart,35,Marcel Lotka,25.05.2001 (22),Deutschland,"1,50 Mio. €"
Borussia Dortmund,Bundesliga,Torwart,33,Alexander Meyer,13.04.1991 (32),Deutschland,"1,00 Mio. €"
Borussia Dortmund,Bundesliga,Torwart,31,Silas Ostrzinski,19.11.2003 (19),Deutschland,150 Tsd. €
Borussia Dortmund,Bundesliga,Abwehr,4,Nico Schlotterbeck,01.12.1999 (23),Deutschland,"40,00 Mio. €"
Borussia Dortmund,Bundesliga,Abwehr,25,Niklas Süle,03.09.1995 (27),Deutschland,"35,00 Mio. €"
```

- First row usually stores object / attribute / column description.
- Column separator:
  - Default with comma (“,” → CSV)
  - Alternatively: tabulator (“\t” → TSV)
- Row separator: usually newline (“\n”)
- Strings can be enclosed in quotation marks to escape special characters .
- Quotation marks are escaped by another quotation marks.

Technology
Arts Sciences
TH Köln



```
club_name,club_league,player_position,player_number,player_name,player_dob,player_country,player_value
Borussia Dortmund,Bundesliga,Torwart,1,Gregor Kobel,06.12.1997 (25),Schweiz,"35,00 Mio. €"
Borussia Dortmund,Bundesliga,Torwart,35,Marcel Lotka,25.05.2001 (22),Deutschland,"1,50 Mio. €"
Borussia Dortmund,Bundesliga,Torwart,33,Alexander Meyer,13.04.1991 (32),Deutschland,"1,00 Mio. €"
Borussia Dortmund,Bundesliga,Torwart,31,Silas Ostrzinski,19.11.2003 (19),Deutschland,150 Tsd. €
Borussia Dortmund,Bundesliga,Abwehr,4,Nico Schlotterbeck,01.12.1999 (23),Deutschland,"40,00 Mio. €"
Borussia Dortmund,Bundesliga,Abwehr,25,Niklas Süle,03.09.1995 (27),Deutschland,"35,00 Mio. €"
```

CSV (cont.)

- The **advantage** of CSV files is **simplicity**… why?
- CSV files are **widely supported** by:
  - many types of programs and programming languages.
  - can be viewed and imported in text editors; Excel; databases; and Python ☺
  - are a straightforward way to represent data.
- Whenever your data has **no nested structure → USE CSV!**
- **Comma is default**, but tabs are often better as you rarely have to escape strings in tab separated files.

In pandas library:
pandas.read_json(file.json)

JSON: JavaScript Object Notation

```
[
        { "club_name": "Borussia Dortmund",
          "club_league ": "Bundesliga",
          "player": ["Gregor Kobel", "Marcel Lotka", "Alexander Meyer"]
        }
]
```

- **JavaScript notation**. **Human readable** but not easy to parse.
- Popular use: Client and **web** server often use JSON to communicate and **exchange data**.
- **JavaScript lists and arrays** are naturally represented with JSON.
- **Almost compatible** to **Python's syntax** of datatypes (booleans, numbers, strings) and containers (arrays, lists, tuples, dictionaries), and object-oriented programming.

JSON (cont.)

JSON is built on two structures:

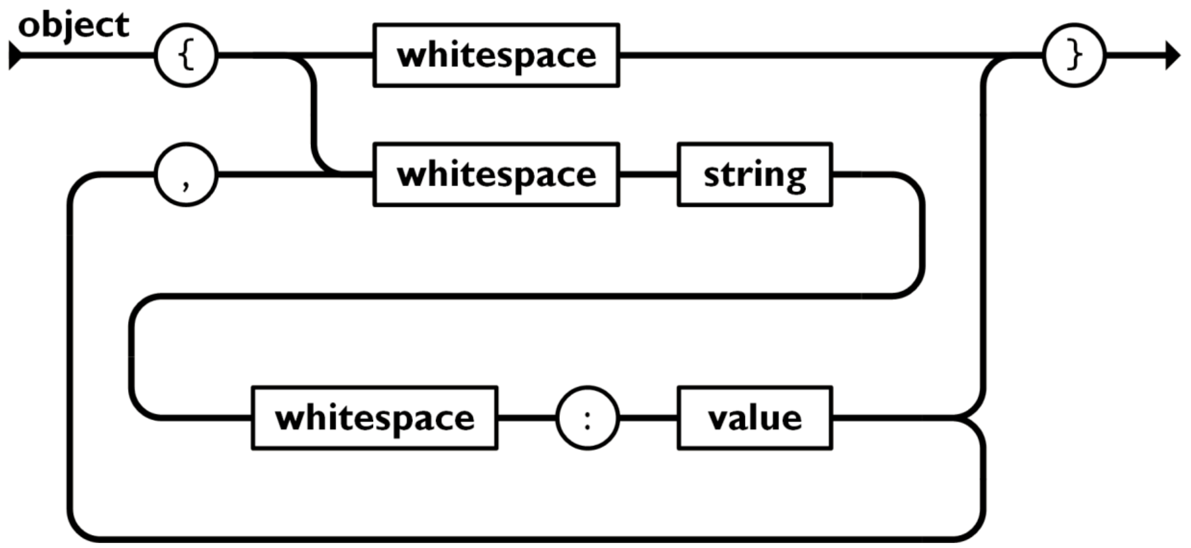
1. A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
2. An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

All details can be found online: https://www.json.org

JSON Object #1

An object is **an unordered set of name/value pairs**.

- An object begins with **{** (left curly parenthesis) and ends with } (right curly parenthesis).
- Each name is followed by **:** (colon) and its value
- Pairs are separated by **,** (comma).

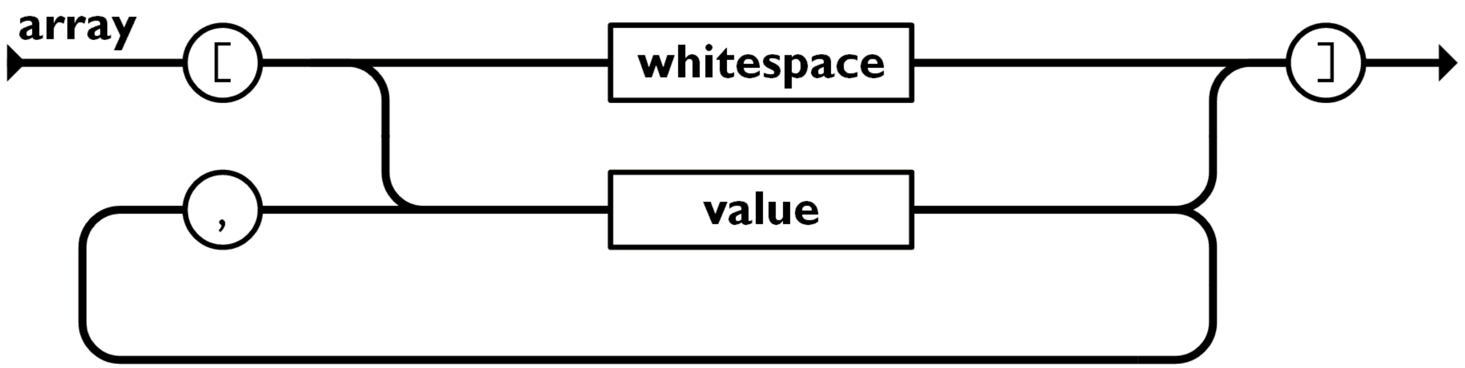


Technology
Arts Sciences
TH Köln

JSON Arrays #2

An array is an ordered collection of values.

- Begins with **[** (left bracket) and ends with **]** (right bracket).
- Values are separated by **,** (comma).



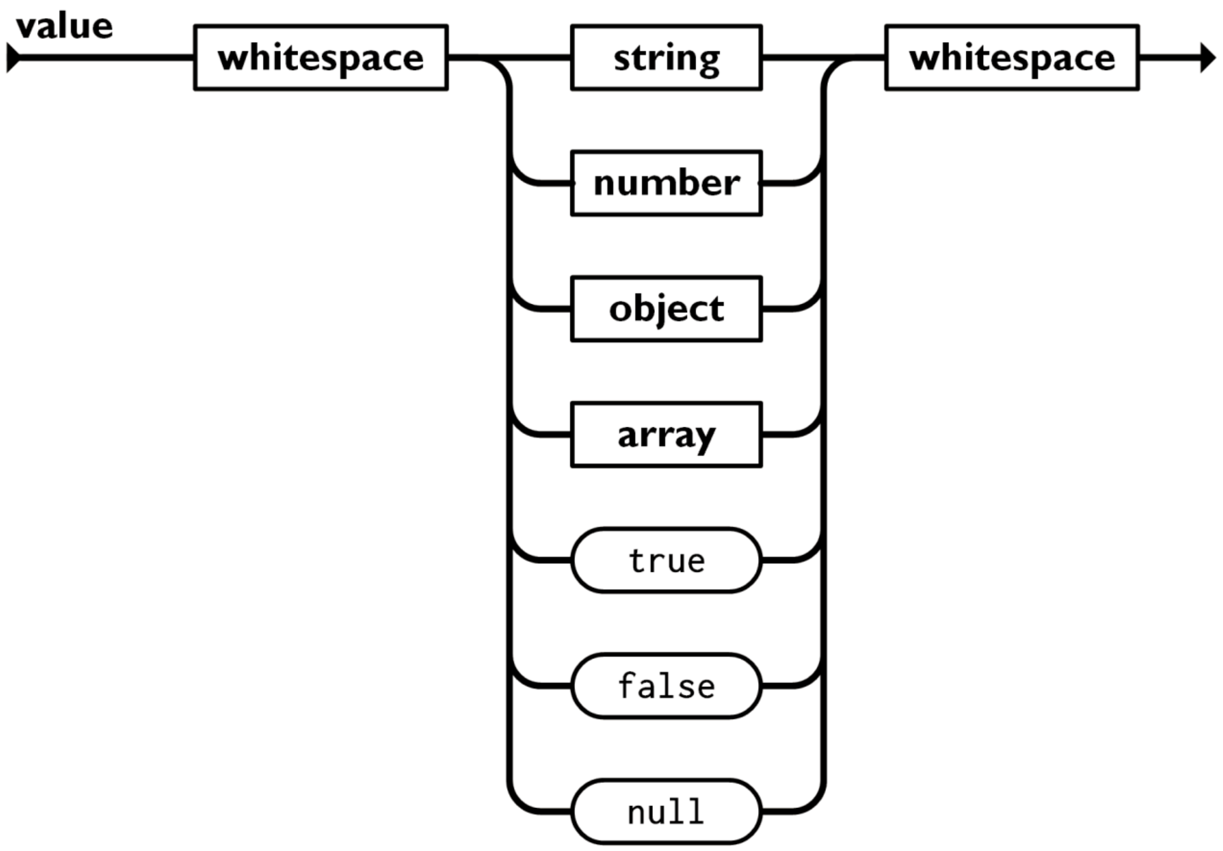
JSON Values

A value can be a

- **String** in double quotes
- **Number**
- **True** or **false** or **null**
- **Object**
- **Array**.

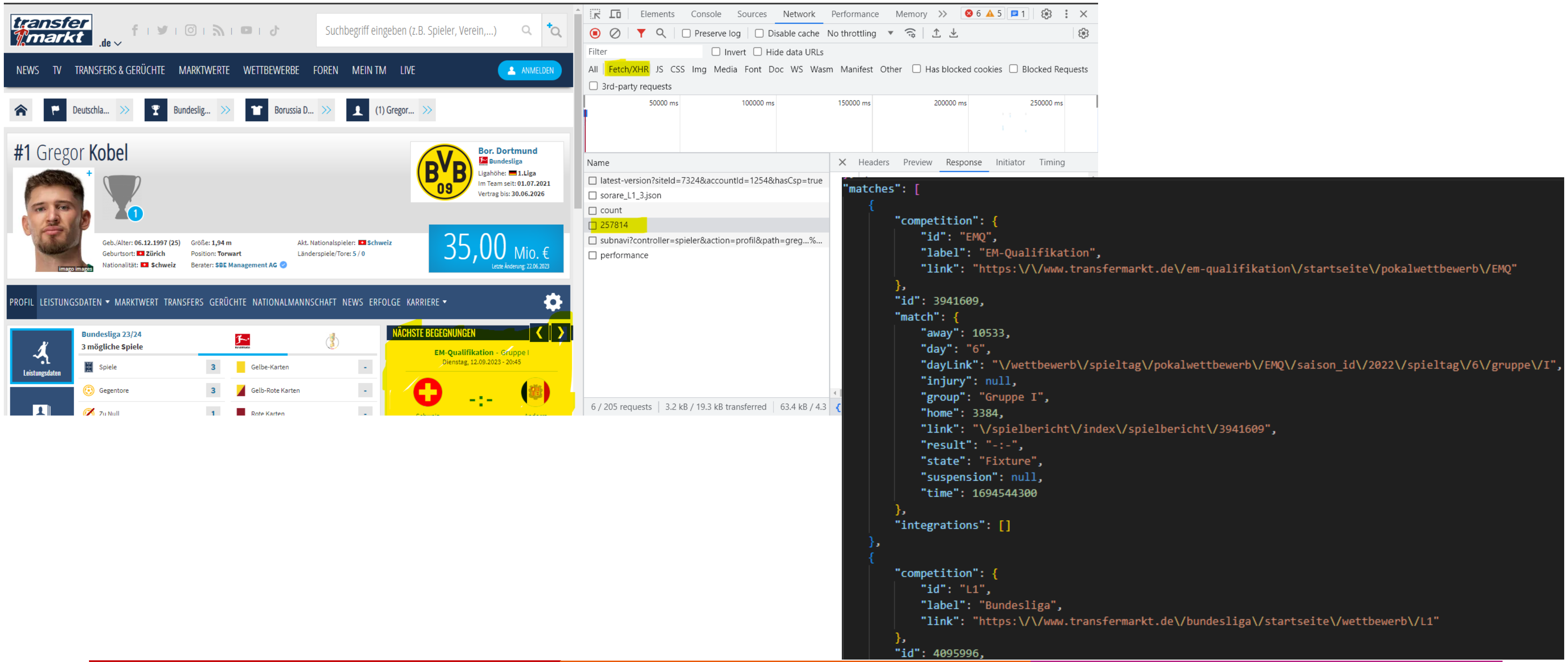
These structures can be nested.

And so on…



Technology Arts Sciences TH Köln

JSON: Example



```
"matches": [
    {
        "competition": {
            "id": "EMQ",
            "label": "EM-Qualifikation",
            "link": "https:\/\/www.transfermarkt.de\/em-qualifikation\/startseite\/pokalwettbewerb\/EMQ"
        },
        "id": 3941609,
        "match": {
            "away": 10533,
            "day": "6",
            "dayLink": "\/wettbewerb\/spieltag\/pokalwettbewerb\/EMQ\/saison_id\/2022\/spieltag\/6\/gruppe\/I",
            "injury": null,
            "group": "Gruppe I",
            "home": 3384,
            "link": "\/spielbericht\/index\/spielbericht\/3941609",
            "result": "-:-",
            "state": "Fixture",
            "suspension": null,
            "time": 1694544300
        },
        "integrations": []
    },
    {
        "competition": {
            "id": "L1",
            "label": "Bundesliga",
            "link": "https:\/\/www.transfermarkt.de\/bundesliga\/startseite\/wettbewerb\/L1"
        },
        "id": 4095996,
```

Technology Arts Sciences TH Köln

Do we need anything more? **XML**

"…defines a **syntax** to provide **structured data** sets of any kind with simple, **understandable markups**, which can be **evaluated by applications** of various kinds."

What is only indirectly stated here:

- XML is the **eXtensible Markup Language**.
- XML is an international **W3C standard**.
- XML documents are **human and machine readable**!
- XML is a **document description** language.
- XML **separates structure** from **presentation**, or content from presentation.
- XML documents are developed according to a **document model**.

Technology Arts Sciences TH Köln

In pandas library:
pandas.read_xml(file.xml)

XML (cont.)

An XML document may contain:

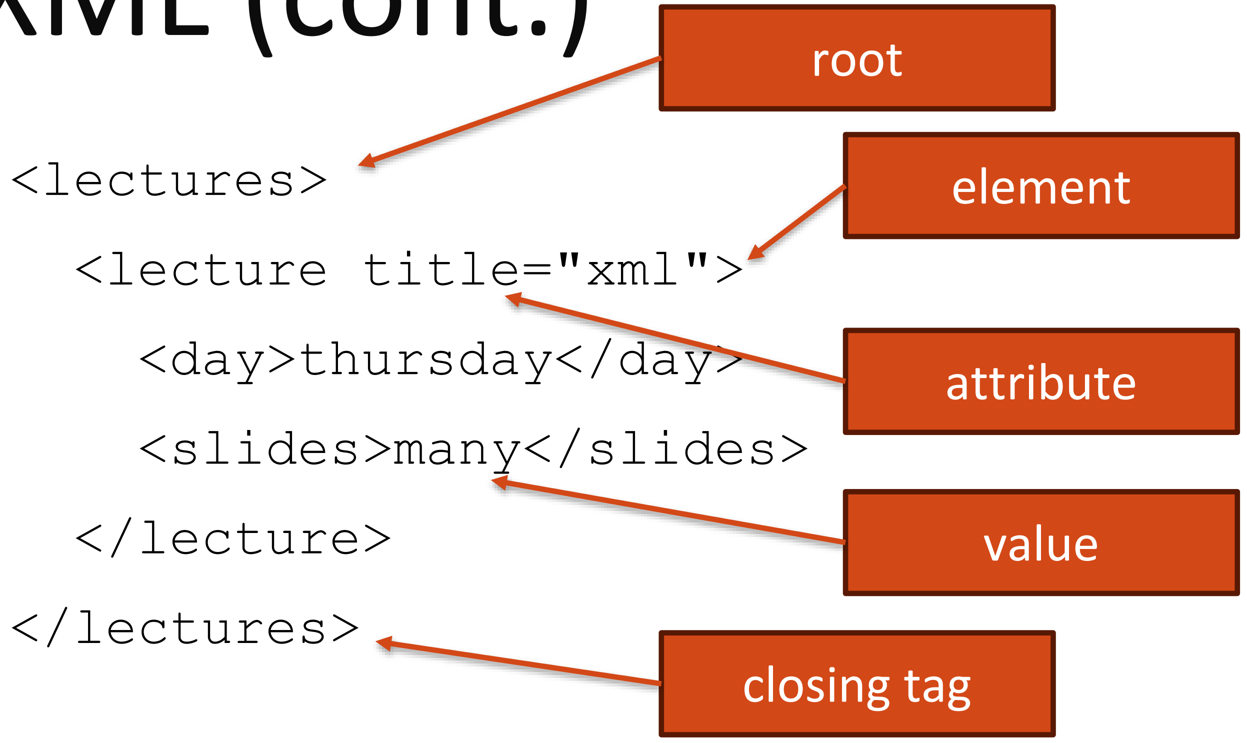
- **Elements**, possibly with attributes
- **Processing instructions**
- **Comments**
- **Entity references**

An XML document must be well-formed and can be validated.

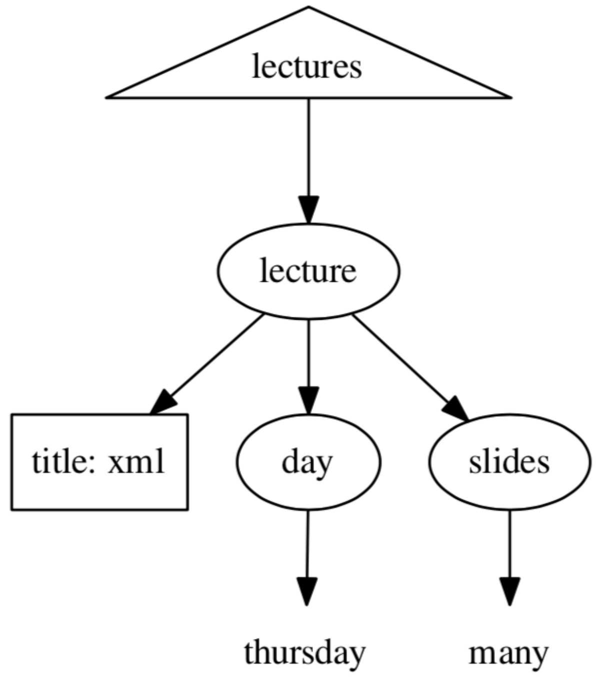
- XML **attribute values** must be in ” (**double quote**s).
- XML **documents** are encoded as **linear strings**.
- XML **documents** begin with a special **processing instruction**, the prologue / header.

Technology
Arts Sciences
TH Köln

XML (cont.)

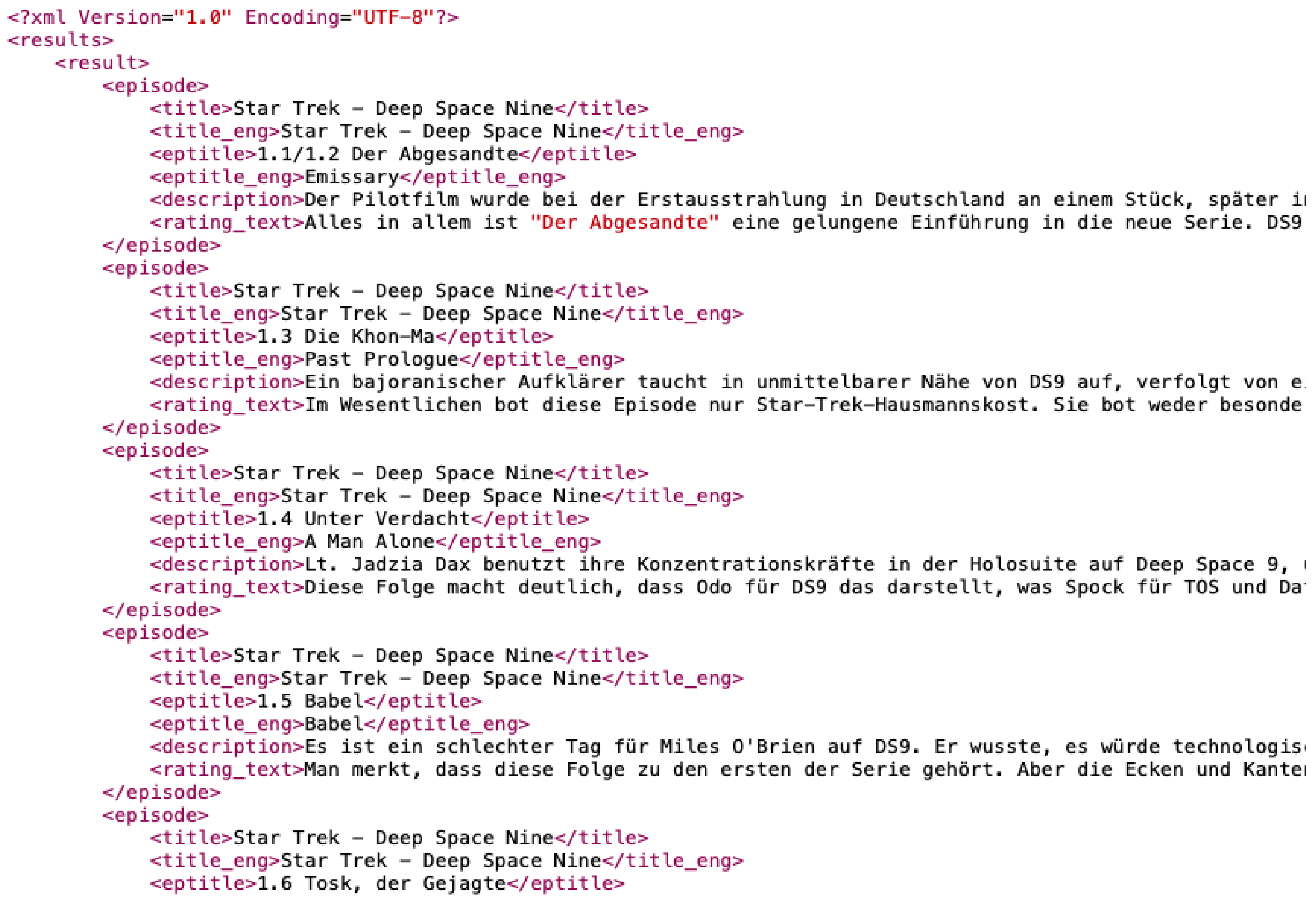


```
<lectures>
  <lecture title="xml">
    <day>thursday</day>
    <slides>many</slides>
  </lecture>
</lectures>
```



- **Very common** data format for web content files.
- **Solves** a lot of **problems** (namespaces, encoding, embedded data).
- Not very readable, very verbose.

XML: Example



```
<?xml Version="1.0" Encoding="UTF-8"?>
<results>
    <result>
        <episode>
            <title>Star Trek - Deep Space Nine</title>
            <title_eng>Star Trek - Deep Space Nine</title_eng>
            <eptitle>1.1/1.2 Der Abgesandte</eptitle>
            <eptitle_eng>Emissary</eptitle_eng>
            <description>Der Pilotfilm wurde bei der Erstausstrahlung in Deutschland an einem Stück, später i
            <rating_text>Alles in allem ist "Der Abgesandte" eine gelungene Einführung in die neue Serie. DS9
        </episode>
        <episode>
            <title>Star Trek - Deep Space Nine</title>
            <title_eng>Star Trek - Deep Space Nine</title_eng>
            <eptitle>1.3 Die Khon-Ma</eptitle>
            <eptitle_eng>Past Prologue</eptitle_eng>
            <description>Ein bajoranischer Aufklärer taucht in unmittelbarer Nähe von DS9 auf, verfolgt von e
            <rating_text>Im Wesentlichen bot diese Episode nur Star-Trek-Hausmannskost. Sie bot weder besonde
        </episode>
        <episode>
            <title>Star Trek - Deep Space Nine</title>
            <title_eng>Star Trek - Deep Space Nine</title_eng>
            <eptitle>1.4 Unter Verdacht</eptitle>
            <eptitle_eng>A Man Alone</eptitle_eng>
            <description>Lt. Jadzia Dax benutzt ihre Konzentrationskräfte in der Holosuite auf Deep Space 9,
            <rating_text>Diese Folge macht deutlich, dass Odo für DS9 das darstellt, was Spock für TOS und Da
        </episode>
        <episode>
            <title>Star Trek - Deep Space Nine</title>
            <title_eng>Star Trek - Deep Space Nine</title_eng>
            <eptitle>1.5 Babel</eptitle>
            <eptitle_eng>Babel</eptitle_eng>
            <description>Es ist ein schlechter Tag für Miles O'Brien auf DS9. Er wusste, es würde technologis
            <rating_text>Man merkt, dass diese Folge zu den ersten der Serie gehört. Aber die Ecken und Kante
        </episode>
        <episode>
            <title>Star Trek - Deep Space Nine</title>
            <title_eng>Star Trek - Deep Space Nine</title_eng>
            <eptitle>1.6 Tosk, der Gejagte</eptitle>
```

Technology
Arts Sciences
TH Köln

XML: well formed

An **element** always has a **start tag** and an **end tag**.

- This is a **`<StartTag>`**.
- This is the end of the **`</StartTag>`**.
- The tag names are **case-sensitive**.
- **`<tag></tag>`** and not `<tag></Tag>`.

Empty elements, called **milestone elements**, can be abbreviated

- `<emptyTag />`.
- Meaning: **attributes may be included**, but **no content** may be placed **between** the **tags**.

XML: well formed (cont.)

All elements must be **nested correctly**!

```
<up>
   <down>Text</down>
</up>
```

```
<up>
<down>Text</up>
</down>
```

Element names

- Must begin with a letter, underscore, or colon.
- Can contain letters, numbers, hyphens, periods, or underscores, as well as umlauts and accents.

An **XML document** has **only exactly one root node**!

XML: Attributes

Elements can be defined in **more detail** by **attributes**.

- **Attributes** are **always** in the **start tag!**
- `<name AttrName="Attributwert"> </name>`

Properties of attributes

- Choice of designer on what information is **element-worthy** and which is **attribute-worthy.**
- **Multiple attributes** are allowed per element.
- Same named attributes in different elements

```
<book>
     <person role="author">Stephen King</person>
     <person role="translator">Ralph Meier</person>
</book>
```

XML: Prolog

An XML document always starts with a prolog that contains

- Declaration

```
<?xml version="1.0"?>
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
```

- Processing information (optional)

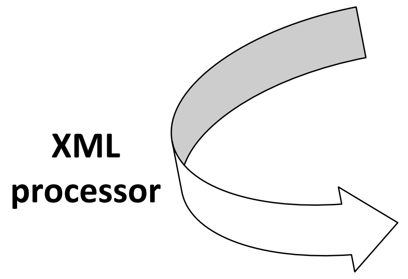
```
<?xsl-stylesheet type="text/xsl" href="myown.xsl"?>
```

- Embedding of the document model (optional)

DTD, XML Schema, RelaxNG, Schematron

```
<!DOCTYPE tei SYSTEM "tei.dtd">
```

This element is encoded as
&lt;code&gt;&lt;Element&gt;...
&lt;/Element&gt;&lt;/code&gt;



This element is encoded as
<code><Element>...</Element></code>

XML: Entities

Entities stand for **something else**.

- To **avoid conflicts** during XML processing!
- XML allows to **set up a reference** that points to an **entity**.
- The **XML processor replaces** the references.

Some predefined entity references e.g., special characters:

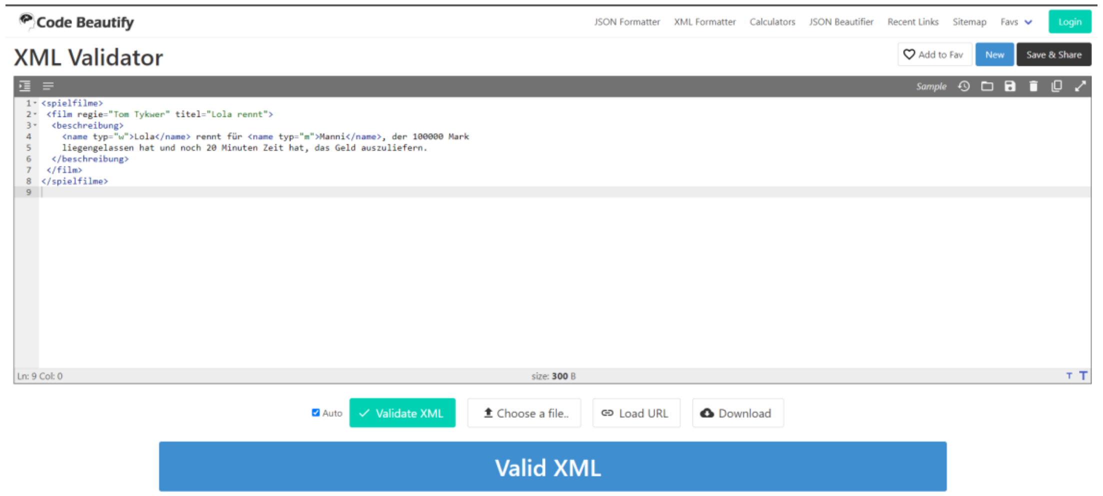
`&lt;`	<
`&gt;`	>
`&amp;`	&

Technology
Arts Sciences
TH Köln

XML: Verification

The verification is done by an XML parser. An XML document

- **Must be well-formed** and **can be valid**.
- Is well-formed if it **complies with the rules of the XML standard**.
- Is valid if it is **well-formed and conforms to the grammar** of the XML schema.



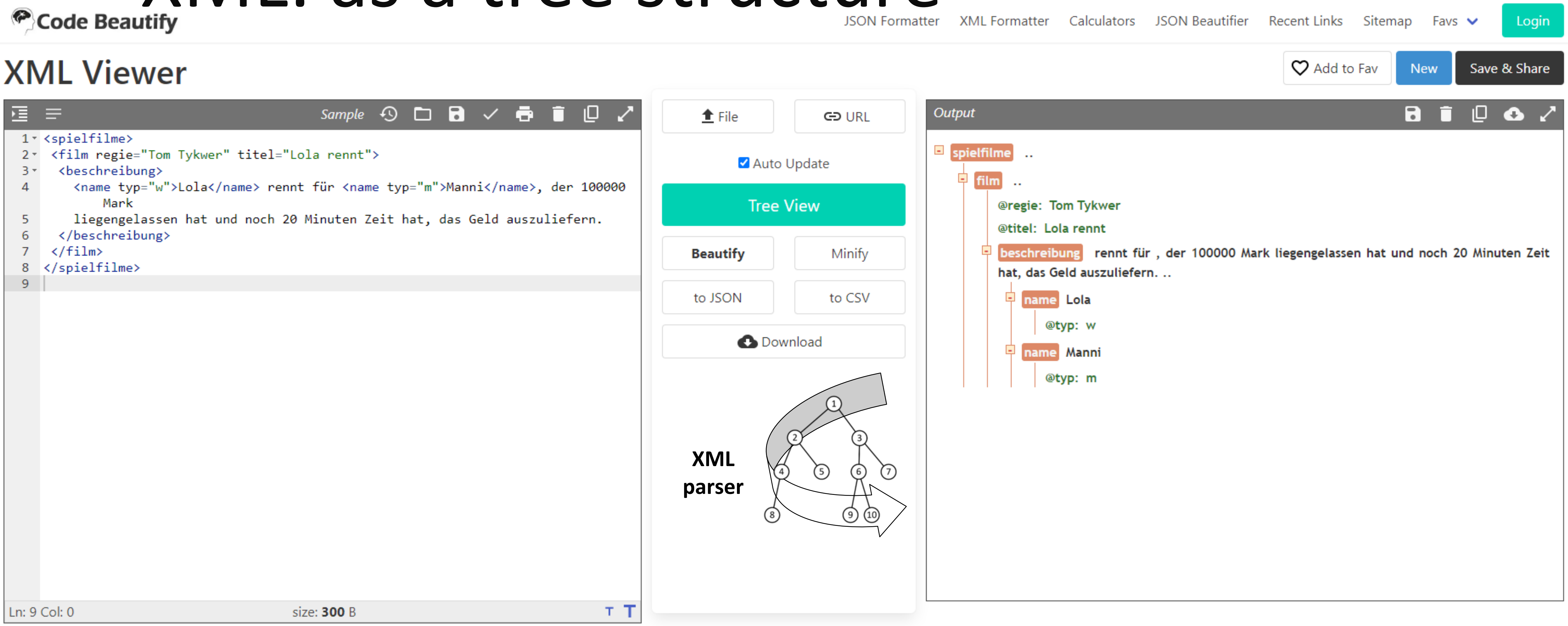
```
<spielfilme>
 <film regie="Tom Tykwer" titel="Lola rennt">
  <beschreibung>
    <name typ="w">Lola</name> rennt für <name typ="m">Manni</name>, der 100000 Mark
    liegengelassen hat und noch 20 Minuten Zeit hat, das Geld auszuliefern.
  </beschreibung>
 </film>
</spielfilme>
```

https://codebeautify.org/xmlvalidator

JSON versus XML

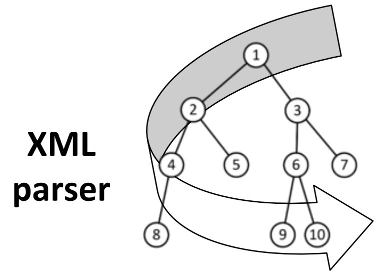
It is based on JavaScript language.	It is derived from SGML.
It is a way of **representing objects**.	It is a markup language and **uses tag structure to represent data items.**
It does not provide any support for **namespaces**.	It supports namespaces.
It supports **arrays**.	It doesn’t support arrays.
Its files are very **easy to read** as compared to XML.	Its documents are **comparatively difficult to read** and interpret.
It doesn’t use end tag.	It has **start** and **end tags**.
It is less **secured**.	It is more secured than JSON.
It doesn’t support **comments**.	It supports comments.
It supports only UTF-8 **encoding**.	It supports various encoding.

XML: as a tree structure



```
<spielfilme>
 <film regie="Tom Tykwer" titel="Lola rennt">
  <beschreibung>
    <name typ="w">Lola</name> rennt für <name typ="m">Manni</name>, der 100000
        Mark
    liegengelassen hat und noch 20 Minuten Zeit hat, das Geld auszuliefern.
  </beschreibung>
 </film>
</spielfilme>
```

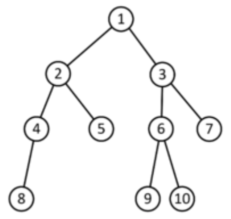
```
spielfilme ..
  film ..
    @regie: Tom Tykwer
    @titel: Lola rennt
    beschreibung  rennt für , der 100000 Mark liegengelassen hat und noch 20 Minuten Zeit
    hat, das Geld auszuliefern. ..
      name Lola
        @typ: w
      name Manni
        @typ: m
```



https://codebeautify.org/xmlviewer

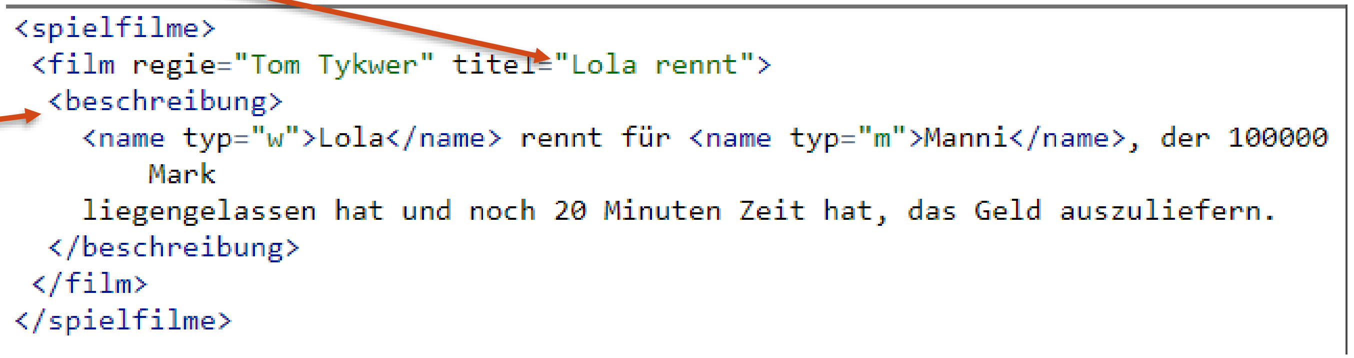
XML: Xpath Example

Xpath uses **path expressions** to **select nodes or node-sets** in an XML document.



/film[@regie='Tom Tykwer']/@titel

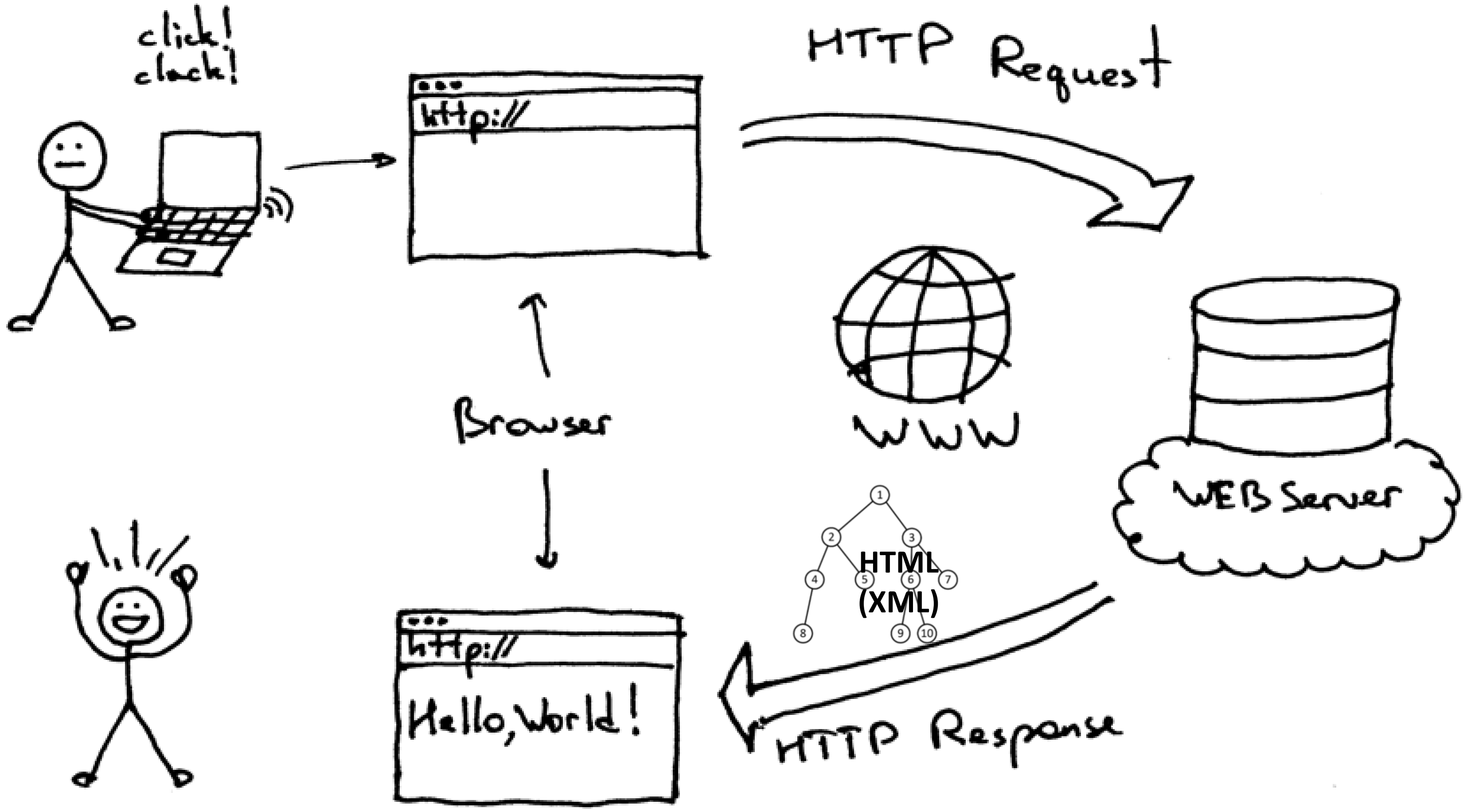
/film/beschreibung



```
<spielfilme>
 <film regie="Tom Tykwer" titel="Lola rennt">
  <beschreibung>
    <name typ="w">Lola</name> rennt für <name typ="m">Manni</name>, der 100000
        Mark
    liegengelassen hat und noch 20 Minuten Zeit hat, das Geld auszuliefern.
  </beschreibung>
 </film>
</spielfilme>
```

...

Web Scraping



https://monashdatafluency.github.io/python-web-scraping/section-2-HTML-based-scraping/

Training #1

Your web browser interprets HTML files (a similar type as XML).

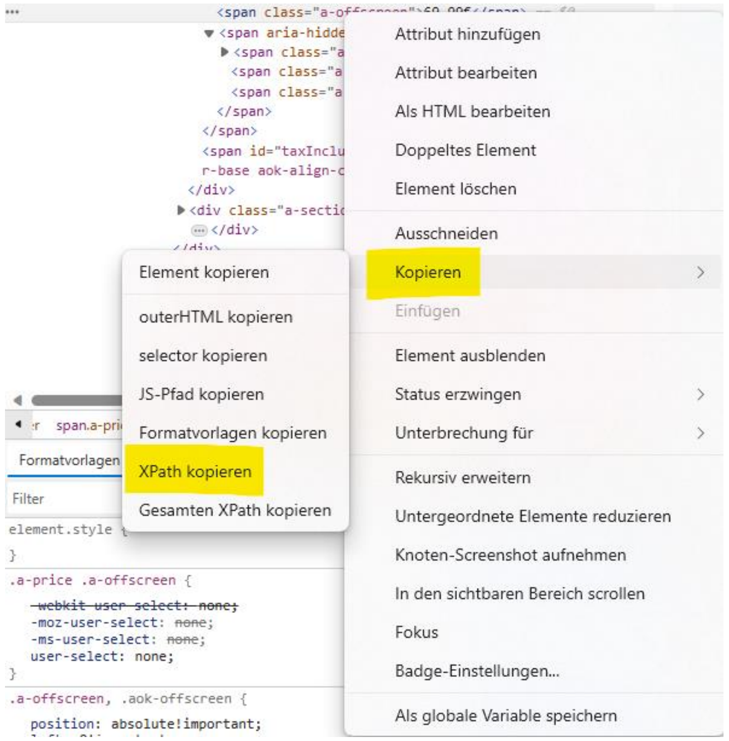
- Open a web site of your interest, e.g.,
  - Buecher.de
  - Transfermarkt and your favourite club
  - and click on Option+Command+U (Mac) or F12 (Windows) on your keyboard to open the developer mode (DevTools).
- Click on the "Select element" button:
  and click on something on the web-page.
- Right click on "Copy > XPath" of the element in the HTML document and note it down.

What element(s) interest you?
→ Imagine you can extract this/these element/s from various categories/products/players.

How could these elements benefit someone from an analytical perspective?

Training #1 (cont.)

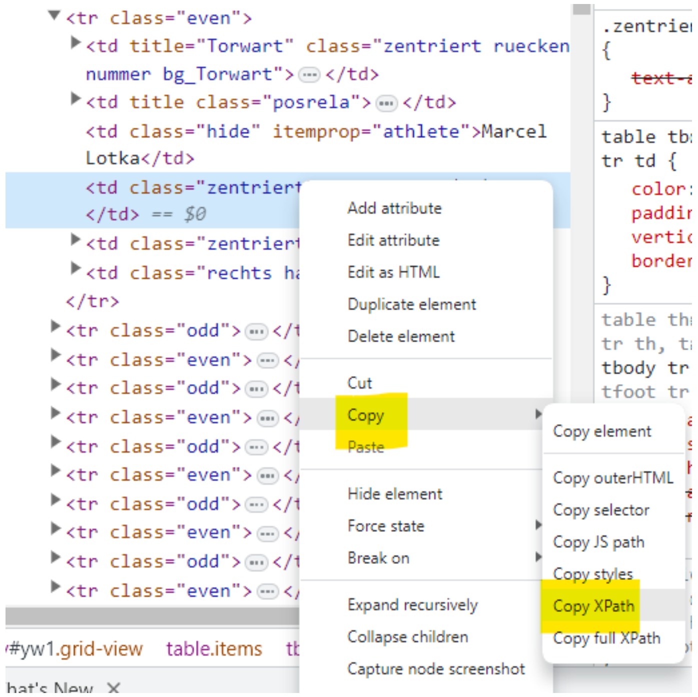
Microsoft Edge



```
<span class="a-offscreen">60,90€</span> == $0
▼<span aria-hidde
  ▶<span class="a
    <span class="a
    <span class="a
  </span>
</span>
<span id="taxInclu
r-base aok-align-c
</div>
▶<div class="a-secti
  ⋯ </div>
</div>
```

```
element.style {
}
.a-price .a-offscreen {
    ~~-webkit-user-select: none;~~
    -moz-user-select: ~~none~~;
    -ms-user-select: ~~none~~;
    user-select: none;
}
.a-offscreen, .aok-offscreen {
    position: absolute!important;
```

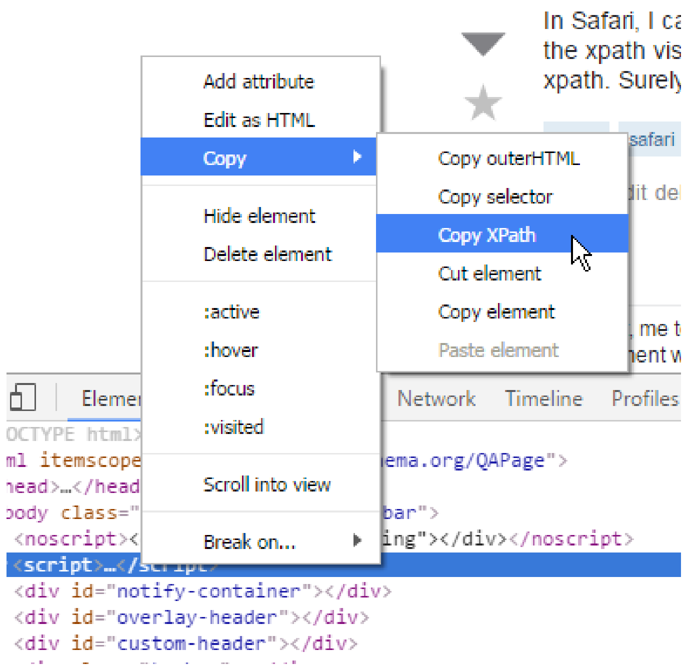
Chrome



```
▼<tr class="even">
  ▶<td title="Torwart" class="zentriert ruecken
    nummer bg_Torwart">⋯</td>
  ▶<td title class="posrela">⋯</td>
    <td class="hide" itemprop="athlete">Marcel
    Lotka</td>
    <td class="zentriert
    </td> == $0
  ▶<td class="zentriert
  ▶<td class="rechts h
  </tr>
▶<tr class="odd">⋯</t
▶<tr class="even">⋯<
▶<tr class="odd">⋯</t
▶<tr class="even">⋯<
▶<tr class="odd">⋯</t
▶<tr class="even">⋯<
▶<tr class="odd">⋯</t
▶<tr class="even">⋯<
▶<tr class="odd">⋯</t
▶<tr class="even">⋯<
```

```
.zentrie
{
  ~~text-a~~
}
table tb
tr td {
  color:
  paddi
  verti
  border
}
table th
tr th, t
tbody tr
tfoot tr
```

Safari (Turn on Developer View)



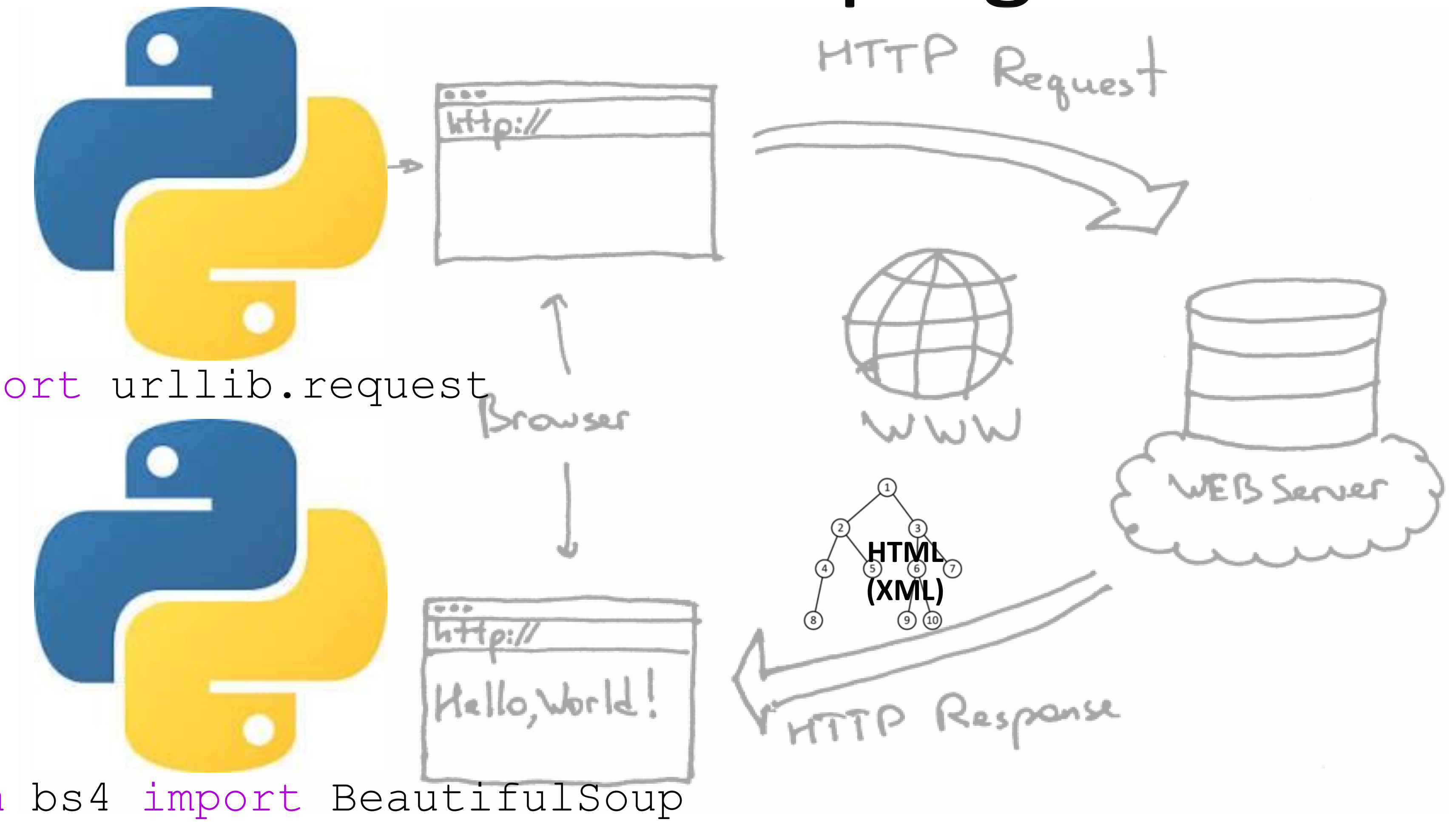
In Safari, I ca
the xpath vis
xpath. Surely

```
OCTYPE html>
ml itemscope                    ema.org/QAPage">
ead>…</head
ody class="                     bar">
 <noscript><                    ing"></div></noscript>
 <script>…</script>
 <div id="notify-container"></div>
 <div id="overlay-header"></div>
 <div id="custom-header"></div>
```

Training#1

What element(s) interest you?	How could these elements benefit someone from an analytical perspective?

Demo: First Web Scraping

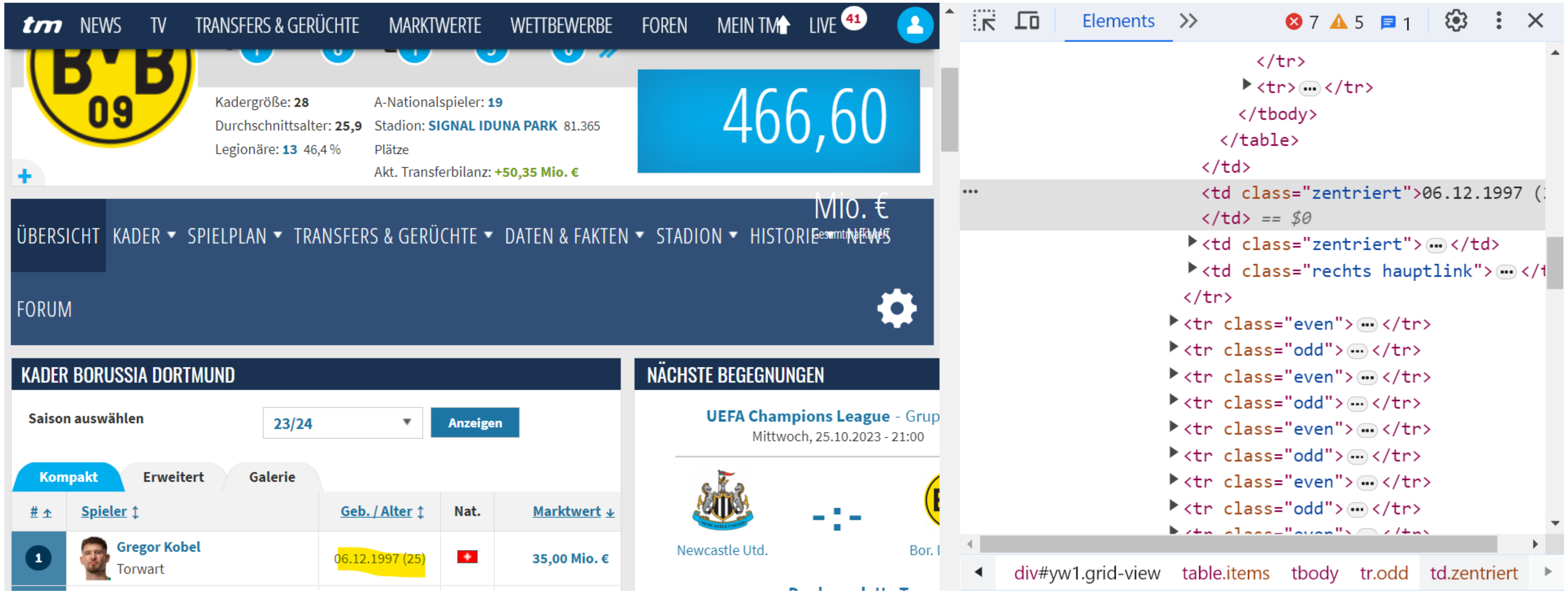


```
import urllib.request
```

```
from bs4 import BeautifulSoup
```

https://monashdatafluency.github.io/python-web-scraping/section-2-HTML-based-scraping/

Demo: First Web Scraping



Kadergröße: **28**
Durchschnittsalter: **25,9**
Legionäre: **13** 46,4 %
A-Nationalspieler: **19**
Stadion: **SIGNAL IDUNA PARK** 81.365 Plätze
Akt. Transferbilanz: **+50,35 Mio. €**

466,60 Mio. €
Gesamtmarktwert

KADER BORUSSIA DORTMUND

Saison auswählen 23/24 Anzeigen

Kompakt | Erweitert | Galerie

#	Spieler	Geb. / Alter	Nat.	Marktwert
1	Gregor Kobel Torwart	06.12.1997 (25)		35,00 Mio. €

NÄCHSTE BEGEGNUNGEN

UEFA Champions League - Grup
Mittwoch, 25.10.2023 - 21:00

Newcastle Utd. -:- Bor.

```
            </tr>
          ▶<tr>…</tr>
          </tbody>
        </table>
      </td>
      <td class="zentriert">06.12.1997 (
      </td> == $0
     ▶<td class="zentriert">…</td>
     ▶<td class="rechts hauptlink">…</t
    </tr>
   ▶<tr class="even">…</tr>
   ▶<tr class="odd">…</tr>
   ▶<tr class="even">…</tr>
   ▶<tr class="odd">…</tr>
   ▶<tr class="even">…</tr>
   ▶<tr class="odd">…</tr>
   ▶<tr class="even">…</tr>
   ▶<tr class="odd">…</tr>
```

div#yw1.grid-view table.items tbody tr.odd td.zentriert

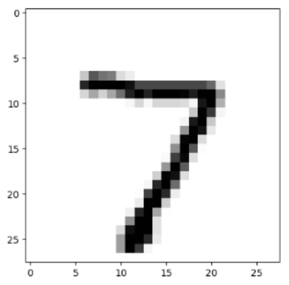
Xpath:[@id="yw1"]/table/tbody/tr[1]/td[3]

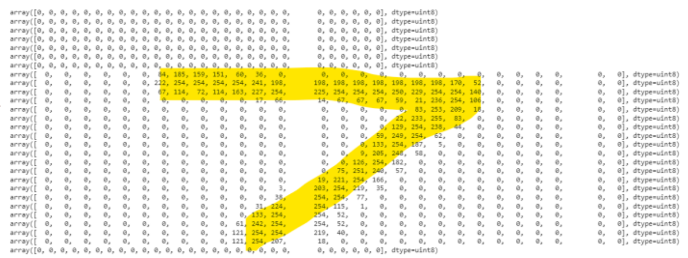
Takeaways

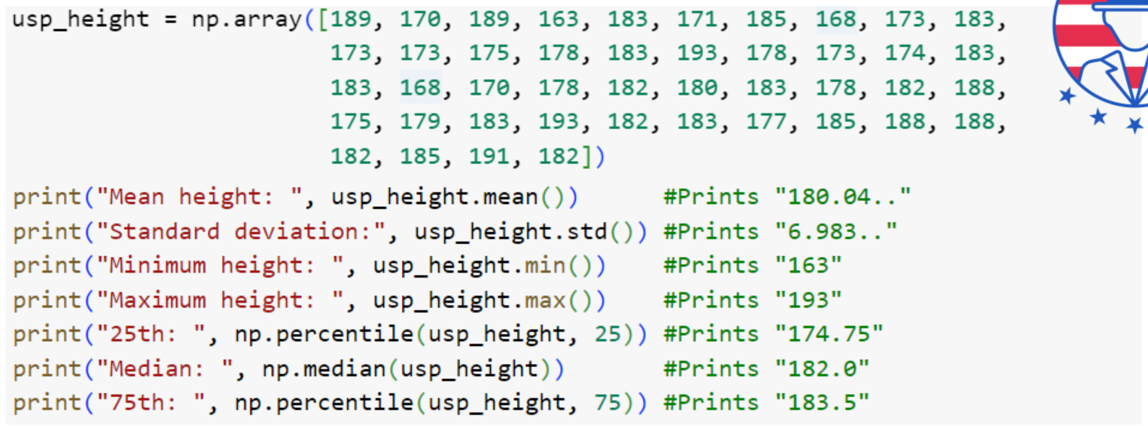
- CSV, JSON, and XML are typical **text-based** file formats that **you will encounter in the wild.**
- All have their **use cases and applications,** and it always depends on what you would like to accomplish!
- It’s **good practice to know all three** - one of these is usually available and applicable.

Outlook

- In the next week 5 we will dive deep into Python NumPy.
- This is the core library for numerically typed arrays. You might use it in your project for Data Wrangling purposes.







```
usp_height = np.array([189, 170, 189, 163, 183, 171, 185, 168, 173, 183,
                       173, 173, 175, 178, 183, 193, 178, 173, 174, 183,
                       183, 168, 170, 178, 182, 180, 183, 178, 182, 188,
                       175, 179, 183, 193, 182, 183, 177, 185, 188, 188,
                       182, 185, 191, 182])
print("Mean height: ", usp_height.mean())       #Prints "180.04.."
print("Standard deviation:", usp_height.std()) #Prints "6.983.."
print("Minimum height: ", usp_height.min())    #Prints "163"
print("Maximum height: ", usp_height.max())    #Prints "193"
print("25th: ", np.percentile(usp_height, 25)) #Prints "174.75"
print("Median: ", np.median(usp_height))        #Prints "182.0"
print("75th: ", np.percentile(usp_height, 75)) #Prints "183.5"
```

- In week 6+7 we will dive deep into Python Pandas‘ data frames to arrange tabular data to analytical goals.

See you again next week **online**!

Questions?