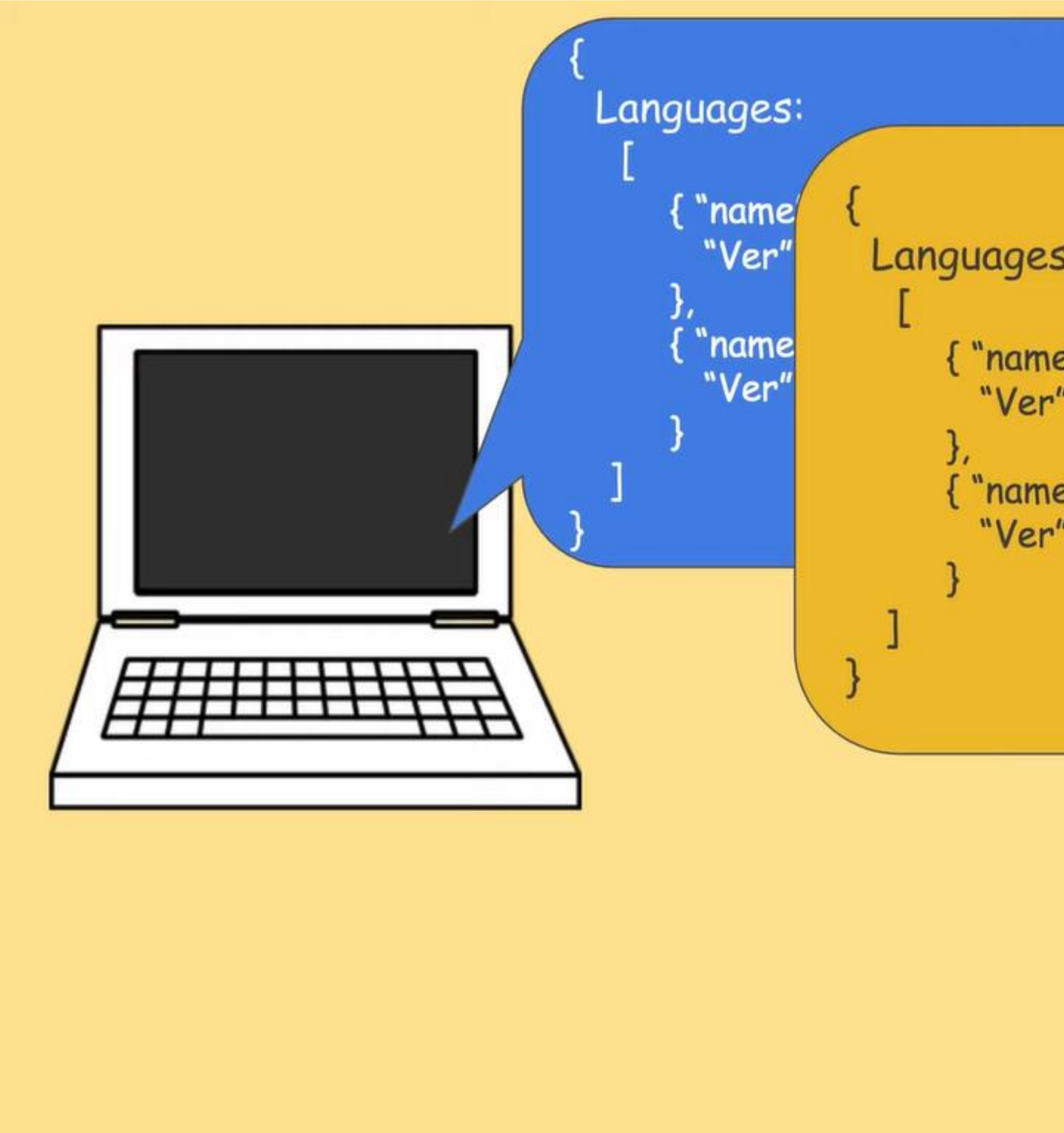

{
Languages:
[
{ "name
"Ver"
},
{ "name
"Ver"
}
]
}
{
Languages
[
{ "name
"Ver"
},
{ "name
"Ver"
}
]
}

The format of communication we

JSO

JavaScript Objec

JSON is a way of communicating data wit

Using **Key Value** pairs

The syntax is taken from JavaScript but JS

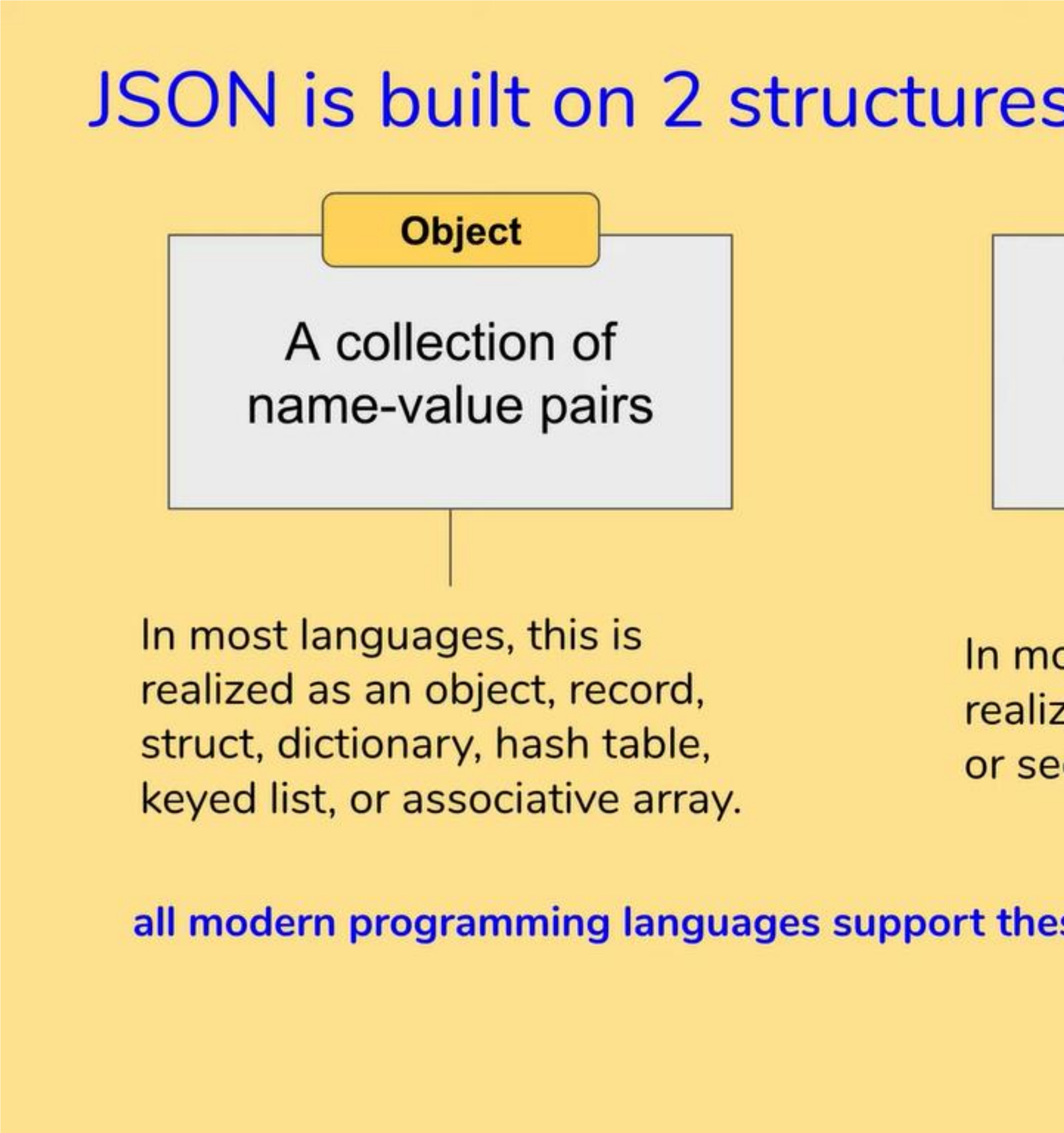
from JSON.org

JSON is a lightweight data-intercha

It is easy for humans to read and w

It is easy for machines to parse and

It is based on a subset of the JavaS
Standard ECMA-262 3rd Edition -



JSON is built on 2 structures

Object

A collection of name-value pairs

In most languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.

In mo
realiz
or se

all modern programming languages support thes

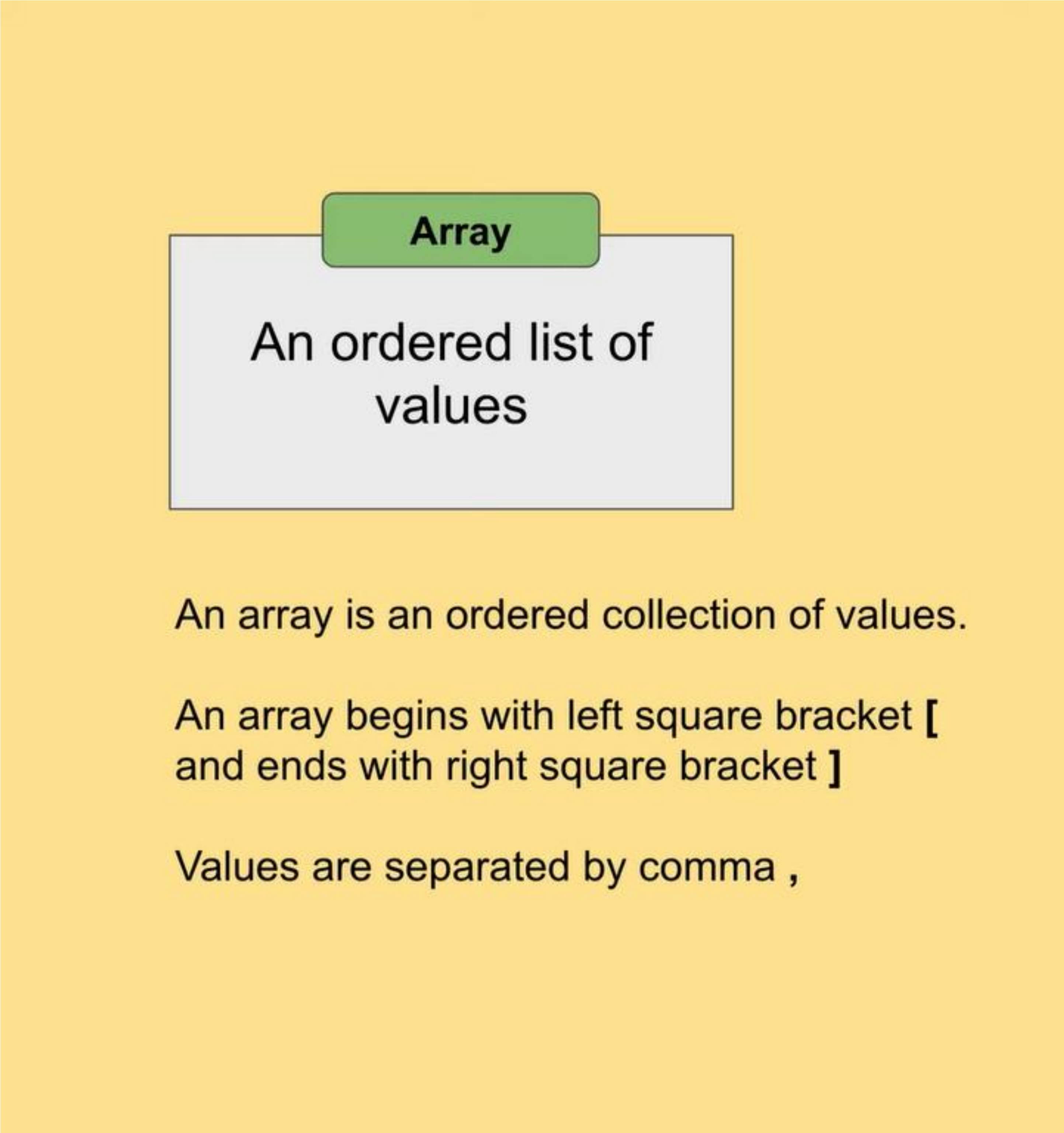
Object

A collection of name-value pairs

An object is an unordered set of name/value pairs

An object begins with left curly brace **{**
and ends with right curly brace **}**

Each name is followed by colon **:**
and the name/value pairs are separated by comma **,**



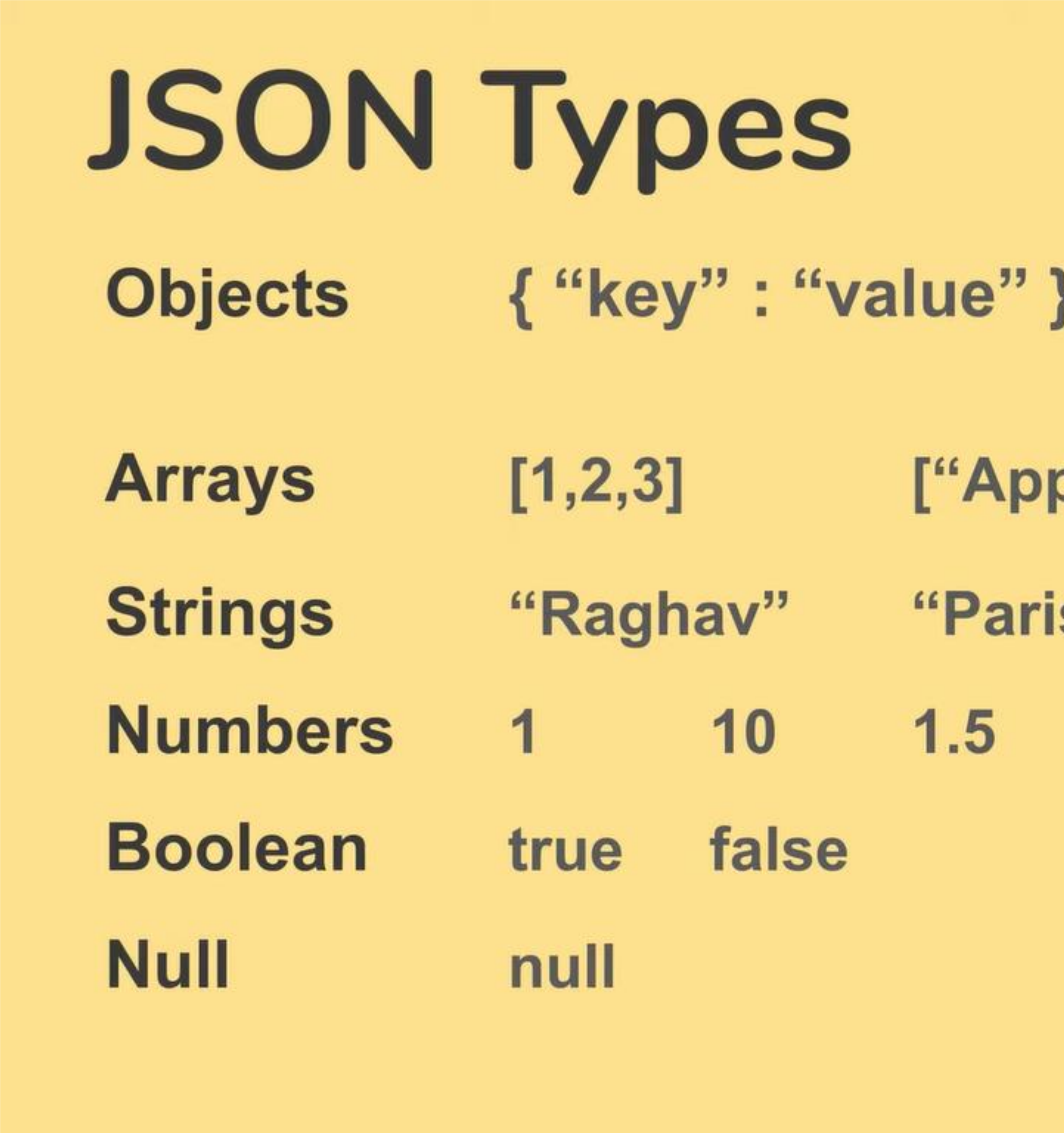
Array

An ordered list of values

An array is an ordered collection of values.

An array begins with left square bracket **[** and ends with right square bracket **]**

Values are separated by comma **,**



JSON Types

Objects	{ “key” : “value” }		
Arrays	[1,2,3]		[“App
Strings	“Raghav”		“Pari
Numbers	1	10	1.5
Boolean	true	false	
Null	null		

JSON Syntax Rules

- Data is in name/value pairs
- Data is separated by comm
- Curly braces hold objects
- Square brackets hold array

Example 1

```
{
"users":
[
{
        "firstName":"Ram",
        "lastName":"Thapa",
        "age":20
},
{
"firstName":"Shyam",
"lastName":"shrestha",
"age":40
"hobbies":["reading","writing"],
"friends":
[
{
"firstName":"Hari",
"lastName":"Maharjan"
}
]
],
"subject":
[
{

"name":"Rita",

"id":1

},

{

"name":"sita",

"id":2

}

]

}
```

Both JSON and XML can be used to receive data from a web server.

The following JSON and XML examples both define an employees object, with an array of 3 employees:

JSON Example

```
{"employees":[
  { "firstName":"John", "lastName":"Doe" },
  { "firstName":"Anna", "lastName":"Smith" },
  { "firstName":"Peter", "lastName":"Jones" }
]}
```

XML Example

```
<employees>
  <employee>
    <firstName>John</firstName> <lastName>Doe</lastName>
  </employee>
  <employee>
    <firstName>Anna</firstName> <lastName>Smith</lastName>
  </employee>
  <employee>
    <firstName>Peter</firstName> <lastName>Jones</lastName>
  </employee>
</employees>
```

JSON DATA

```
{
 "menu": {
   "id": "file",
    "value":"File",
     "popup": {
       "menuitem":[
        {"value":"New","onclick":"CreateNewDoc()"},
         {"value":"Open","onclick":"openDoc()"},
         {"value":"Close","onclick":"CloseDoc()"}
         ]
}
       } }
```

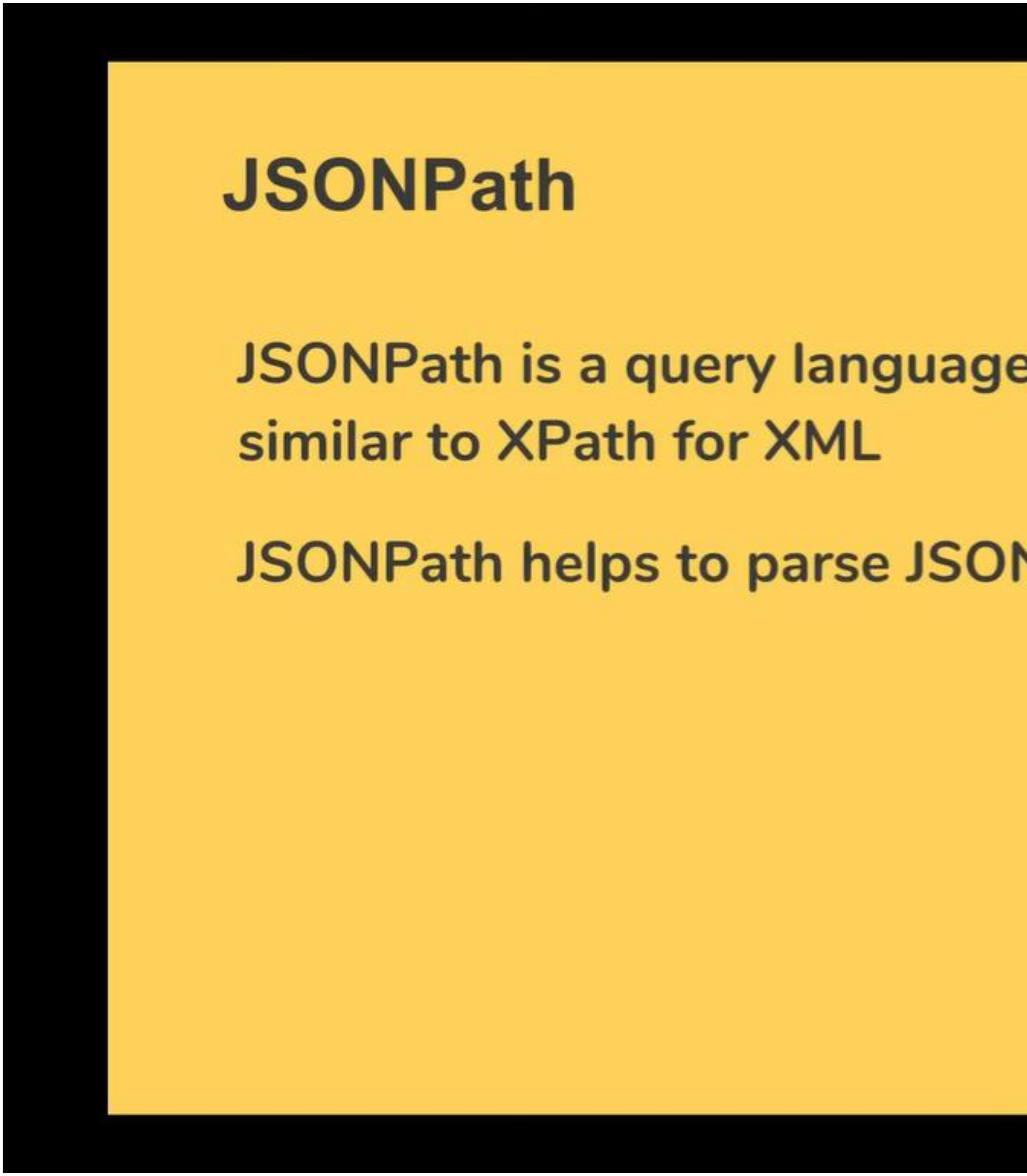
Its Equivalence XML format

```
<?xml version="1.0" encoding="UTF-8" ?>
        <menu>
                <id>file</id>
                <value>File</value>
                <popup>
                        <menuitem>
                                <value>New</value>
                                <onclick>CreateNewDoc()</onclick>
                        </menuitem>
                        <menuitem>
                                <value>Open</value>
                                <onclick>openDoc()</onclick>
                        </menuitem>
                        <menuitem>
                                <value>Close</value>
                                <onclick>CloseDoc()</onclick>
                        </menuitem>
                </popup>
        </menu>
```

How to validate json

Jsonlint.com
Jsonschemavalidator
Freeformatter.com
Jsoonformatter.curiousconcept.com
Json beautify

Codebeautify.org/jsonviewer



JSONPath

JSONPath is a query language
similar to XPath for XML

JSONPath helps to parse JSON

How to Find Json Path

1. Jsonpath.com
2. Jsonpathfinder.com
3.