

Lecture 17

Hierarchical Data

1 What is Hierarchical Data?

2 JSON

3 XML

4 Recap

Hierarchical Data

Consider a data set of TV shows.

Each show has...

Hierarchical Data

Consider a data set of TV shows.

Each show has...

- a name
- a network
- ...

Hierarchical Data

Consider a data set of TV shows.

Each show has...

- a name
- a network
- ...
- multiple seasons,
each of which has...

Hierarchical Data

Consider a data set of TV shows.

Each show has...

- a name
- a network
- ...
- multiple seasons,
each of which has...
 - a number

Hierarchical Data

Consider a data set of TV shows.

Each show has...

- a name
- a network
- ...
- multiple seasons,
each of which has...
 - a number
 - a premiere date

Hierarchical Data

Consider a data set of TV shows.

Each show has...

- a name
- a network
- ...
- multiple seasons,
each of which has...
 - a number
 - a premiere date
 - an end date
 - ...

Hierarchical Data

Consider a data set of TV shows.

Each show has...

- a name
- a network
- ...
- multiple seasons,
each of which has...
 - a number
 - a premiere date
 - an end date
 - ...
 - multiple episodes,
each of which has...
 - a name
 - an airdate
 - ...

Hierarchical Data

Consider a data set of TV shows.

Each show has...

- a name
- a network
- ...
- multiple seasons,
each of which has...
 - a number
 - a premiere date
 - an end date
 - ...
 - multiple episodes,
each of which has...
 - a name
 - an airdate
 - ...
- a cast of multiple actors.

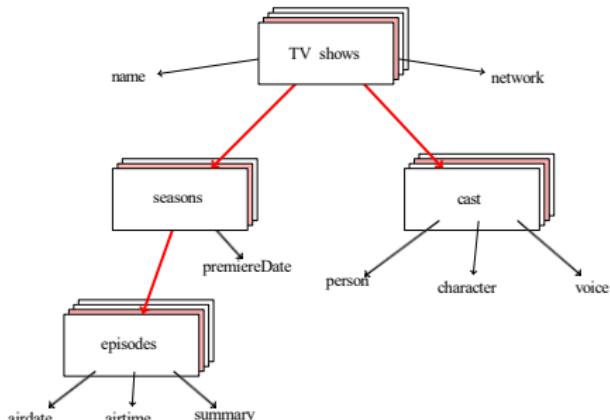
Hierarchical Data

Consider a data set of TV shows.

Each show has...

- a name
- a network
- ...
- multiple seasons, each of which has...
 - a number
 - a premiere date
 - an end date
 - ...
 - multiple episodes, each of which has...
 - a name
 - an airdate
 - ...
- a cast of multiple actors.

We can think of this data as a *tree*.



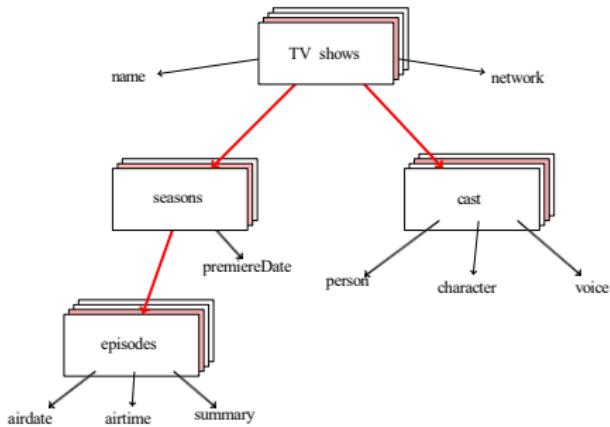
Hierarchical Data

Consider a data set of TV shows.

Each show has...

- a name
- a network
- ...
- multiple seasons, each of which has...
 - a number
 - a premiere date
 - an end date
 - ...
 - multiple episodes, each of which has...
 - a name
 - an airdate
 - ...
- a cast of multiple actors.

We can think of this data as a *tree*.



But how would we represent this data as a file on disk?

1 What is Hierarchical Data?

2 JSON

3 XML

4 Recap

JavaScript Object Notation (JSON)

JSON is one way to represent hierarchical data. In Python, JSON is represented as a `dict`.

JavaScript Object Notation (JSON)

JSON is one way to represent hierarchical data. In Python, JSON is represented as a `dict`.

```
[{'name': 'Girls',
 'network': {'name': 'NBC', ...},
 ...,
 'cast': [{('person': {'name': 'Lena Dunham', ...},
 'character': {'name': 'Hannah Horvath', ...},
 'voice': False},
 ...
 ],
 'seasons': [{('premiereDate': '2012-04-15',
 ...
 'episodes': [{('name': 'Pilot',
 'number': 1,
 'runtime': 30,
 ...
 ...
 }],
 ...
 }],
 },
 ...
 ]}
```

Working with JSON

Let's work with this JSON data in a Colab.



1 What is Hierarchical Data?

2 JSON

3 XML

4 Recap

eXtensible Markup Language (XML)

XML is another way to represent hierarchical data.

eXtensible Markup Language (XML)

XML is another way to represent hierarchical data.

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <show>
    <name>Girls</name>
    <network>
      <name>NBC</name>
      ...
    </network>
    <cast>
      <person>...</person>
      <character>....</character>
      <voice>...</voice>
    </cast>
    <cast>
      ...
    </cast>
    <season>
      <episode>...</episode>
      <episode>...</episode>
      ...
    </season>
    <season>
      ...
    </season>
  </show>
</root>
```

eXtensible Markup Language (XML)

XML is another way to represent hierarchical data.

- Fields are represented by named *tags*.

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <show>
    <name>Girls</name>
    <network>
      <name>NBC</name>
      ...
    </network>
    <cast>
      <person>...</person>
      <character>....</character>
      <voice>...</voice>
    </cast>
    <cast>
      ...
    </cast>
    <season>
      <episode>...</episode>
      <episode>...</episode>
      ...
    </season>
    <season>
      ...
    </season>
  </show>
</root>
```

eXtensible Markup Language (XML)

XML is another way to represent hierarchical data.

- Fields are represented by named *tags*.
- Each tag has an open `<tag>` and a close `</tag>`.

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <show>
    <name>Girls</name>
    <network>
      <name>NBC</name>
      ...
    </network>
    <cast>
      <person>...</person>
      <character>....</character>
      <voice>...</voice>
    </cast>
    <cast>
      ...
    </cast>
    <season>
      <episode>...</episode>
      <episode>...</episode>
      ...
    </season>
    <season>
      ...
    </season>
  </show>
</root>
```

eXtensible Markup Language (XML)

XML is another way to represent hierarchical data.

- Fields are represented by named *tags*.
- Each tag has an open `<tag>` and a close `</tag>`.
- Children are represented by nested tags.

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <show>
    <name>Girls</name>
    <network>
      <name>NBC</name>
      ...
    </network>
    <cast>
      <person>...</person>
      <character>....</character>
      <voice>...</voice>
    </cast>
    <cast>
      ...
    </cast>
    <season>
      <episode>...</episode>
      <episode>...</episode>
      ...
    </season>
    <season>
      ...
    </season>
  </show>
</root>
```

eXtensible Markup Language (XML)

XML is another way to represent hierarchical data.

- Fields are represented by named *tags*.
- Each tag has an open `<tag>` and a close `</tag>`.
- Children are represented by nested tags.
- Repeated fields are represented by repeated tags.

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <show>
    <name>Girls</name>
    <network>
      <name>NBC</name>
      ...
    </network>
    <cast>
      <person>...</person>
      <character>....</character>
      <voice>...</voice>
    </cast>
    <cast>
      ...
    </cast>
    <season>
      <episode>...</episode>
      <episode>...</episode>
      ...
    </season>
    <season>
      ...
    </season>
  </show>
</root>
```

eXtensible Markup Language (XML)

XML is another way to represent hierarchical data.

- Fields are represented by named *tags*.
- Each tag has an open `<tag>` and a close `</tag>`.
- Children are represented by nested tags.
- Repeated fields are represented by repeated tags.

Technical details:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <show>
    <name>Girls</name>
    <network>
      <name>NBC</name>
      ...
    </network>
    <cast>
      <person>...</person>
      <character>....</character>
      <voice>...</voice>
    </cast>
    <cast>
      ...
    </cast>
    <season>
      <episode>...</episode>
      <episode>...</episode>
      ...
    </season>
    <season>
      ...
    </season>
  </show>
</root>
```

eXtensible Markup Language (XML)

XML is another way to represent hierarchical data.

- Fields are represented by named *tags*.
- Each tag has an open `<tag>` and a close `</tag>`.
- Children are represented by nested tags.
- Repeated fields are represented by repeated tags.

Technical details:

- XML documents must begin with the declaration

```
<?xml ... ?>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <show>
    <name>Girls</name>
    <network>
      <name>NBC</name>
      ...
    </network>
    <cast>
      <person>...</person>
      <character>....</character>
      <voice>...</voice>
    </cast>
    <cast>
      ...
    </cast>
    <season>
      <episode>...</episode>
      <episode>...</episode>
      ...
    </season>
    <season>
      ...
    </season>
  </show>
</root>
```

eXtensible Markup Language (XML)

XML is another way to represent hierarchical data.

- Fields are represented by named *tags*.
- Each tag has an open `<tag>` and a close `</tag>`.
- Children are represented by nested tags.
- Repeated fields are represented by repeated tags.

Technical details:

- XML documents must begin with the declaration
`<?xml ... ?>`.
- XML documents must have a `<root>` tag.

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <show>
    <name>Girls</name>
    <network>
      <name>NBC</name>
      ...
    </network>
    <cast>
      <person>...</person>
      <character>....</character>
      <voice>...</voice>
    </cast>
    <cast>
      ...
    </cast>
    <season>
      <episode>...</episode>
      <episode>...</episode>
    ...
  </season>
  <season>
    ...
  </season>
  </show>
</root>
```

XML declaration.
Indicates that the file is in XML format.
version="1.0" → XML version
encoding="UTF-8" → character encoding

The **root** (top-level) element of the XML document.
In XML, all data must be contained within a single root element.

Working with XML

Let's work with the same data, as an XML, in a Colab.



1 What is Hierarchical Data?

2 JSON

3 XML

4 Recap

JSON vs. XML

Which is better?

- JSON has largely won over XML.

JSON vs. XML

- Which is better?
 - JSON has largely won over XML.
 - You can occasionally still find hierarchical data as XML, but it usually is JSON.

JSON vs. XML

- Which is better?
 - JSON has largely won over XML.
 - You can occasionally still find hierarchical data as XML, but it usually is JSON.
 - XML is still relevant in data science for one reason: it is a generalization of HTML, the language used to specify webpages.

JSON vs. XML

- Which is better?
 - JSON has largely won over XML.
 - You can occasionally still find hierarchical data as XML, but it usually is JSON.
 - XML is still relevant in data science for one reason: it is a generalization of HTML, the language used to specify webpages.

JSON vs. XML

Which is better?

- JSON has largely won over XML.
- You can occasionally still find hierarchical data as XML, but it usually is JSON.
- XML is still relevant in data science for one reason: it is a generalization of HTML, the language used to specify webpages.

We'll leverage our knowledge of XML next week to scrape webpages.