JSON

JavaScript Object Notation

What is JSON

- Extended from the JavaScript
- Media type is application/json
- Extension is .json

Always starts and ends with curly brackets { }

Name and value is separated by a colon :

More than one pair is separated by comma ,

Basics of JSON

key/name value pairs

```
 { "name" : "value" }
```

Objects are comma separated

```
{ "name1" : "value" , "name2" : "value", "name3" : "value"}
```

Arrays have square brackets with values separated by comma

```
{ "name" : [ { "name" : "value" }, { "name" : "value" }] }
```

JSON LINT MAKES MORE READABLE

```
{
     "name": "value"
}

{
     "name1": "value",
     "name2": "value",
     "name3": "value"
}


{
     "name": [{
          "name": "value"
     }, {
          "name": "value"
     }]
}
```

Tool : https://jsonlint.com/

Data Structures

collection of name/value pairs : Think Object format

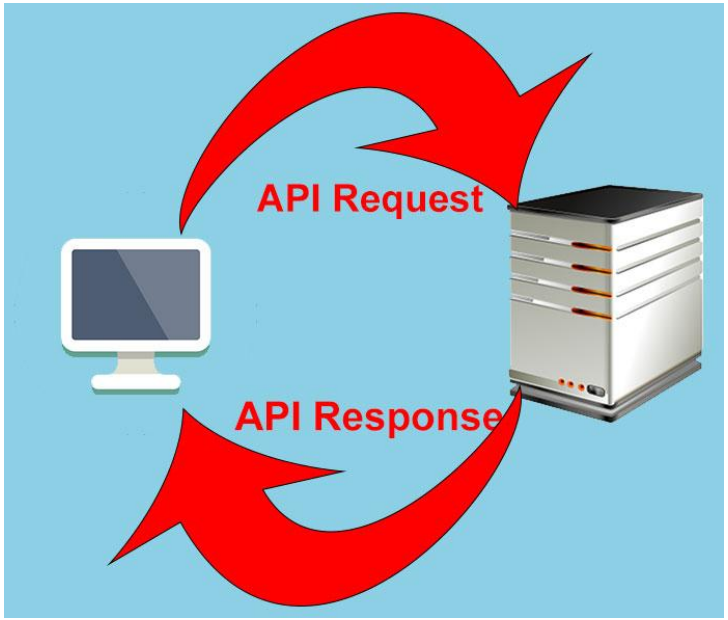
ordered list of values : Think Array format

Structured Data

As many levels of lists as needed to organize the data.

APIs communication between software components

APIs are made up of requests and responses



Data Types in JSON value can be any of these

Number - double- precision floating-point can be digits, positive or negative, decimal fractions, exponents… {"name":10}

String - double-quoted Unicode with backslash escaping {"name":"Hello world"}

Boolean - true or false {"name":true}

Array - ordered sequence of values uses square brackets. Values are each separated by a comma. Indexing starts with 0. {"name": [{"name1": 1}, "hello", "world"]}

Object - unordered collection with key:value pairs. Wrapped with curly brackets {}. Colons to separate key and name. Keys should be strings and have unique names. {"name": {"name1": 1,"name2": 1}}

Null - just empty {"name": null}

How to create an Object

JSON is an object which can be used to describe something

https://jsonlint.com/

https://jsonschema.net/

```
{
  "car1": "black",
  "car2": "blue"
}
```

Objects in JavaScript

Try this in the console

```
var myJSON = {};
myJSON.car1 = "black"
console.log(myJSON)
myJSON.car2 = "blue"
console.log(myJSON)

var myJSON = {};
myJSON["car1"] = "black"
console.log(myJSON)
myJSON["car2"] = "blue"
console.log(myJSON)

var myJSON = {"car1" : "black" ,"car2" : "blue"};
console.log(myJSON)
```

Add one more car3 with a color

Dot notation vs Bracket Notation

```
var myJSON = {}
myJSON.car1 = "black"
myJSON["car1"] = "blue"
console.log(myJSON)
```

Array of items

Better way

```
var cars = {"car":["Blue","black"]}
console.log(cars)
```

Now we can add more details to each item :)

```
var myJSON = {"car1" : {"color":"black"} ,"car2" : {"color" :"blue" }};
console.log(myJSON)
```

Even more details as much as we want!!!

```
var myJSON = {"car1" : {"color":"black", "model":"Mustang"} ,"car2" : {"color"
:"blue","model":"F150" }};
console.log(myJSON)
```

Examples of JSON Data for APIs

https://en.wikipedia.org/w/api.php?action=query&titles=Main%20Page&prop=revisions&rvprop=content&format=json&formatversion=2

https://developers.google.com/maps/documentation/geocoding/start?csw=1

http://maps.googleapis.com/maps/api/geocode/json?address=Toronto

Search APIs

https://apigee.com/console/

JSON vs XML vs YAML

JSON and XML are human readable formats JSON is faster to write. XML has not arrays. JSON much easier to parse in JavaScript

```json
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021"
  },
  "phoneNumber": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "fax",
      "number": "646 555-4567"
    }
  ],
  "gender": {
    "type": "male"
  }
}
```

```xml
<person>
  <firstName>John</firstName>
  <lastName>Smith</lastName>
  <age>25</age>
  <address>
    <streetAddress>21 2nd Street</streetAddress>
    <city>New York</city>
    <state>NY</state>
    <postalCode>10021</postalCode>
  </address>
  <phoneNumber>
    <type>home</type>
    <number>212 555-1234</number>
  </phoneNumber>
  <phoneNumber>
    <type>fax</type>
    <number>646 555-4567</number>
  </phoneNumber>
  <gender>
    <type>male</type>
  </gender>
</person>
```

```yaml
firstName: John
lastName: Smith
age: 25
address:
  streetAddress: 21 2nd Street
  city: New York
  state: NY
  postalCode: '10021'
phoneNumber:
- type: home
  number: 212 555-1234
- type: fax
  number: 646 555-4567
gender:
  type: male
```

JSON Schema

JSON Schema specifies a JSON-based format to define the structure of JSON data for validation, documentation, and interaction control. It provides a contract for the JSON data required by a given application, and how that data can be modified.

https://en.wikipedia.org/wiki/JSON

Tool https://jsonschema.net/

```
{
  "$schema": "http://json-schema.org/schema#",
  "title": "Product",
  "type": "object",
  "required": ["id", "name", "price"],
  "properties": {
    "id": {
      "type": "number",
      "description": "Product identifier"
    },
    "name": {
      "type": "string",
      "description": "Name of the product"
    },
    "price": {
      "type": "number",
      "minimum": 0
    },
    "tags": {
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "stock": {
      "type": "object",
      "properties": {
        "warehouse": {
          "type": "number"
        },
        "retail": {
```

Difference: JSON & JavaScript Object

JSON all *keys* must be quoted, object literals it is not necessary:

```
{ "foo": "bar" }
```

```
var o = { foo: "bar" };
```

JSON has double quotes while JavaScript can use single or doubles

JavaScript can include functions which is not available in JSON.