

# JSON

Author: Michelle Craig



UNIVERSITY OF  
TORONTO

# JSON

- light-weight data interchange format
- JavaScript Object Notation
- not JUST for JavaScript
- simple format

# Objects & Arrays

Object { name1: value1, name2: value2 }

Array [value1, value2, value3]

familiar look from Python

names are strings

values can be more complicated

Arrays, strings, numbers etc. are not Objects

# Values

strings

numbers

objects

arrays

true, false, null

```
{ "order": 435,  
  "customer": "Michelle",  
  "items": [{  
    "coffee": {  
      "fixings": ["cream", "sugar"],  
      "size": "small"}}], "timbit"],  
  "drive_thru": false}
```

```
{ "order": 435,  
  "customer": "Michelle",  
  "order_items": [{  
    "coffee": {  
      "fixings": ["cream", "sugar"],  
      "size": "small"}}], "timbit"],  
  "drive_thru": false}
```

```
{ "order": 435,  
  "customer": "Michelle",  
  "order_items": [{  
    "coffee": {  
      "fixings": ["cream", "sugar"],  
      "size": "small"}}, "timbit"],  
  "drive_thru": false}
```

```
{ "order": 435,  
  "customer": "Michelle",  
  "order_items": [{  
    "coffee": {  
      "fixings": ["cream", "sugar"],  
      "size": "small"}}, "timbit"],  
  "drive_thru": false}
```



# Why?

- simple to parse (programmatically)
- flexible
- human readable
- standard
- tools: lint and schema validators

<https://jsonlint.com>

# JSON Schema

- Applications often depend not only on getting valid JSON but on the JSON object having a particular format
- <https://json-schema.org>
- <https://jsonschemalint.com>
- Demo (schema for JSON Timmy's order)

```
"type": "object",
"properties": {
  "order": {
    "type": "number"
  },
  "customer": {
    "type": "string"
  },
  "order_items": {
    "type": "array",
    "items": {
      "anyOf": [{
        "$ref": "#/definitions/coffee"
      },
      {
        "type": "string"
      }
    ]
  },
  "drive_thru": {
    "type": "boolean"
  }
},
"required": ["order", "customer", "order_items"],
"additionalProperties": false
}
```

```
{ "description": "Tim's order example",  
  "definitions": {  
    "coffee": {  
      "type": "object",  
      "properties": {  
        "coffee": {  
          "type": "object",  
          "properties": {  
            "size": {  
              "type": "string",  
              "enum": ["small", "medium", "large"]  
            },  
            "fixings": {  
              "type": "array",  
              "items": {  
                "type": "string",  
                "enum": ["sugar", "cream", "milk"]  
              }  
            }  
          }  
        }  
      }  
    }  
  },  
}
```