

# JSON

step-by-step



# The JSON object



john.json

```
{  
  "name": "John",  
  "age": 27,  
  "employed": true,  
  "hasCar": null,  
  "parents": ["Anna", "Michael"],  
  "pet": {  
    "name": "Brutus",  
    "species": "dog",  
    "age": 7  
  }  
}
```

# The JSON object



 john.json

```
{  
  "name": "John",  
  "age": 27,  
  "employed": true,  
  "hasCar": null,  
  "parents": ["Anna", "Michael"],  
  "pet": {  
    "name": "Brutus",  
    "species": "dog",  
    "age": 7  
  }  
}
```

Objects are  
enclosed in curly  
braces

# The JSON object



 john.json

```
{  
  "name": "John",  
  "age": 27,  
  "employed": true,  
  "hasCar": null,  
  "parents": ["Anna", "Michael"],  
  "pet": {  
    "name": "Brutus",  
    "species": "dog",  
    "age": 7  
  }  
}
```

Information is  
stored in  
"key": "value"  
pairs

# The JSON object



john.json

```
{  
  "name": "John",  
  "age": 27,  
  "employed": true,  
  "hasCar": null,  
  "parents": ["Anna", "Michael"],  
  "pet": {  
    "name": "Brutus",  
    "species": "dog",  
    "age": 7  
  }  
}
```

Information is  
stored in  
"key": "value"  
pairs

**keys** are of  
datatype **string**

# The JSON object



john.json

```
{  
  "name": "John",  
  "age": 27,  
  "employed": true,  
  "hasCar": null,  
  "parents": ["Anna", "Michael"],  
  "pet": {  
    "name": "Brutus",  
    "species": "dog",  
    "age": 7  
  }  
}
```

values **must be** one  
of the following  
data types:

- string
- number
- boolean
- null
- array
- object

# The JSON object



john.json

```
{
  "name": "John",
  "age": 27,
  "employed": true,
  "hasCar": null,
  "parents": ["Anna", "Michael"],
  "pet": {
    "name": "Brutus",
    "species": "dog",
    "age": 7
  }
}
```

values **must be** one of the following data types:

- string
- number
- boolean
- null
- array
- object

# The JSON object



john.json



**strings** are any kind of characters enclosed in “ “

- “word”
- “This is also a string.”
- “7 bananas”

```
{  
  "name": "John",  
  "age": 27,  
  "employed": true,  
  "hasCar": null,  
  "parents": ["Anna", "Michael"],  
  "pet": {  
    "name": "Brutus",  
    "species": "dog",  
    "age": 7  
  }  
}
```

boolean  
null  
array  
object



# The JSON object



john.json

```
{  
  "name": "John",  
  "age": 27,  
  "employed": true,  
  "hasCar": null,  
  "parents": ["Anna", "Michael"],  
  "pet": {  
    "name": "Brutus",  
    "species": "dog",  
    "age": 7  
  }  
}
```

values **must be** one  
of the following  
data types:

- string
- number**
- boolean
- null
- array
- object

# The JSON object



john.json



**numbers** can be:

- integers (e.g. 42)
- floats (e.g. 0.0005)

```
{  
  "name": "John",  
  "age": 27,  
  "employed": true,  
  "hasCar": null,  
  "parents": ["Anna", "Michael"],  
  "pet": {  
    "name": "Brutus",  
    "species": "dog",  
    "age": 7  
  }  
}
```

boolean  
null  
array  
object

# The JSON object



john.json

```
{  
  "name": "John",  
  "age": 27,  
  "employed": true,  
  "hasCar": null,  
  "parents": ["Anna", "Michael"],  
  "pet": {  
    "name": "Brutus",  
    "species": "dog",  
    "age": 7  
  }  
}
```

values **must be** one  
of the following  
data types:

- string
- number
- boolean**
- null
- array
- object

# The JSON object



john.json



a **boolean** has one of two possible values

- true / false
- 1 / 0

```
{  
  "name": "John",  
  "age": 27,  
  "employed": true,  
  "hasCar": null,  
  "parents": ["Anna", "Michael"],  
  "pet": {  
    "name": "Brutus",  
    "species": "dog",  
    "age": 7  
  }  
}
```

boolean  
null  
array  
object

# The JSON object



john.json

```
{
  "name": "John",
  "age": 27,
  "employed": true,
  "hasCar": null,
  "parents": ["Anna", "Michael"],
  "pet": {
    "name": "Brutus",
    "species": "dog",
    "age": 7
  }
}
```

values **must be** one  
of the following  
data types:

- string
- number
- boolean
- null**
- array
- object

# The JSON object



john.json



**null** can only have the value NULL.  
The variable of data type **null** has  
**no value assigned** to it.

```
{  
  "name": "John",  
  "age": 27,  
  "employed": true,  
  "hasCar": null,  
  "parents": ["Anna", "Michael"],  
  "pet": {  
    "name": "Brutus",  
    "species": "dog",  
    "age": 7  
  }  
}
```

boolean  
**null**  
array  
object

# The JSON object



john.json

```
{
  "name": "John",
  "age": 27,
  "employed": true,
  "hasCar": null,
  "parents": ["Anna", "Michael"],
  "pet": {
    "name": "Brutus",
    "species": "dog",
    "age": 7
  }
}
```

values **must be** one  
of the following  
data types:

- string
- number
- boolean
- null
- array**
- object

# The JSON object



john.json



An **array** is a collection of elements.  
Can be understood as a list.

- ["Bibi", "Tina"]
- [1,2,3]
- ["some string", 0.5, true]

```
{
  "name": "John",
  "age": 27,
  "employed": true,
  "hasCar": null,
  "parents": ["Anna", "Michael"],
  "pet": {
    "name": "Brutus",
    "species": "dog",
    "age": 7
  }
}
```

boolean  
null  
**array**  
object



# The JSON object



john.json

```
{
  "name": "John",
  "age": 27,
  "employed": true,
  "hasCar": null,
  "parents": ["Anna", "Michael"],
  "pet": {
    "name": "Brutus",
    "species": "dog",
    "age": 7
  }
}
```

values **must be** one of the following data types:

- string
- number
- boolean
- null
- array
- object**

# The JSON object



john.json



An **object** contains **key/value pairs**,  
**seperated by commata** and is  
enclosed by **{ }**

```
{ "name": "Bill",  
  "jobTitle": "Postdoc",  
  "city": "New York",  
  "age": 36 }
```

```
{  
  "name": "John",  
  "age": 27,  
  "employed": true,  
  "hasCar": null,  
  "parents": ["Ann", "John"],  
  "pet": {  
    "name": "Brutus",  
    "species": "dog",  
    "age": 7  
  }  
}
```

non  
array  
object

# The JSON object



john.json

```
{  
  "name": "John",  
  "age": 27,  
  "employed": true,  
  "hasCar": null,  
  "parents": ["Anna", "Michael"],  
  "pet": {  
    "name": "Brutus",  
    "species": "dog",  
    "age": 7  
  }  
}
```

Data is  
separated by  
commas

# The JSON object - indentation



 john.json

```
{
  "name": "John",
  "age": 27,
  "employed": true,
  "hasCar": null,
  "parents": ["Anna", "Michael"],
  "pet": {
    "name": "Brutus",
    "species": "dog",
    "age": 7
  }
}
```

# Structured metadata – JSON vs. XML



john.json

```
{  
  "name": "John",  
  "age": 27,  
  "employed": true,  
  "hasCar": null,  
  "parents": ["Anna", "Michael"],  
  "pet": {  
    "name": "Brutus",  
    "species": "dog",  
    "age": 7  
  }  
}
```



john.xml

```
<name>John</name>  
<age>27</age>  
<employed>TRUE</employed>  
<hasCar>NULL</hasCar>  
<parents>Anna</parents>  
<parents>Michael</parents>  
<pet>  
  <name>Brutus</name>  
  <species>dog</species>  
  <age>7</age>  
</pet>
```

# Structured metadata



## XML

```
<example>
```

```
  <superhero>Wonder Woman</superhero>
```

```
  <publisher>DC Comics</publisher>
```

```
  <identities>
```

```
    <identity>Princess Diana</identity>
```

```
    <identity>Diana Prince</identity>
```

```
  </identities>
```

```
  <pet>
```

```
    <name>Jumpa</name>
```

```
    <species>kangaroo</species>
```

```
  </pet>
```

```
</example>
```

## JSON

```
{
```

```
  "superhero": "Wonder Woman",
```

```
  "publisher": "DC Comics",
```

```
  "identities": [
```

```
    "Princess Diana",
```

```
    "Diana Prince"
```

```
  ],
```

```
  "pet": {
```

```
    "name": "Jumpa",
```

```
    "species": "kangaroo"
```

```
  }
```

```
}
```

## YAML

```
---
```

```
superhero: Wonder Woman
```

```
publisher: DC Comics
```

```
identities:
```

```
- Princess Diana
```

```
- Diana Prince
```

```
pet:
```

```
  name: Jumpa
```

```
  species: kangaroo
```

if you are interested in YAML,  
also see <https://yaml.org/>

Questions?

