

[1350] JSON data

Nowadays, Internet technologies are being used more and more in place of traditional Client/Server technologies. Specifically, instead of using proprietary APIs and protocols to access databases, standard HTTP requests are being used. In addition, instead of hard-to-parse binary data being returned from databases, easier-to-use text formats like XML and JSON are being returned.

Initially, this revolution was led by XML - eXtensible Markup Language - which is similar to HTML but with more tags. Here is an example of an XML file:

```
<widget>
  <debug>on</debug>
  <window title="Sample Konfabulator Widget">
    <name>main_window</name>
    <width>500</width>
    <height>500</height>
  </window>
  <image src="Images/Sun.png" name="sun1">
    <hOffset>250</hOffset>
    <vOffset>250</vOffset>
    <alignment>center</alignment>
  </image>
  <text data="Click Here" size="36" style="bold">
    <name>text1</name>
    <hOffset>250</hOffset>
    <vOffset>100</vOffset>
    <alignment>center</alignment>
    <onMouseUp>
      sun1.opacity = (sun1.opacity / 100) * 90;
    </onMouseUp>
  </text>
</widget>
```

As you can see, XML is readable, but also somewhat verbose. Those extra closing tags get annoying after awhile. While XML is still used in a number of places, it is in the process of being gradually replaced with a easier-to-use, more compact text format called JSON.

JSON stands for "JavaScript Object Notation". Here's an example of a JSON file:

```
{ "menu": {
  "header": "SVG Viewer",
  "items": [
    { "id": "Open" },
    { "id": "OpenNew", "label": "Open New" },
    null,
    { "id": "ZoomIn", "label": "Zoom In" },
    { "id": "ZoomOut", "label": "Zoom Out" },
    { "id": "OriginalView", "label": "Original View" },
    null,
    { "id": "Quality" },
  ]
}
```

```
{
  "id": "Pause",
  "id": "Mute",
  null,
  {"id": "Find", "label": "Find..."},
  {"id": "FindAgain", "label": "Find Again"},
  {"id": "Copy"},
  {"id": "CopyAgain", "label": "Copy Again"},
  {"id": "CopySVG", "label": "Copy SVG"},
  {"id": "ViewSVG", "label": "View SVG"},
  {"id": "ViewSource", "label": "View Source"},
  {"id": "SaveAs", "label": "Save As"},
  null,
  {"id": "Help"},
  {"id": "About", "label": "About Adobe CVG Viewer..."}
]
```

This JSON file represents a single object called "menu" that has two fields, "header" and "items". "header" is a simple string field. "items" is an array of objects.

The item objects inside of the "items" array all have an "id" field. Some of them have other fields too. It is up to the program reading this file to decide what to do if a field is missing from an item object - but usually it is set to its default value (blank, zero, null, etc.).

Other Note:

- In JSON, fields are name/value pairs with the the name (in quotes), a colon character, and then the value (which may be a string, a number, an object or an array).
- In JSON, an Object is surrounded by curly braces { }
- An array of JSON items is surrounded by square braces [] and the items are separated by commas.
- JSON Data can - and should - be explored using <https://jsonlint.com/> [\(https://jsonlint.com/\)](https://jsonlint.com/)