JSON

Website Development 2

JSON

- JavaScript Object Notation (JSON), is a minimal, readable format for structuring data. It is used primarily to transmit data between a server and web application, as an alternative to XML. It is easy for humans to read and write. It is easy for machines to parse and generate.
- It is a subset of JavaScript notation describing a data structure.

Why use JSON?

- **Standard Structure:** JSON objects have a standard structure that makes developers job easy to read and write code, because they know what to expect from JSON.
- **Light weight:** . Since JSON is light weighted, it becomes easier to get and load requested data quickly.
- **Scalable**: JSON is language independent, which means it can work well with most of the modern programming language. Let’s say if we need to change the server side language, in that case it would be easier for us to go ahead with that change as JSON structure is same for all the languages.

JSON Syntax

- JSON syntax is derived from JavaScript object notation syntax:
  - Data is in name/value pairs.
  - Data is separated by commas.
  - Curly braces hold objects.
  - Square brackets hold arrays.

JSON Data - A Name and a Value

- The two primary parts that make up JSON are keys and values. Together they make a key/value pair.
  - **Key:** A key is always a string enclosed in quotation marks.
  - **Value:** A value can be a string, number, boolean expression, array, or object.
  - **Key/Value Pair:** A key value pair follows a specific syntax, with the key followed by a colon followed by the value. Key/value pairs are comma separated. JSON data is written as name/value pairs.
- A name/value pair consists of a field name (in double quotes), followed by a colon, followed by a value:
  Example: "firstName":"Ray"

- A data item:
  - **title** - a string

- 2 data items separated by a comma:
  - **title** - a string
  - **artist** - a string

"title": "Piano Concerto No. 6",
"artist": "Beethoven"

- A song object, each containing:
  - **title** - a string
  - **artist** - a string


```
{
  "title": "Piano Concerto No. 6",
  "artist": "Beethoven"
}
```

- Three song objects, each containing:
  - **title** - a string
  - **artist -** a string

```
{
  "title": "Piano Concerto No. 0",
  "artist": "Beethoven"
},
{
  "title": "Piano Concerto No. 4",
  "artist": "Beethoven"
},
{
  "title": "Piano Concerto No. 6",
  "artist": "Beethoven"
}
```

- **songs** - an array of three song objects, each containing:
  - **title** - a string
  - **artist** - a string

```
"songs": [
  {
    "title": "Piano Concerto No. 0",
    "artist": "Beethoven"
  },
  {
    "title": "Piano Concerto No. 4",
    "artist": "Beethoven"
  },
  {
    "title": "Piano Concerto No. 6",
    "artist": "Beethoven"
  }
]
```

- A **playlist** object, containing:
  - **title** - a string
  - **songs** - an array of objects, each containing:
    - **title** - a string
    - **artist** - a string

```
{
  "title": "Beethoven Concertos",
  "songs": [
    {
      "title": "Piano Concerto No. 0",
      "artist": "Beethoven"
    },
    {
      "title": "Piano Concerto No. 4",
      "artist": "Beethoven"
    },
    {
      "title": "Piano Concerto No. 6",
      "artist": "Beethoven"
    }
  ]
}
```

- 2 **playlist** objects, each containing:
  - **title** - a string
  - **songs** - an array of objects, each containing:
    - **title** - a string
    - **artist** - a string


```
{
  "title": "Beethoven Sonatas",
  "songs": [
    {
      "title": "Piano Sonata No. 3",
      "artist": "Beethoven"
    },
    {
      "title": "Piano Sonata No. 7",
      "artist": "Beethoven"
    },
    {
      "title": "Piano Sonata No. 10",
      "artist": "Beethoven"
    }
  ]
},
{
  "title": "Beethoven Concertos",
  "songs": [
    {
      "title": "Piano Concerto No. 0",
      "artist": "Beethoven"
    },
    {
      "title": "Piano Concerto No. 4",
      "artist": "Beethoven"
    },
    {
      "title": "Piano Concerto No. 6",
      "artist": "Beethoven"
    }
  ]
}
```

- A **playlistCollection** object, containing:
  - an array of 2 **playlist** objects, each containing:
    - **title** - a string
    - **songs** - an array of objects, each containing:
      - **title** - a string
      - **artist** - a string


```
{
  "playlistCollection" : [
   {
    "title": "Beethoven Sonatas",
    "songs": [
     {
      "title": "Piano Sonata No. 3",
      "artist": "Beethoven"
     },
     {
      "title": "Piano Sonata No. 7",
      "artist": "Beethoven"
     },
     {
      "title": "Piano Sonata No. 10",
      "artist": "Beethoven"
     }
    ]
   },
   {
    "title": "Beethoven Concertos",
    "songs": [
     {
      "title": "Piano Concerto No. 0",
      "artist": "Beethoven"
     },
     {
      "title": "Piano Concerto No. 4",
      "artist": "Beethoven"
     },
     {
      "title": "Piano Concerto No. 6",
      "artist": "Beethoven"
     }
    ]
   }
  ]
}
```

Class Example - 1

For one person, we will record their name, image details, year they joined, their job role, where they live and how many friends they have.
So, we create an object for that person with Key/Value pairs as follows:

```
{
 "name":"Helen",
 "fullname":"Helen Hess",
 "image":"images/helen.jpg",
 "joined":2013,
 "job":"Art Director",
 "city":"New York",
 "mates":22
}
```

Class Example - 2

- For many people, we need an array of objects:

```
[
 {
  "name":"Stevie",
  "fullname":"Stevie Feliciano",
  "image":"images/stevie.jpg",
  "joined":2016,
  "job":"Web Developer",
  "city":"Boston",
  "mates":35
 },
 {
  "name":"Helen",
  "fullname":"Helen Hess",
  "image":"images/helen.jpg",
  "joined":2013,
  "job":"Art Director",
  "city":"New York",
  "mates":22
 },
 ...
]
```

Class Exercise - 1

- Write the code for a JSON object that contains the following information:
  - colour - red
  - hexvalue - #f00

Class Exercise - 2

- Expand the previous JSON object to now contain an array of colour objects as follows:
  - object: colour – red; hexvalue - #f00
  - object: colour – green; hexvalue - #0f0
  - object: colour – blue; hexvalue - #00f
  - object: colour – cyan; hexvalue - #0ff
  - object: colour – magenta; hexvalue - #f0f
  - object: colour – yellow; hexvalue - #ff0

Class Exercise - 3

- Write the code for a JSON object that contains the following information:
  - title - Dr. Zhivago
  - year - 1965
  - stars – array: Omar Sharif, Julie Cristie, Geraldine Chaplin
  - director – object: name: David Lean, DOB: 1908-03-25
  - rating - 8