Semi-Structured Data: JSON

School of Information Studies
Syracuse University

JSON

JavaScript Object Notation

Data interchange format

"Lightweight" format

- Data representations
- Easy for users to read
- Easy for parsers to translate

Main Structures

Object

- Unordered set of name/value pairs
- Uses outer {}
- Members separated by commas
- Each member—string

Array

- Ordered collection of values
- Uses outer []
- Values separated by commas

Value

- Object, array, string, number, true or false, null
- String—any Unicode character

Simple JSON Sample

```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    }
  ],
  "children": [],
  "spouse": null
}
```

Using JSON Objects

Twitter returns a JSON object

Use Python ‘json’ package

Converts to internal data structures

- Lists
- Dictionaries
- Can convert them back to strings

JSON Functions

Json.loads (jsonstring)

- Parses the JSON string

Json.dumps (python_object, sort_keys = True, indent=4)

- Does a ‘pretty print’
- Saves JSON data in a file