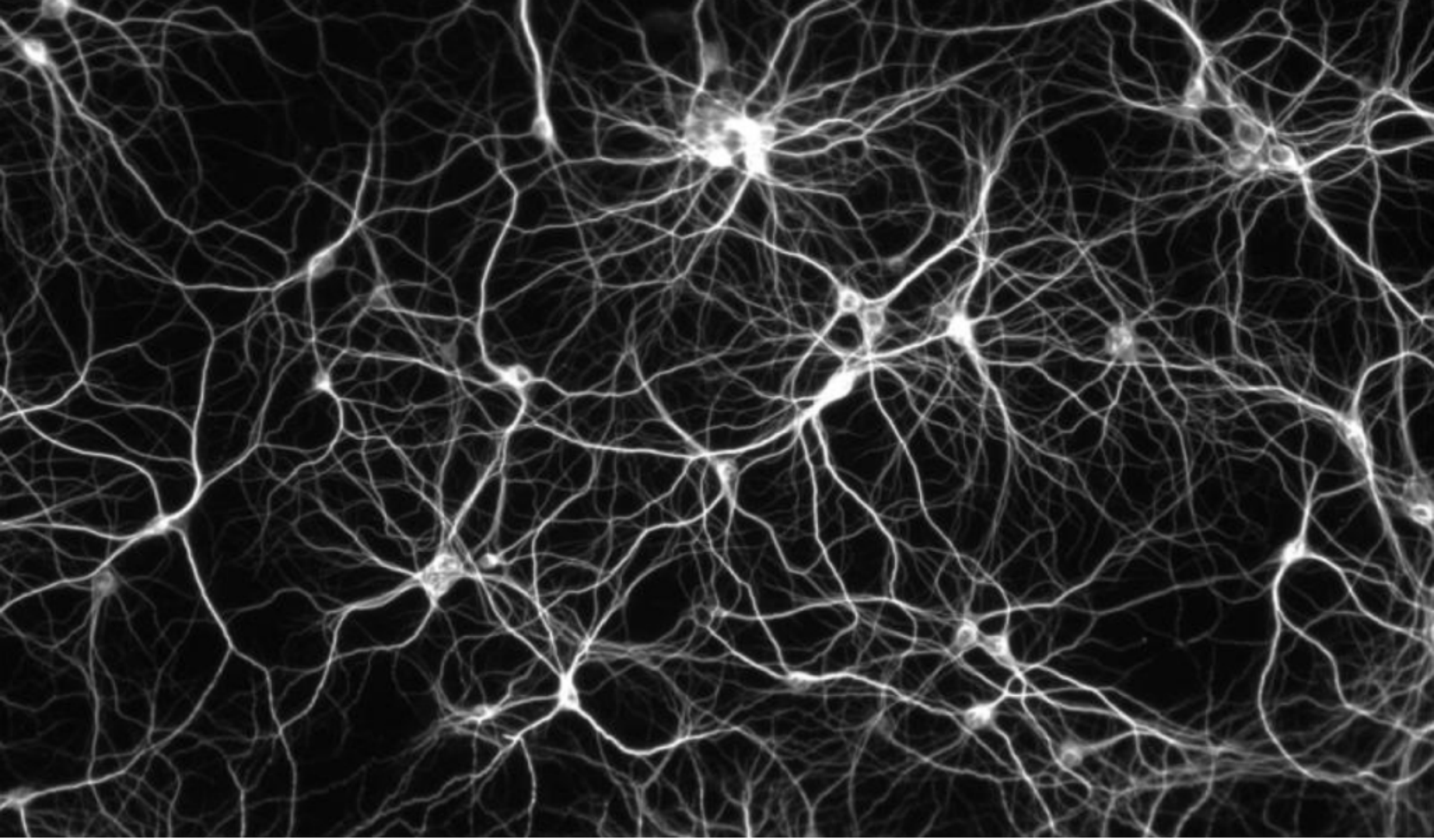


An Introduction to JSON
JavaScript Object Notation

Ahmed Muzammil | @ahmedmzl

What's inside

Introducing JSON
Why JSON
JSON Structures
Data in JSON
JSON Arrays, Objects
JSON Values, String, Number
Tools for Developers
JSON Data Example

JavaScript Object Notation

The fat free alternative to XML

Why JSON ??

- It's easy for humans to read and write
- It's also easy for computers to read and parse
- JSON's structure is significantly simple.
- Parsing efficiency is more when compared to XML
- Lighter and faster than XML as on-the-wire data format

JSON objects are "typed" vs. XML is "typeless"

XML

All Strings

JSON

string

number

array

boolean

strongly typed

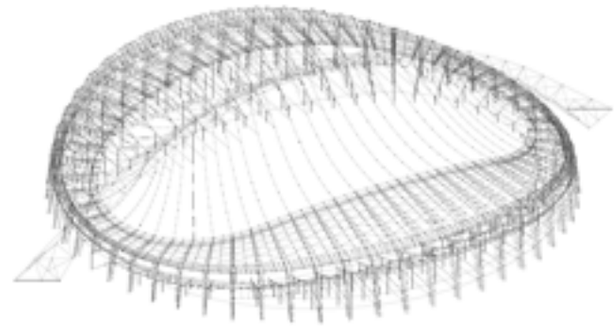
JSON | JavaScript Object Notation

JSON ➡ Native data form for JavaScript

- Data is readily accessible as JSON objects
- Retrieving values from JSON is as easy as importing an object in JavaScript

- XML data needs parsing
- Needs to use tedious DOM APIs and processing power to assign to variables

JSON is built on two structures



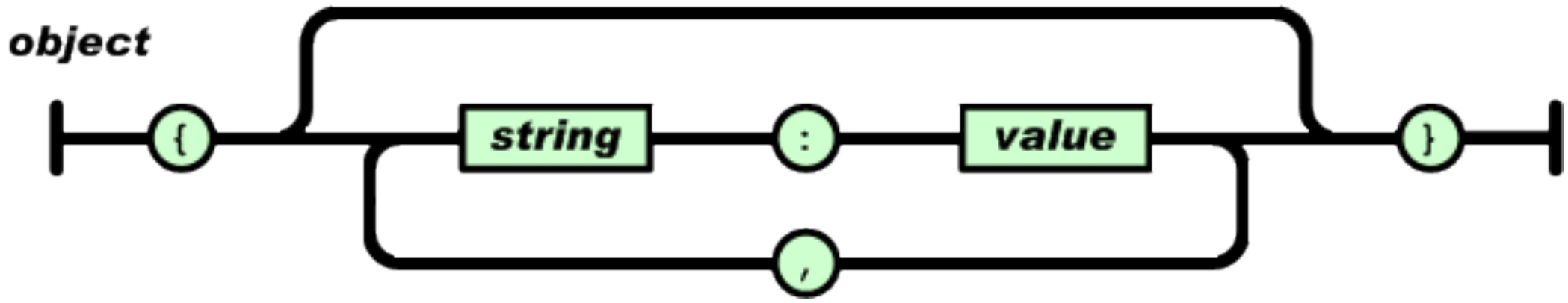
- A collection of name/value pairs
  - Realization (as in various programming languages)
    - Object, Record, Struct, Dictionary, HashTable, Keyed List, or Associative Array

- An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.
  - Realization (as in various programming languages
    - array, vector, list, or sequence

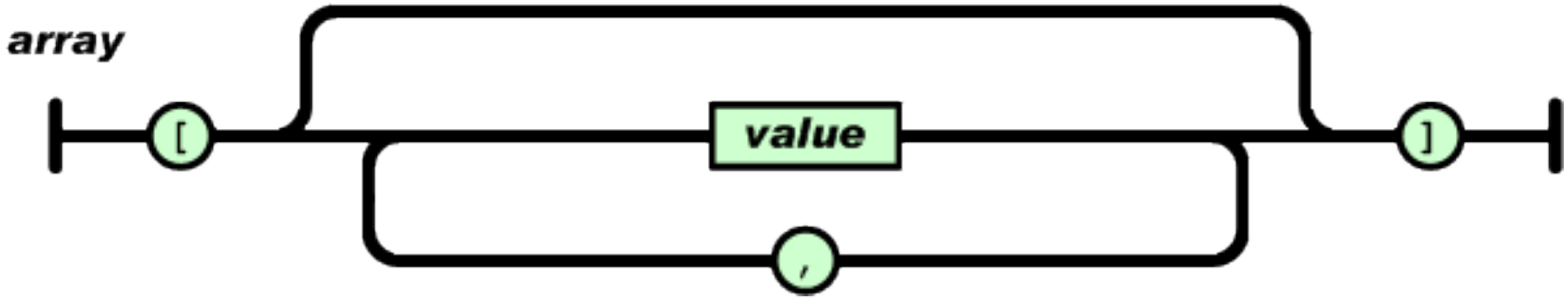
- JSON follows universal data structures
  - It is interoperable between programming languages

Data Storage In JSON

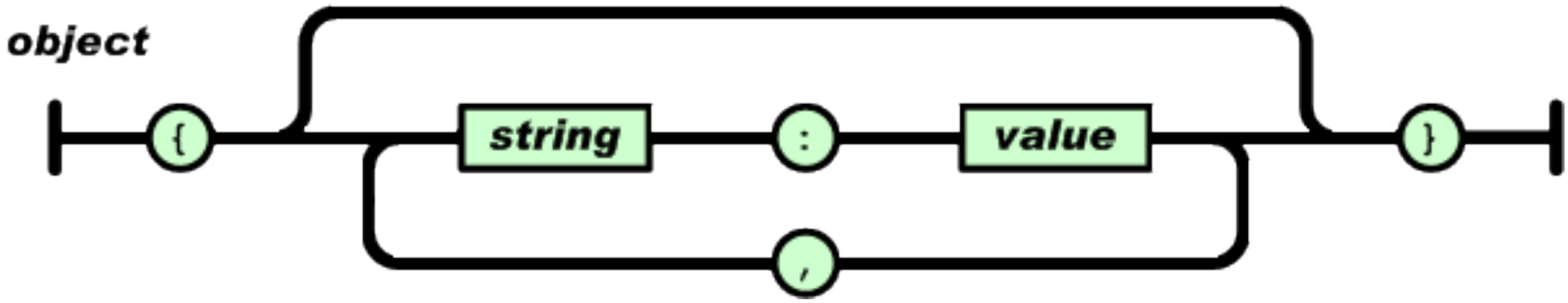
- Object



- Array



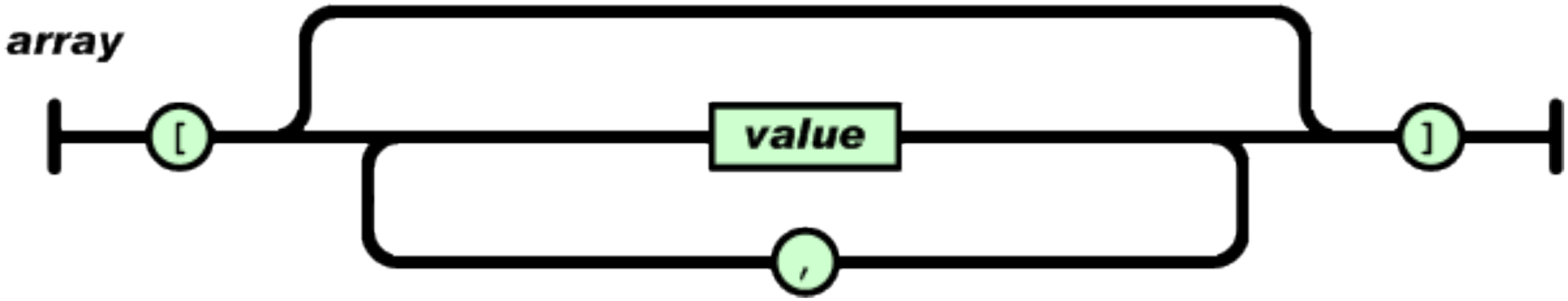
JSON :: Object



- **Example**

```
var myJSONObject =   { “Fruits” :
                                  [
                                  “Apple” ,
                                  “Orange”,
                                  “Grapes”
                                  ],
                     };
```

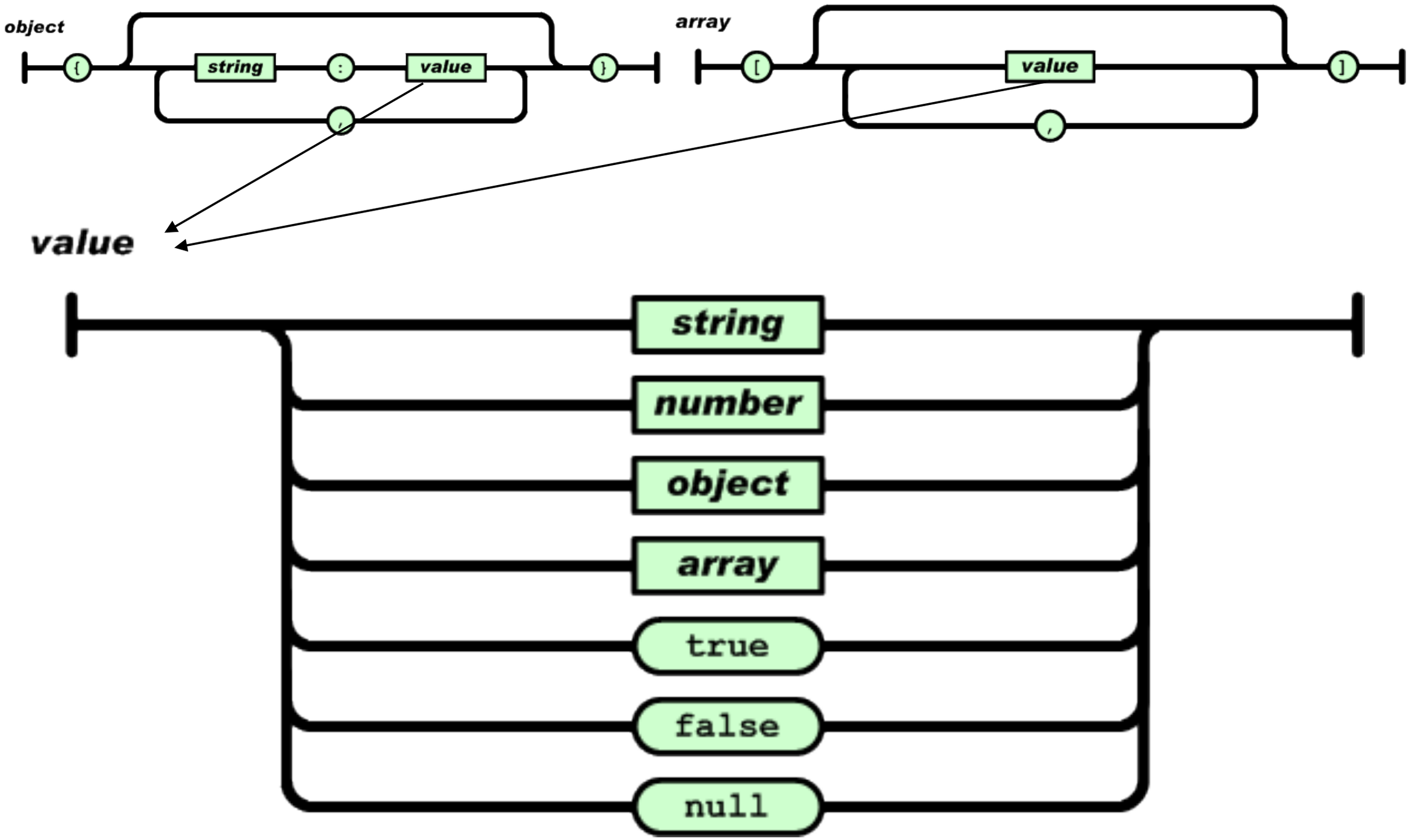
JSON :: Array



- **Example**

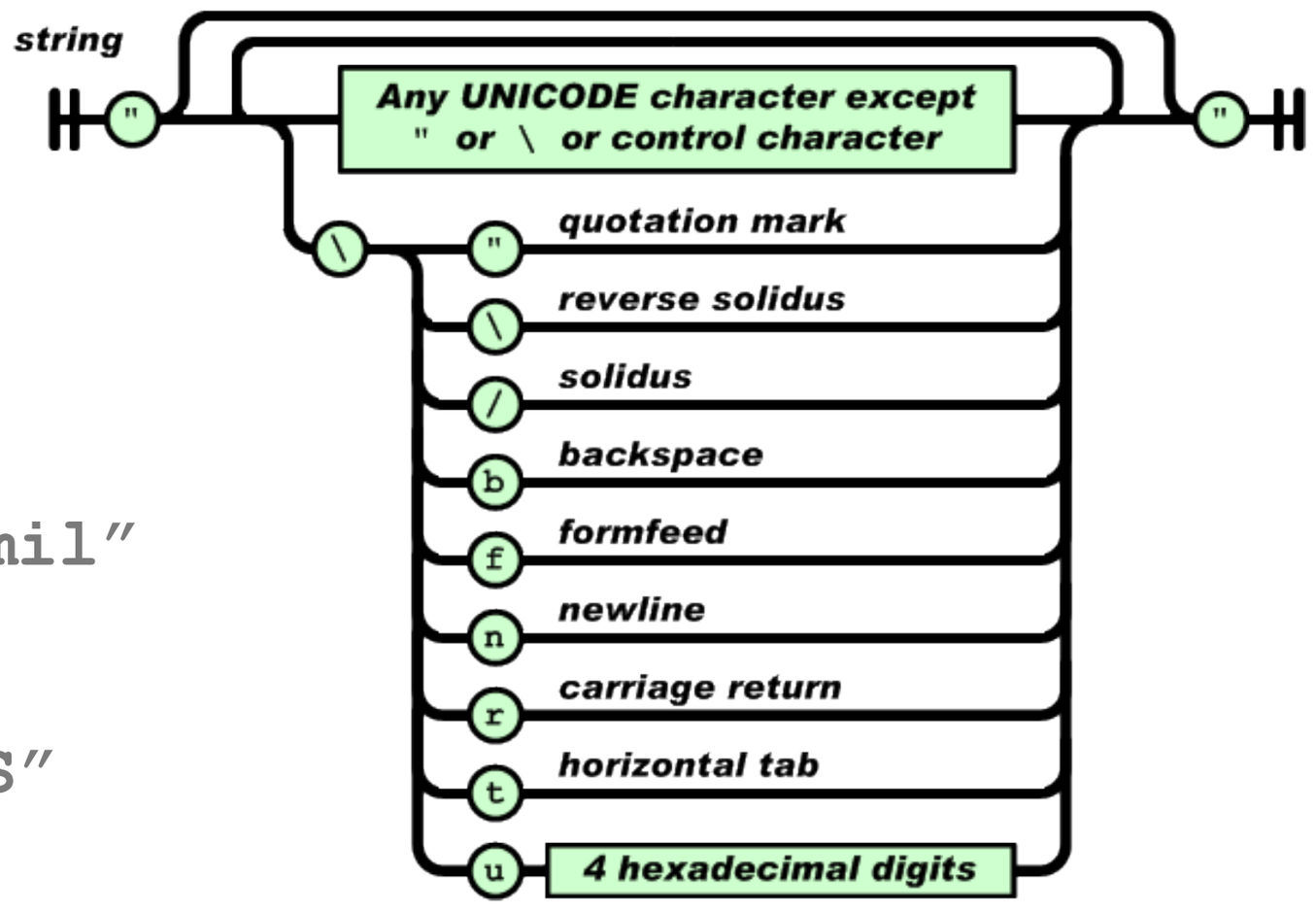
```
var myJSONArray = [
                    “Apple” ,
                    “Orange”,
                    “Grapes”
                  ];
```

JSON Values


object
{
string
:
value
}
,
array
[
value
]
,
value
string
number
object
array
true
false
null

JSON String

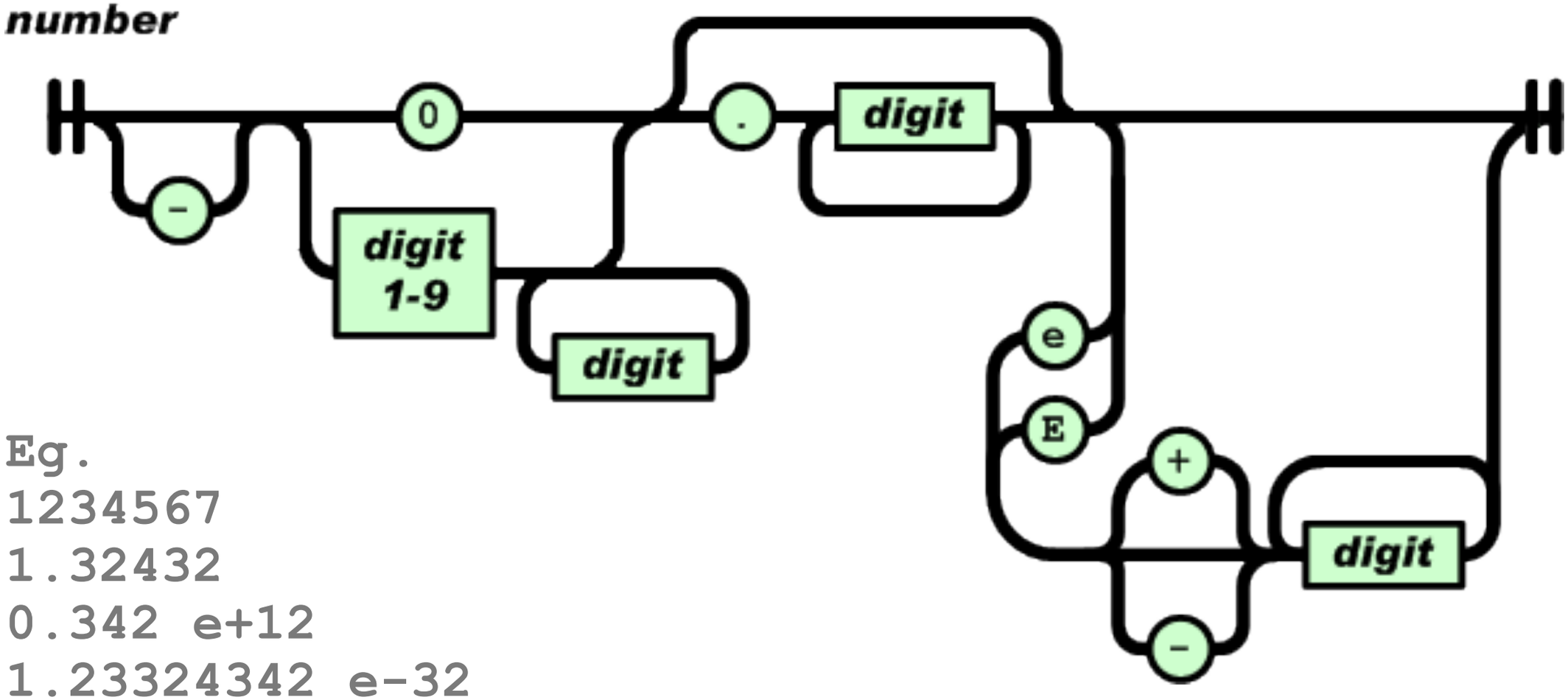
- A **string** is very much like a C or Java string.
  - **string** is a collection of zero or more Unicode characters within double quotes and using backslash escapes
  - **character** is represented as a **single character string**




```
Eg.
"Ahmed Muzammil"
"\"Jamal\""
"\nTest"
"\uAF34\u34DS"
```

JSON Number

- A number is very much like a C or Java number
  ! octal and hexadecimal formats are not used.




```
Eg.
1234567
1.32432
0.342 e+12
1.23324342 e-32
```

JSON Tools for Developers

- Available for download @ www.json.org

- Parser
  Parse JSON text files and convert these to a Java model

- Renderer
  Render a Java representation into text

- Serializer
  Serialize plain POJO clusters to a JSON representation

- Validator
  Validate the contents of a JSON file using a JSON schema

- Tools are also available for other languages
  - ASP, ActionScript, C Family, ColdFusion, Delphi, JavaScript, Perl, PHP, PL/SQL, Ruby, Symbian & many more

Where is JSON Used?

- Ajax applications / Web 2.0
- Represent configuration information
- Implement communication protocols
- Data Exchange
- Remote Procedure Call / RMI
- Service Oriented Architecture

Let’s end with an example 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“
                  }
              ]
}
```

The same Example in XML

```
<Object>
<Property><Key>firstName</Key> <String>John</String></Property>
<Property><Key>lastName</Key> <String>Smith</String></Property>
<Property><Key>age</Key> <Number>25</Number></Property>
<Property><Key>address</Key> <Object> <Property><Key>streetAddress</Key>
<String>21 2nd Street</String></Property>
<Property><Key>city</Key> <String>New York</String></Property>
<Property><Key>state</Key> <String>NY</String></Property>
<Property><Key>postalCode</Key> <String>10021</String></Property>
</Object>
</Property> <Property><Key>phoneNumber</Key>
<Array> <Object> <Property><Key>type</Key> <String>home</String></Property>
<Property><Key>number</Key> <String>212 555-1234</String></Property></Object>
<Object>
<Property><Key>type</Key> <String>fax</String></Property> <Property><Key>number</
Key> <String>646 555-4567</String></Property> </Object> </Array>
</Property>
</Object>
```

Want more of JSON ? Go here...

- Introducing JSON
  - http://www.json.org/

- Introduction to JSON
  - http://www.javapassion.com/ajax/JSON.pdf

- JSON in JavaScript
  - http://www.json.org/js.html

- JSON in Java
  - http://www.json.org/java/index.html

JSON is GOOD !

- Like this presentation? Share it...

- Questions?
  - Tweet me @ahmedmzl

Buy a "I ♥ JSON Tshirt" like this here at zazzle.com

- I can help you on:
  - Service Oriented Architecture (SOA)
  - Business Process Management (BPM)
  - Business Intelligence (BI)
  - User Experience (UX)
  - Product Development
  - Project Management