

Foundational Technologies - JSON

JSON is an open standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and arrays

- JSON stands for JavaScript Object Notation
- It is a lightweight data-interchange format
- JSON is both human-readable and machine-parseable
- JSON is often used to transmit data between a server and a web application, as well as for configuration files and data storage.
- JSON data is represented as key-value pairs, where keys are strings enclosed in double quotes, followed by a colon, and then the associated value
- Keys are strings, and values can be strings, numbers, booleans, null, arrays, or other JSON objects
- JSON objects are unordered collections of key-value pairs enclosed in curly braces {}
- JSON arrays are enclosed in square brackets []
- Key-value pairs in JSON are separated by colons
- Items in JSON arrays are separated by commas
- It is widely supported by various programming languages
- JSON is often used for configuration files and data storage
- Common file extensions for JSON files include .json.
- JSON is language-independent, meaning it can be used with various programming languages. It was derived from JavaScript
- JSON is commonly used for representing structured data in web applications, including APIs (Application Programming Interfaces)
- Libraries and built-in functions are available in many programming languages to work with JSON data, making it easy to parse and generate JSON
- ECMAScript is a standard for scripting languages, including JavaScript, JScript, and ActionScript

JSON Is the format which is used within a JavaScript program in order to define the contents of an object

```
// object person
person={"firstname": "Thierry", "lastName" : "Henry"}
console.log(person.firstname)
// modify value firstname
person.firstname= "Samuel"
console.log(person.firstname)
```

STDIN

Input for t

Output:

Thierry
Samuel

JSON Data Structure

- Number
- String - sequence of unicode characters
- Boolean - True or False
- Array (to store an ordered collection of values) - arrays can contain JSON objects
- JSON Object - JSON objects can contain other JSON objects as values for their keys. This is known as nesting or embedding JSON objects within other JSON objects. This hierarchical structure allows for the representation of complex data relationships and structures
- Null - represents the absence of a value

```
{
  "exchange-rate":12.07,
  "age":57,
  "charge":-1.6e-19,
  "firstName":"Neil",
  "empty-string":"",
  "isTrainer":true,
  "phoneNumbers":["12345","34567","98765"],
  "address":{
    "street":"123 High Street",
    "city":"London",
    "postcode":"AB12 3FE"
  }
  "car":null
}
```

JSON Example

JSON representation describing a person

```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 27,
  "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": [
    "Catherine",
    "Thomas",
    "Trevor"
  ],
  "spouse": null
}
```

JSON Data Structure Example

- Some JSON Data files can start and end with [] (square brackets)
- Each element represents a single employee
- Element 0, element 1, etc
- JSON array of data values

```
[
  {
    "emp_no": 14397,
    "birth_date": "1957-09-15",
    "first_name": "Feixiong",
    "last_name": "Hardiono",
    "gender": "F",
    "hire_date": "1993-08-05"
  },
  {
    "emp_no": 50788,
    "birth_date": "1960-07-12",
    "first_name": "Zhiwei",
    "last_name": "Steinauer",
    "gender": "F",
    "hire_date": "1995-08-30"
  },
  {
    "emp_no": 30860,
    "birth_date": "1962-10-13",
    "first_name": "Karlis",
    "last_name": "Mapelli",
    "gender": "M",
    "hire_date": "1985-06-26"
  },
  {
    "emp_no": 13622,
    "birth_date": "1954-01-02",
    "first_name": "Tamiya",

```

XML

- XML stands for eXtensible Markup Language
- It is a markup language designed for storing and transporting data
- XML is a text-based format that is both human-readable and machine-readable
- It was initially developed to be a self-descriptive and platform-independent way to structure data
- XML documents consist of elements, attributes, and text content.
- Elements are enclosed in angle brackets (<>) and define the structure of the data.
- Elements can have child elements, forming a hierarchical structure
- Attributes provide additional information about elements and are contained within the start tag of an element
- Text content can be placed within elements to represent data
- XML documents must have a single root element that contains all other elements
- XML allows users to define their own document structure (via Document Type Definitions or XML Schemas)
- It is widely used for representing structured data in various domains, including web services, configuration files, and data exchange between applications
- XML is a generic format and not tied to any specific programming language.
- Common file extensions for XML files include .xml
- It's often used in combination with other technologies like XPath and XSLT for querying and transforming XML data
- XML has been largely replaced by JSON for web-based data interchange due to its simplicity and reduced verbosity, but it's still prevalent in certain contexts

Differences between JSON and XML

- Syntax: JSON uses a simple and compact syntax with key-value pairs, making it easy to read and write for both humans and machines. XML uses a more verbose and complex markup language with tags, attributes, and text content
- Readability: JSON is generally considered more readable due to its concise and straightforward syntax. XML can be less readable due to the additional markup, especially for complex documents.
- Data Types: JSON provides data type information such as strings, numbers, booleans, arrays, and null values. XML primarily deals with text data and requires additional parsing to handle different data types.

XML Example

```
<person>
  <firstName>John</firstName>
  <lastName>Smith</lastName>
  <isAlive>true</isAlive>
  <age>27</age>
  <address>
    <streetAddress>21 2nd Street</streetAddress>
    <city>New York</city>
    <state>NY</state>
    <postalCode>10021-3100</postalCode>
  </address>
  <phoneNumbers>
    <phoneNumber>
      <type>home</type>
      <number>212 555-1234</number>
    </phoneNumber>
    <phoneNumber>
      <type>office</type>
      <number>646 555-4567</number>
    </phoneNumber>
  </phoneNumbers>
  <children>
    <child>Catherine</child>
    <child>Thomas</child>
    <child>Trevor</child>
  </children>
  <spouse></spouse>
</person>
```

Unicode

Unicode is a standardised character encoding system that assigns a unique numeric code point to every character from various writing systems and languages worldwide

- Unicode is a global standard for character encoding
- Universal character set
- Common encoding schemes include UTF-8 and UTF-16
- Crucial for software localisation and globalisation.
- Includes a wide range of emoji character