

Social Data Science Base Camp

Digital Data Structures

Friedolin Merhout
MSc Social Data Science
September 20, 2021

UNIVERSITY OF COPENHAGEN



Outline for today

1. Welcome back and check-in
2. Lecture
 1. Data Formatting
 2. Hypertext Markup Language (HTML)
 3. Cascading Style Sheets (CSS)
 4. JavaScript Object Notation (JSON)
3. Demo
 1. Hands on "W38Mon-Demo-DigitalDataStructures.ipynb"
4. Review Survey
5. Exercise Preview

Welcome back and check-in



[Image Source](#)

- Recap
 - (Conditional) flow control
 - Naming Practices
 - Containers / Data Structures
 - Access, order, duplication, mutability
- Outlook
 - Wednesday: Loops / (Repetition) flow control
 - Next Monday: Review and Building Bigger Programs

More Structures I

Digital Data Structures

Building on slides by Ulf Aslak

Wake Up Exercise: Sort that file

File Name	Data Type
"description.txt"	
"rows.csv"	
"table.html"	
"observations.json"	
"data.xlsx"	

Wake Up Exercise: Sort that file

File Name	Data Type
"description.txt"	Non-tabular
"rows.csv"	
"table.html"	
"observations.json"	
"data.xlsx"	

Wake Up Exercise: Sort that file

File Name	Data Type
"description.txt"	Non-tabular
"rows.csv"	Tabular
"table.html"	
"observations.json"	
"data.xlsx"	

Wake Up Exercise: Sort that file

File Name	Data Type
"description.txt"	Non-tabular
"rows.csv"	Tabular
"table.html"	Non-tabular
"observations.json"	
"data.xlsx"	

Wake Up Exercise: Sort that file

File Name	Data Type
"description.txt"	Non-tabular
"rows.csv"	Tabular
"table.html"	Non-tabular
"observations.json"	Non-tabular
"data.xlsx"	

Wake Up Exercise: Sort that file

File Name	Data Type
"description.txt"	Non-tabular
"rows.csv"	Tabular
"table.html"	Non-tabular
"observations.json"	Non-tabular
"data.xlsx"	Tabular

What does this mean?

Data Formatting

- Gist:
 - Varying nature and application of data leads to wide variation in representation and storage formats
- Three main types of formats
 - Tabular (CSV, XLSX)
 - Non-tabular (JSON, XML)
 - Associative (*Networks*)

Data Formatting

- Gist:
 - Varying nature and application of data leads to wide variation in representation and storage formats
- Three main types of formats
 - **Tabular (CSV, XLSX)**
 - Structure: rows and columns
 - Common terminology: rows/observations/cases, columns/vectors/variables

	description	role
0	David Dreyer Lassen is the Director of SODAS a...	SODAS steering committee
1	Morten Axel Pedersen is Deputy Director of SOD...	SODAS steering committee
2	Rebecca Adler-Nissen is Professor in Political...	SODAS steering committee
3	Sune Lehmann is a Professor of Complexity and ...	SODAS steering committee
4	Anders Blok is Associate Professor in Sociolog...	SODAS steering committee
5	Søren Kyllingsbæk is Professor in Cognitive Ps...	SODAS steering committee
6	Robert Böhm is a Professor of Applied Social P...	SODAS steering committee

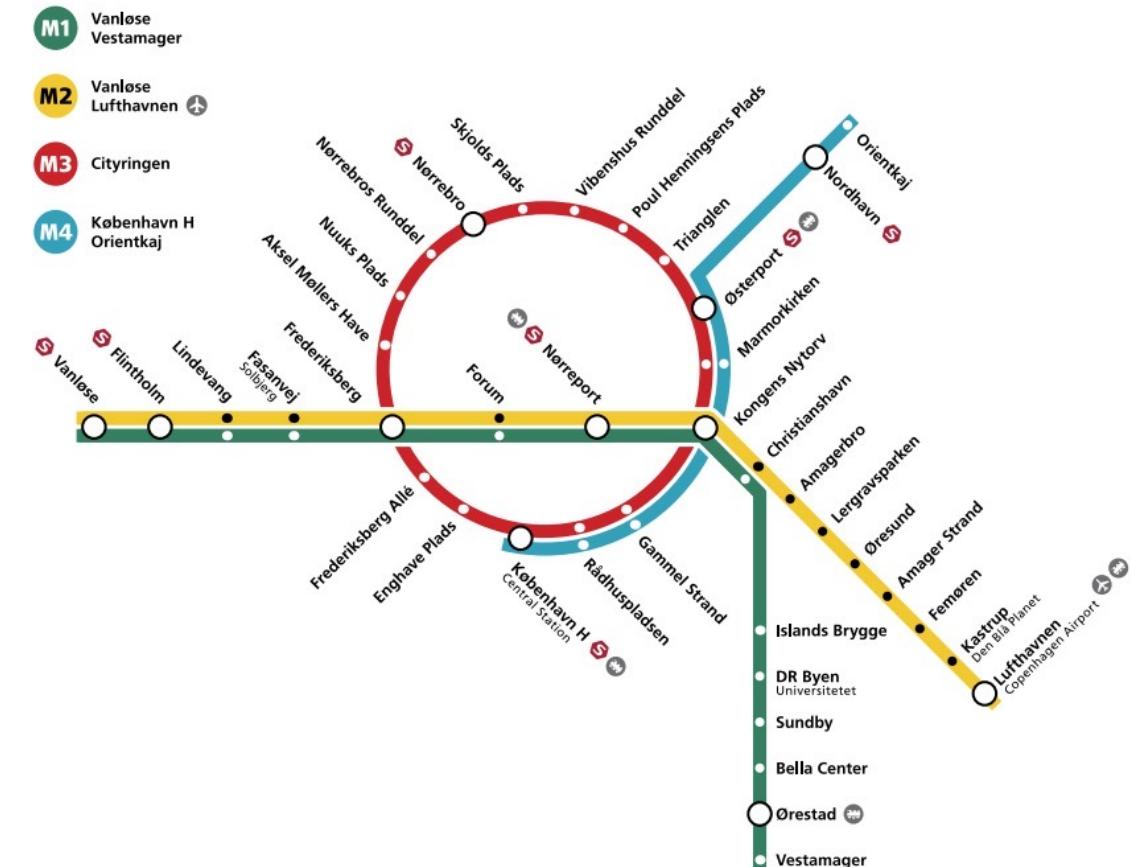
Data Formatting

- Gist:
 - Varying nature and application of data leads to wide variation in representation and storage formats
- Three main types of formats
 - **Non-tabular (JSON, XML)**
 - Structure: varying but often tree-like (stem-branches-leaves) or hierarchical
 - Common terminology: keys/attribute, values

```
</div>
<!-- Standard doctype: -->
<h1 class="title">
  People
</h1>
<h2>
  SODAS steering committee
</h2>
<table class="table table-bordered table-hover">
  <tbody>
    <tr>
      <td>
        <p>
          
          <a href="https://www.economics.ku.dk/staff/vip/?pure=en/persons/28460" title="David Dreyer Lassen">
            David Dreyer Lassen
          </a>
          <span>
            is the Director of SODAS and professor of Economics at the University of Copenhagen. He is currently the
          </span>
        </p>
      </td>
    </tr>
  </tbody>
</table>
```

Data Formatting

- Gist:
 - Varying nature and application of data leads to wide variation in representation and storage formats
- Three main types of formats
 - **Associative (*Networks*)**
 - Structure: network (connected observations)
 - Common terminology: nodes/vertices, edge/tie/link



[Image Source](#)

Today's focus: non-tabular data in the wild

Digital Data Structures

HyperText Markup Language

HyperText Markup Language

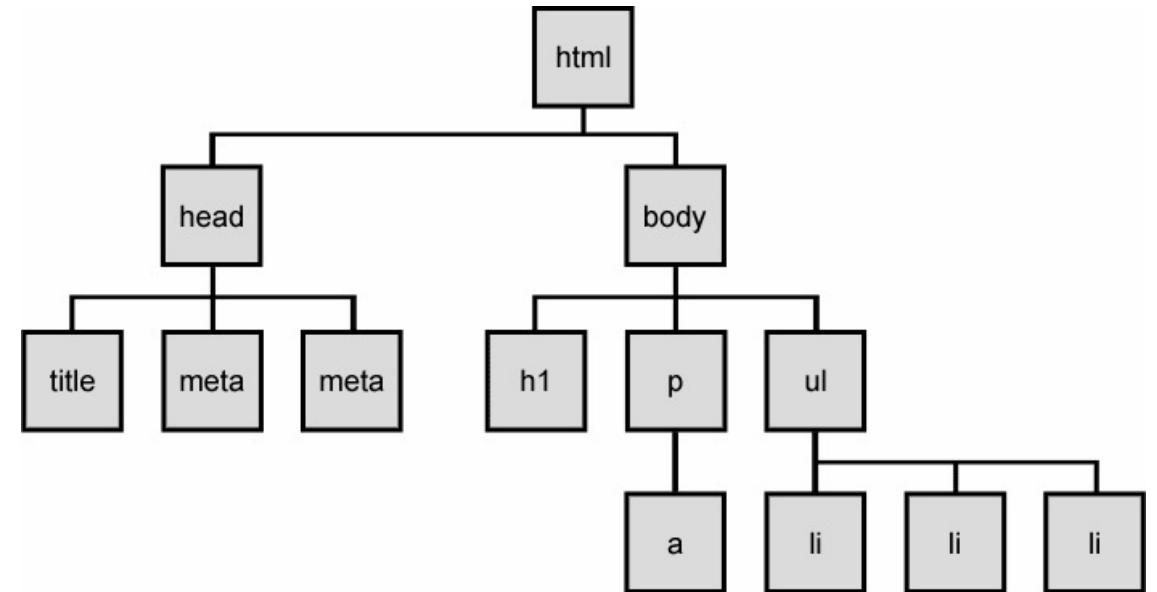
- **Web pages** are built up and stored as plain text, encoding hypertext controlling appearance and functionality of pages
- To ensure proper presentation of pages, links, lists, etc., requires **common format** for marking up text: That's **HTML**.



Image Source

HTML Fundamentals

- HyperText Markup Language is a hierarchical format using **pre-defined tags** to structure text and media content
- Existing tags include indicators for headings `<h1>`, image embedders ``, and hyperlinks `<a>` and many more



[Image Source](#)

HTML Tags I

- Some tags **wrap around** content and need to be closed, e.g.
 - <h1>This is a heading</h1>
- Others **contain** content and can stand by themselves, e.g.
 -

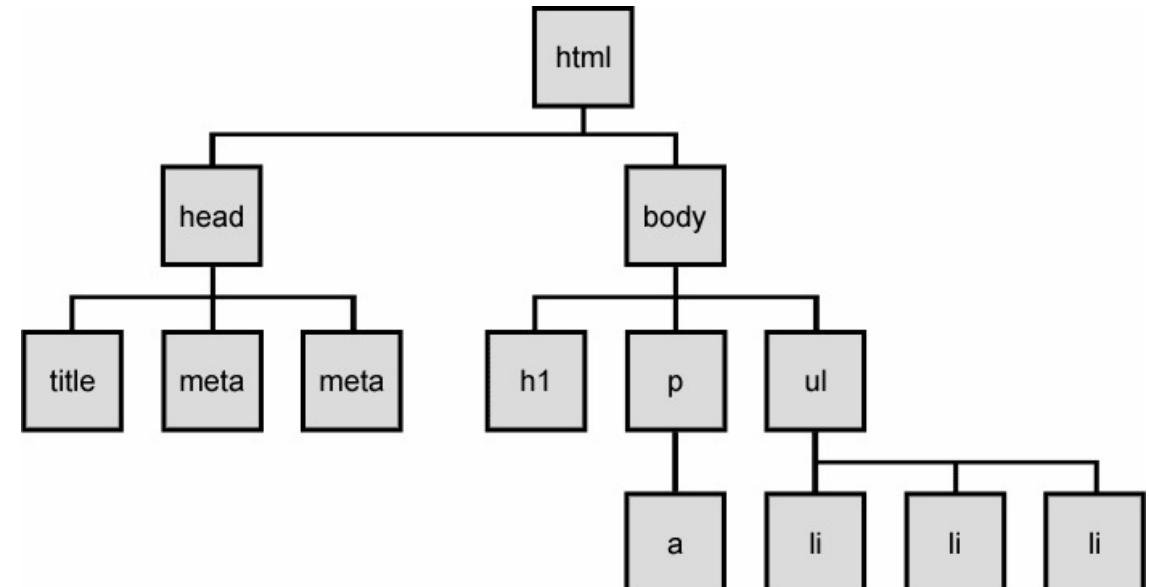


Image Source

HTML Tags II

- Tags can have **attributes** with additional information, e.g.

```
  
<a  
href="https://fmerhout.github.i  
o">a link</a>
```

- One fundamental tag is `<div>`, defining sections containing distinct content

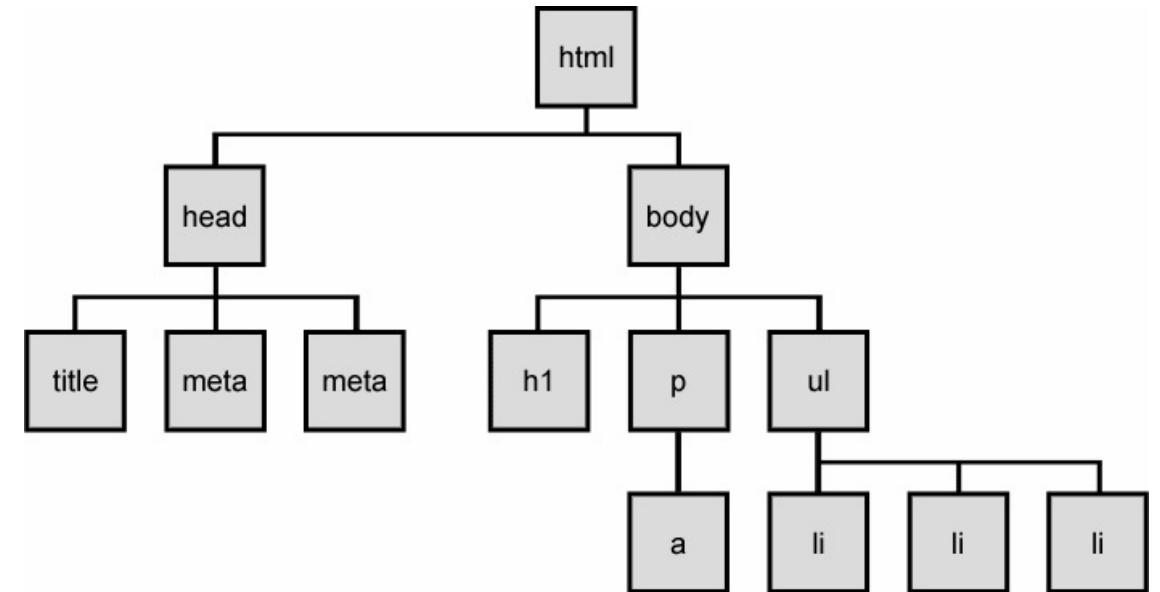
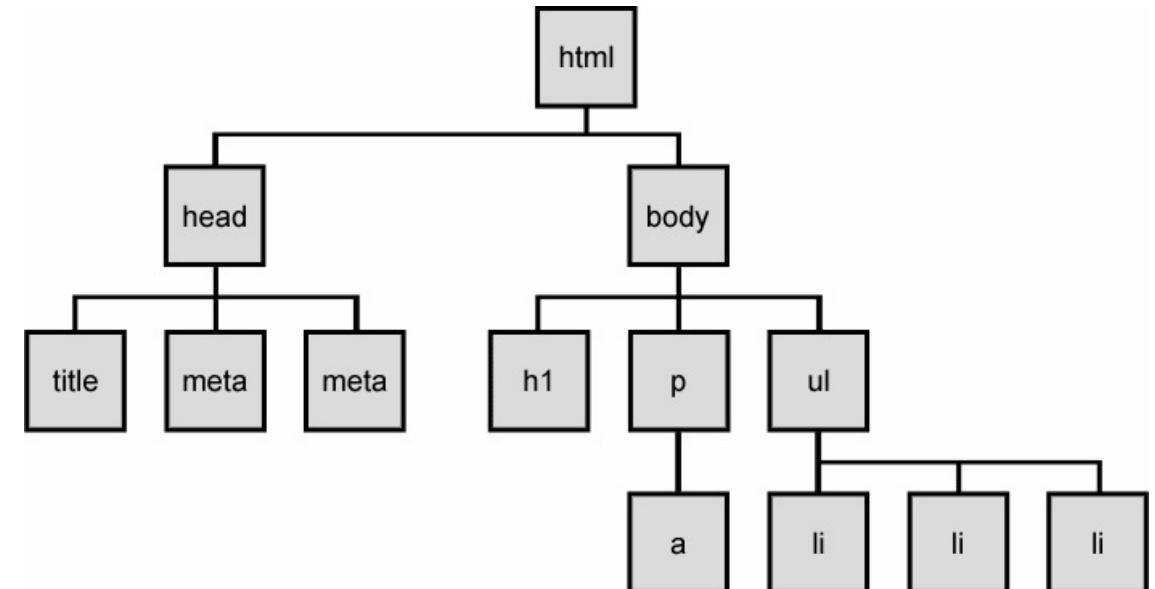


Image Source

HTML in Practice

- Browser create internal representation of plain text HTML called **Document Object Model**
- Modern websites build on frameworks allowing users to interact with it, **dynamically** loading or changing content



[Image Source](#)

HTML In-Lecture Exercise



- Open the following page in Firefox or Chrome

bit.ly/3kjGmPN

- Right click on “Social Data Science” and click “Inspect”
- What is the hyperlink attribute of the tag? Does it look normal?
 - **2020-2021/STUDYBOARD_2162**
- What are the two tags enclosing it? What do you think they mean?
 - **** and ****

Digital Data Structures

Cascading Style Sheets

Cascading Style Sheets (CSS)

- CSS is a language to define how documents written in markup are formatted
- It allows separating content from presentation, simplifying the content documents
- Also, benefits page loading speed, adaptation to medium, and uniform presentation



[Image Source](#)

CSS in Practice and SDS Applications

- Content document meta-data reference style sheet, which browser loads separately
 - Connection between content and style via attributes e.g., class or id
 - Improves content accessibility for data collection via CSS attributes

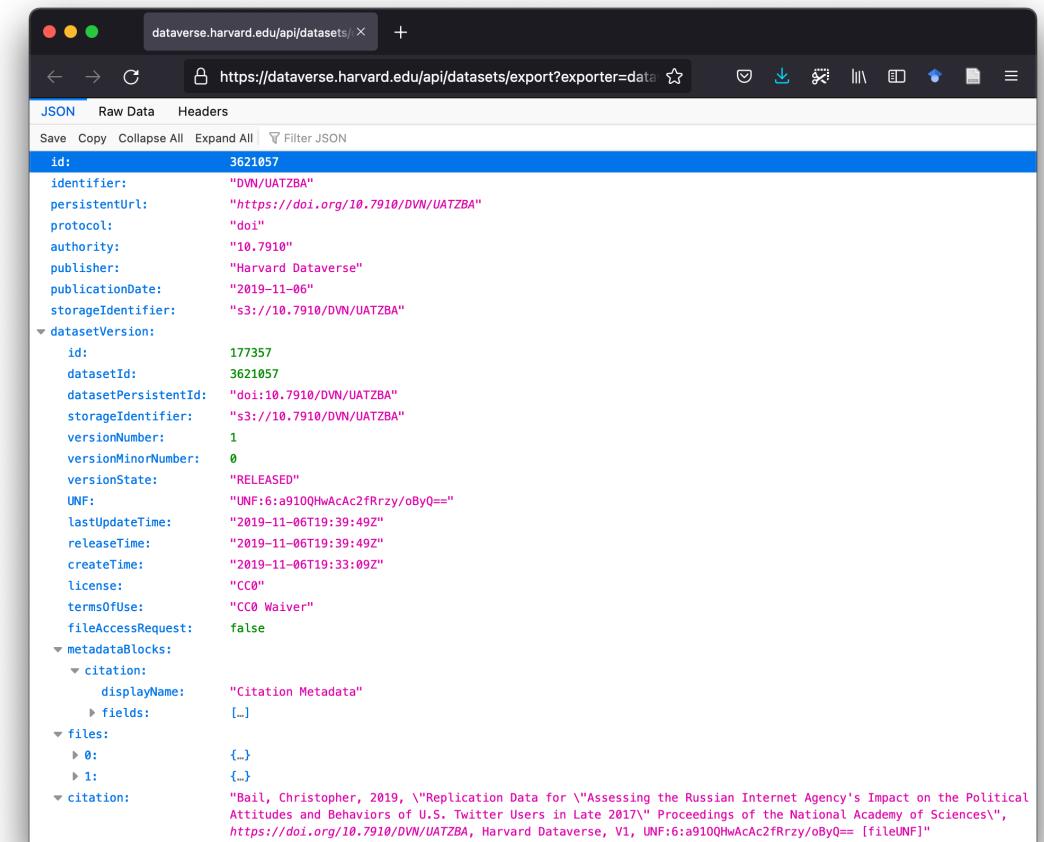
```
/*!
 * Bootstrap v3.3.7 (http://getbootstrap.com)
 * Copyright 2011-2016 Twitter, Inc.
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)
 */
/* normalize.css v8.0.1 | MIT License | github.com/necolas/normalize.css */html{line-height:1.15;-webkit-text-size-adjust:100%}body{margin:0}main{display:block}h1{font-size:2em;margin:.67em 0}hr{border:none;outline:none}a{background-color:transparent}abbr[title]{border-bottom:none;text-decoration:underline;text-decoration:underline dotted}b,strong{font-weight:bolder}code,kbd,samp{font-family:monospace,monospace;font-size:1em}small{font-size:80%}sub,sup{line-height:1;position:relative;vertical-align:baseline}sub{bottom:-.25em}sup{top:-.5em}img{border-style:none}button,input,optgroup,select,textarea{font-family:inherit;font-size:100%;line-height:1.15;margin:0}button,input{overflow:visible}button,select{text-transform:none}[type=button],[type=reset],[type=submit],button{border:none;outline:none}[type=button]:-moz-focus-inner,[type=reset]:-moz-focus-inner,[type=submit]:-moz-focus-inner,button:-moz-focus-inner{border-style:none;padding:0}button:-moz-focusing,[type=reset]:-moz-focusing,[type=submit]:-moz-focusing,button:-moz-focusing{outline:1px dottedButtonText}fieldset{padding:.35em;.75em}.625emlegend{box-sizing:border-box;color:inherit;display:table;max-width:100%;padding:0;white-space:normal}progress{vertical-align:baseline}textarea{overflow:auto}input{type=checkbox},input{type=radio}{box-sizing:border-box;padding:0}input{type=number}:--webkit-inner-spin-button,[type=number]:--webkit-outer-spin-button{height:auto}input{type=search}{-webkit-appearance:textfield;outline-offset:-2px}input{type=search}:--webkit-search-decoration{border:none}details{display:block}summary{display:list-item}template{display:none}[hidden]{display:none}@font-face{font-family:ku-symbols;font-display:swap;src:url("../fonts/KU.eot");src:url("../fonts/KU.eot?#iefix") format('embedded-opentype'),url("../fonts/KU.woff2") format('woff2'),url("../fonts/KU.woff") format('woff')},url("../fonts/KU.ttf") format('truetype'),url("../fonts/KU.svg#ku_symbols") format('svg')}@font-face{font-family:'Glyphicons Halflings';font-display:swap;src:url("../fonts/glyphicons-halflings-regular.eot");src:url("../fonts/glyphicons-halflings-regular.eot?#iefix") format('embedded-opentype'),url("../fonts/glyphicons-halflings-regular.woff2") format('woff2'),url("../fonts/glyphicons-halflings-regular.woff") format('woff')},url("../fonts/glyphicons-halflings-regular.ttf") format('truetype'),url("../fonts/glyphicons-halflings-regular.svg") format('svg')}.glyphicon{position:relative;top:1px;display:inline-block;font-family:'Glyphicons Halflings';font-style:normal;font-weight:400;line-height:1;-webkit-font-smoothing:antialiased;-moz-osx-font-smoothing:grayscale}.glyphicon-asterisk:before{content:'\002a'}.glyphicon-plus:before{content:'\002b'}.glyphicon-eur:before,.glyphicon-euro:before{content:'\20ac'}.glyphicon-minus:before{content:'\2022'}.glyphicon-cloud:before{content:'\2601'}.glyphicon-envelope:before{content:'\2709'}.glyphicon-pencil:before{content:'\270f'}.glyphicon-glass:before{content:'\e001'}.glyphicon-music:before{content:'\e002'}.glyphicon-search:before{content:'\e003'}.glyphicon-heart:before{content:'\e005'}.glyphicon-star:before{content:'\e006'}.glyphicon-star-empty:before{content:'\e007'}.glyphicon-user:before{content:'\e008'}.glyphicon-con-
```

Digital Data Structures

JavaScript Object Notation

JavaScript Object Notation (JSON)

- JSON is a **non-proprietary, language independent** data and file format
- Data stored in **name-value pairs** and arrays
- File extension **.json**

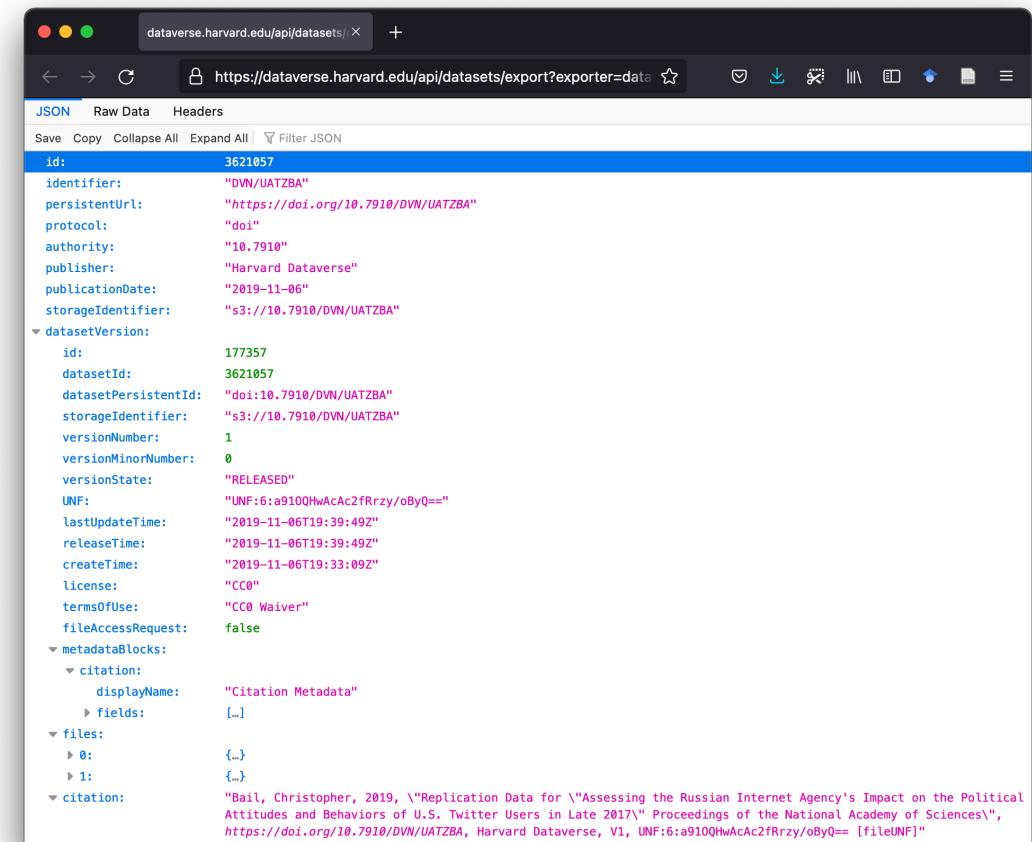


A screenshot of a web browser window displaying a JSON object. The URL in the address bar is <https://dataverse.harvard.edu/api/datasets/export?exporter=data>. The JSON object contains various fields such as id, identifier, persistentUrl, protocol, authority, publisher, publicationDate, storageIdentifier, datasetVersion, id, datasetId, datasetPersistentId, storageIdentifier, versionNumber, versionMinorNumber, versionState, UNF, lastUpdateTime, releaseTime, createTime, license, termsOfUse, fileAccessRequest, metadataBlocks, citation, fields, files, and citation. The JSON is displayed in a hierarchical tree view with expandable nodes.

```
id: 3621057
identifier: "DVN/UATZBA"
persistentUrl: "https://doi.org/10.7910/DVN/UATZBA"
protocol: "doi"
authority: "10.7910"
publisher: "Harvard Dataverse"
publicationDate: "2019-11-06"
storageIdentifier: "s3://10.7910/DVN/UATZBA"
datasetVersion:
  id: 177357
  datasetId: 3621057
  datasetPersistentId: "doi:10.7910/DVN/UATZBA"
  storageIdentifier: "s3://10.7910/DVN/UATZBA"
  versionNumber: 1
  versionMinorNumber: 0
  versionState: "RELEASED"
  UNF: "UNF:6:a910QhWAcAc2fRrzy/oByQ=="
  lastUpdateTime: "2019-11-06T19:39:49Z"
  releaseTime: "2019-11-06T19:39:49Z"
  createTime: "2019-11-06T19:33:09Z"
  license: "CC0"
  termsOfUse: "CC0 Waiver"
  fileAccessRequest: false
metadataBlocks:
  citation:
    displayName: "Citation Metadata"
    fields: [...]
files:
  0: {...}
  1: {...}
citation:
  "Bail, Christopher, 2019, \"Assessing the Russian Internet Agency's Impact on the Political Attitudes and Behaviors of U.S. Twitter Users in Late 2017\" Proceedings of the National Academy of Sciences\", https://doi.org/10.7910/DVN/UATZBA, Harvard Dataverse, V1, UNF:6:a910QhWAcAc2fRrzy/oByQ== [fileUNF]"
```

JSON Structure and Syntax

- Hierarchical or tree-like, unordered data structure – akin to Python dictionaries
- Each branch enclosed by curly braces ({}) with name-value pairs separated by commas
- Values can also be ordered data, in arrays using square brackets ([])

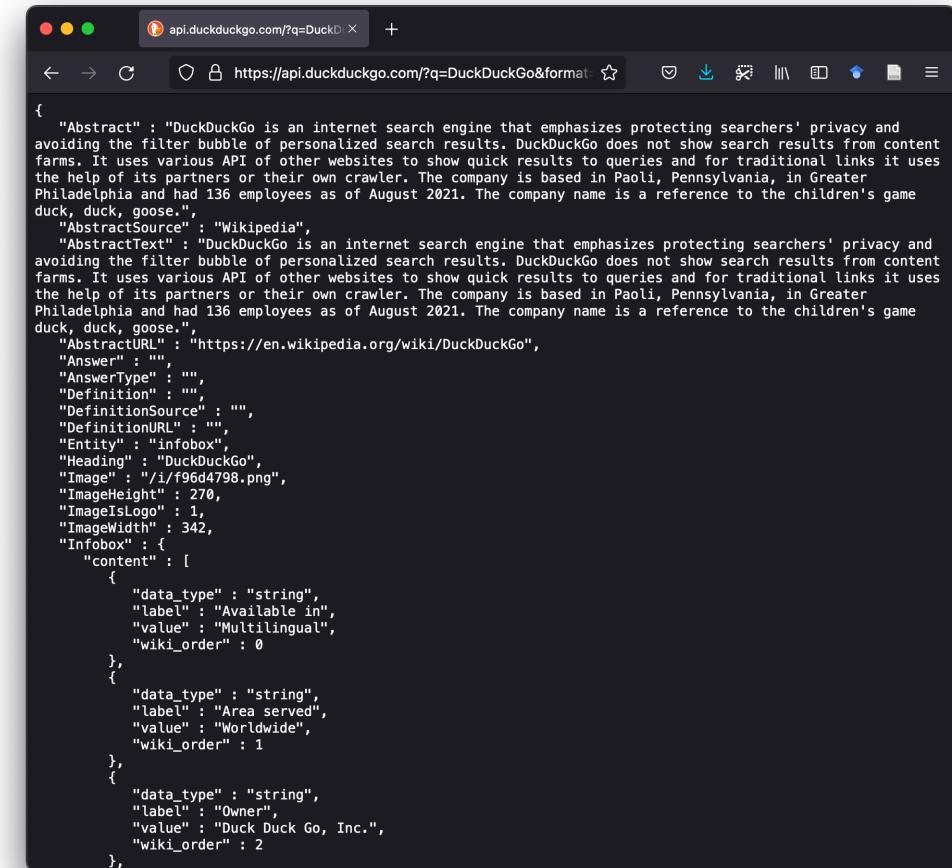


The screenshot shows a browser window displaying JSON data from the URL <https://dataverse.harvard.edu/api/datasets/export?exporter=data>. The JSON structure includes fields like id, identifier, persistentUrl, protocol, authority, publisher, publicationDate, storageIdentifier, datasetVersion, and various metadata blocks such as citation, fields, and files.

```
id: 3621057
identifier: "DVN/UATZBA"
persistentUrl: "https://doi.org/10.7910/DVN/UATZBA"
protocol: "doi"
authority: "10.7910"
publisher: "Harvard Dataverse"
publicationDate: "2019-11-06"
storageIdentifier: "s3://10.7910/DVN/UATZBA"
datasetVersion:
  id: 177357
  datasetId: 3621057
  datasetPersistentId: "doi:10.7910/DVN/UATZBA"
  storageIdentifier: "s3://10.7910/DVN/UATZBA"
  versionNumber: 1
  versionMinorNumber: 0
  versionState: "RELEASED"
  UNF: "UNF:6:a910QHwAcAc2fRrzy/oByQ=="
  lastUpdateTime: "2019-11-06T19:39:49Z"
  releaseTime: "2019-11-06T19:39:49Z"
  createTime: "2019-11-06T19:33:09Z"
  license: "CC0"
  termsOfUse: "CC0 Waiver"
  fileAccessRequest: false
metadataBlocks:
  citation:
    displayName: "Citation Metadata"
    fields: [...]
files:
  0: {...}
  1: {...}
citation:
  "Bail, Christopher, 2019, \"Assessing the Russian Internet Agency's Impact on the Political Attitudes and Behaviors of U.S. Twitter Users in Late 2017\" Proceedings of the National Academy of Sciences\", https://doi.org/10.7910/DVN/UATZBA, Harvard Dataverse, V1, UNF:6:a910QHwAcAc2fRrzy/oByQ== [fileUNF]"
```

JSON Applications and Sources

- Dynamic and ongoing data collection and distribution, without specified dimensions
- Used e.g., in communication between web applications and servers
- Many APIs provide output in JSON e.g., DuckDuckGo

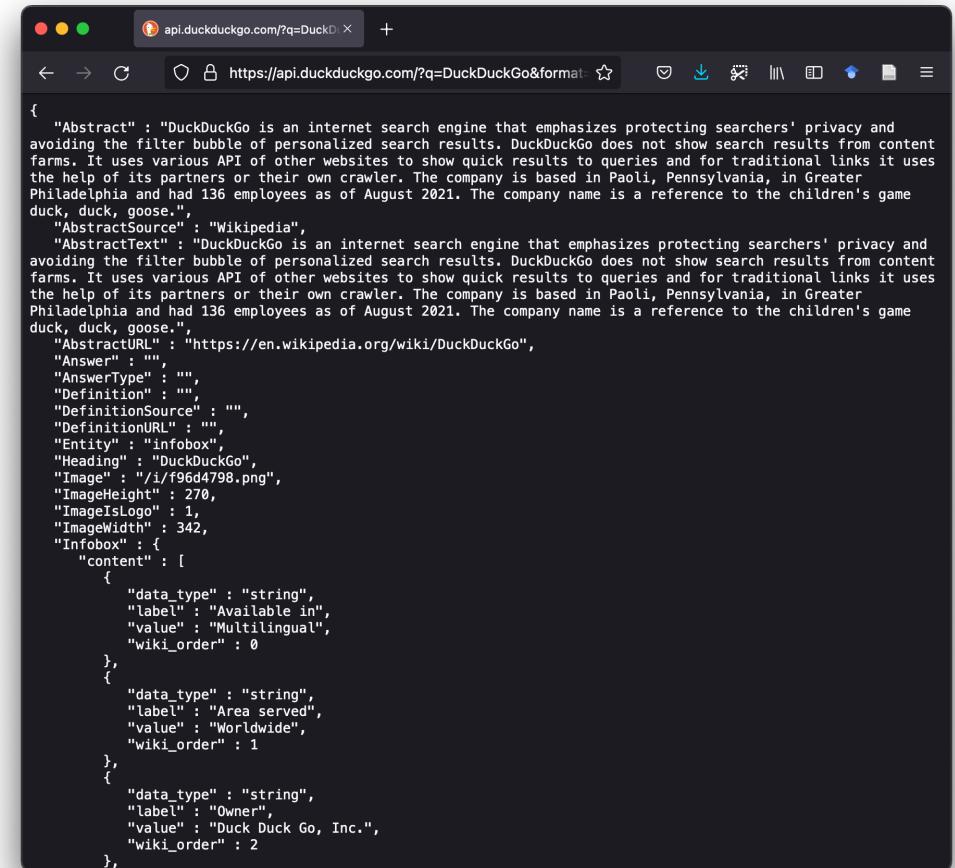


The screenshot shows a browser window with the URL <https://api.duckduckgo.com/?q=DuckDuckGo&format=json>. The page displays a JSON object representing information about DuckDuckGo. The JSON structure includes fields such as "Abstract", "AbstractSource", "AbstractText", "Answer", "AnswerType", "Definition", "DefinitionSource", "DefinitionURL", "Entity", "Heading", "Image", "ImageHeight", "ImageIsLogo", "ImageWidth", and "Infobox". The "Abstract" field contains a detailed description of the search engine. The "Infobox" field is an array containing three objects, each with "data_type", "label", "value", and "wiki_order" properties.

```
{
  "Abstract" : "DuckDuckGo is an internet search engine that emphasizes protecting searchers' privacy and avoiding the filter bubble of personalized search results. DuckDuckGo does not show search results from content farms. It uses various API of other websites to show quick results to queries and for traditional links it uses the help of its partners or their own crawler. The company is based in Paoli, Pennsylvania, in Greater Philadelphia and had 136 employees as of August 2021. The company name is a reference to the children's game duck, duck, goose.",
  "AbstractSource" : "Wikimedia",
  "AbstractText" : "DuckDuckGo is an internet search engine that emphasizes protecting searchers' privacy and avoiding the filter bubble of personalized search results. DuckDuckGo does not show search results from content farms. It uses various API of other websites to show quick results to queries and for traditional links it uses the help of its partners or their own crawler. The company is based in Paoli, Pennsylvania, in Greater Philadelphia and had 136 employees as of August 2021. The company name is a reference to the children's game duck, duck, goose.",
  "Answer" : "",
  "AnswerType" : "",
  "Definition" : "",
  "DefinitionSource" : "",
  "DefinitionURL" : "",
  "Entity" : "infobox",
  "Heading" : "DuckDuckGo",
  "Image" : "/i/96d4798.png",
  "ImageHeight" : 270,
  "ImageIsLogo" : 1,
  "ImageWidth" : 342,
  "Infobox" : [
    {
      "content" : [
        {
          "data_type" : "string",
          "label" : "Available in",
          "value" : "Multilingual",
          "wiki_order" : 0
        },
        {
          "data_type" : "string",
          "label" : "Area served",
          "value" : "Worldwide",
          "wiki_order" : 1
        },
        {
          "data_type" : "string",
          "label" : "Owner",
          "value" : "Duck Duck Go, Inc.",
          "wiki_order" : 2
        }
      ]
    }
  ]
}
```

JSON in Python

- Read in .json as plain text from local file or source
- Use json module to decode into Python object (`load`) and encode into json (`dump`)
- Access values using square brackets ([]) and names / attributes



The screenshot shows a browser window with the URL `https://api.duckduckgo.com/?q=DuckDuckGo&format=json`. The page displays a large block of JSON data. The JSON structure includes fields like "Abstract", "AbstractText", "AbstractSource", "AbstractURL", "Answer", "AnswerType", "Definition", "DefinitionSource", "DefinitionURL", "Entity", "Heading", "Image", "ImageHeight", "ImageIsLogo", "ImageWidth", and "Infobox". The "Infobox" field contains an array of objects, each with "data_type", "label", "value", and "wiki_order" properties.

```
{
  "Abstract" : "DuckDuckGo is an internet search engine that emphasizes protecting searchers' privacy and avoiding the filter bubble of personalized search results. DuckDuckGo does not show search results from content farms. It uses various API of other websites to show quick results to queries and for traditional links it uses the help of its partners or their own crawler. The company is based in Paoli, Pennsylvania, in Greater Philadelphia and had 136 employees as of August 2021. The company name is a reference to the children's game duck, duck, goose.",
  "AbstractSource" : "Wikipedia",
  "AbstractText" : "DuckDuckGo is an internet search engine that emphasizes protecting searchers' privacy and avoiding the filter bubble of personalized search results. DuckDuckGo does not show search results from content farms. It uses various API of other websites to show quick results to queries and for traditional links it uses the help of its partners or their own crawler. The company is based in Paoli, Pennsylvania, in Greater Philadelphia and had 136 employees as of August 2021. The company name is a reference to the children's game duck, duck, goose.",
  "AbstractURL" : "https://en.wikipedia.org/wiki/DuckDuckGo",
  "Answer" : "",
  "AnswerType" : "",
  "Definition" : "",
  "DefinitionSource" : "",
  "DefinitionURL" : "",
  "Entity" : "infobox",
  "Heading" : "DuckDuckGo",
  "Image" : "/i/96d4798.png",
  "ImageHeight" : 270,
  "ImageIsLogo" : 1,
  "ImageWidth" : 342,
  "Infobox" : {
    "content" : [
      {
        "data_type" : "string",
        "label" : "Available in",
        "value" : "Multilingual",
        "wiki_order" : 0
      },
      {
        "data_type" : "string",
        "label" : "Area served",
        "value" : "Worldwide",
        "wiki_order" : 1
      },
      {
        "data_type" : "string",
        "label" : "Owner",
        "value" : "Duck Duck Go, Inc.",
        "wiki_order" : 2
      }
    ]
  }
}
```

Digital Data Structures – Key Take-Homes

1. Non-tabular data formats offer flexibility for data storage and sharing, making them common formats for online data
2. HTML is the common format for web pages, using a hierarchical, tag-based structure to organize and format page content
3. CSS improves formatting work for markup documents, and offers benefits for online data collection
4. JSON is a non-tabular, language independent data format which uses name-value pairs for storage and is commonly used for data storage and sharing online

Digital Data Structures

Demo Script

Digital Data Structures – Demo

Let's look at this in practice

- Open “W38Mon-Demo-DigitalDataStructures.ipynb” and follow along

The screenshot shows a Jupyter Notebook interface running in a web browser. The title bar indicates the window is titled "W38Mon-Demo-DigitalDataStru x". The address bar shows the URL "localhost:8888/notebooks/SD". The main header bar includes the Jupyter logo, the notebook title "W38Mon-Demo-DigitalDataStructures", a Python 3 icon, and a "Logout" button. The menu bar has options: File, Edit, View, Insert, Cell, Kernel, Widgets, Help. Below the menu is a toolbar with various icons for file operations. The main content area displays a section titled "W38Mon: Digital Data Structures - Demo". The text in this section reads:

Welcome to the Demo Notebook for the SDS Base Camp session on digital data structures.

The Demo Notebook mirrors the structure of the lecture, so we will start with HTML and see how we can render it in Jupyter Notebooks and explore its structure and tags. Then we will explore one example of how to use Cascading Style Sheets in Python. We will close with an example of JSON and how to access data stored in JSON files.

1. HTML

As we discussed in the lecture, HTML is the common language used by web pages to structure and format the content they are presenting. This structure uses tags for formatting content and nesting to represent more complex structures. One of the really amazing things about Jupyter Notebooks and HTML is that...

Review Session I Preparation

- Please help us prepare for the first review session, by filling out this short survey on Absalon

bit.ly/3Cm5GuK

“First Review Survey” on Absalon

Exercise Preview

- Exploring HTML in the wild and in Jupyter Notebook
- Exploring CSS in the wild
- Working with JSON in Jupyter Notebook
- Putting it all together
- Daily reflections

The screenshot shows a Jupyter Notebook interface with the title bar "W38Mon-Exercise-DigitalDataStructures_sol". The notebook content area displays the following text:

Exercises W38Mon: Digital Data Structures

You fingers should now be itching for some new data to play with. And guess what? The Internet is full of it! Every **website** you can imagine can be **turned into a dataset** for analysis and you will now learn how to do this.

1. HTML

It all starts with learning a little about the language of the World Wide Web. **HTML is the markup language used by web pages**. It's ubiquitous on the web; even when editing this notebook you are interacting with HTML (right click and hit "View Page Source" if you need proof).

HTML consists of a series of elements that label pieces of content. Elements tell the browser how to display the content and are defined by a start tag followed by some content and an end tag: `<tagname> Content </tagname>`.

Start tags are enclosed in angled brackets `<>` while end tags have a forward slash between the angled brackets `</>`.