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
   }
}
```

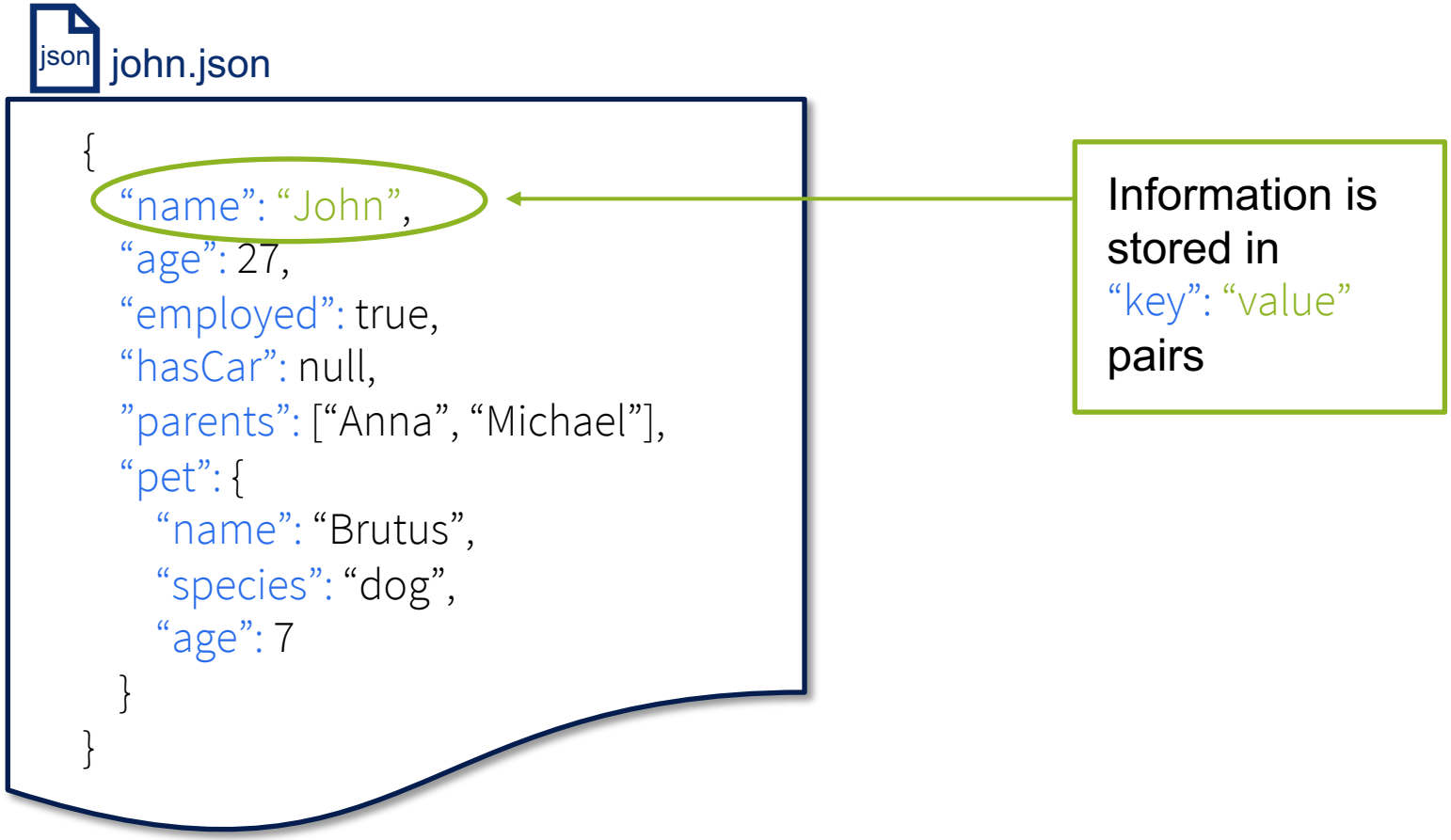
The JSON object: Characteristics

json john.json

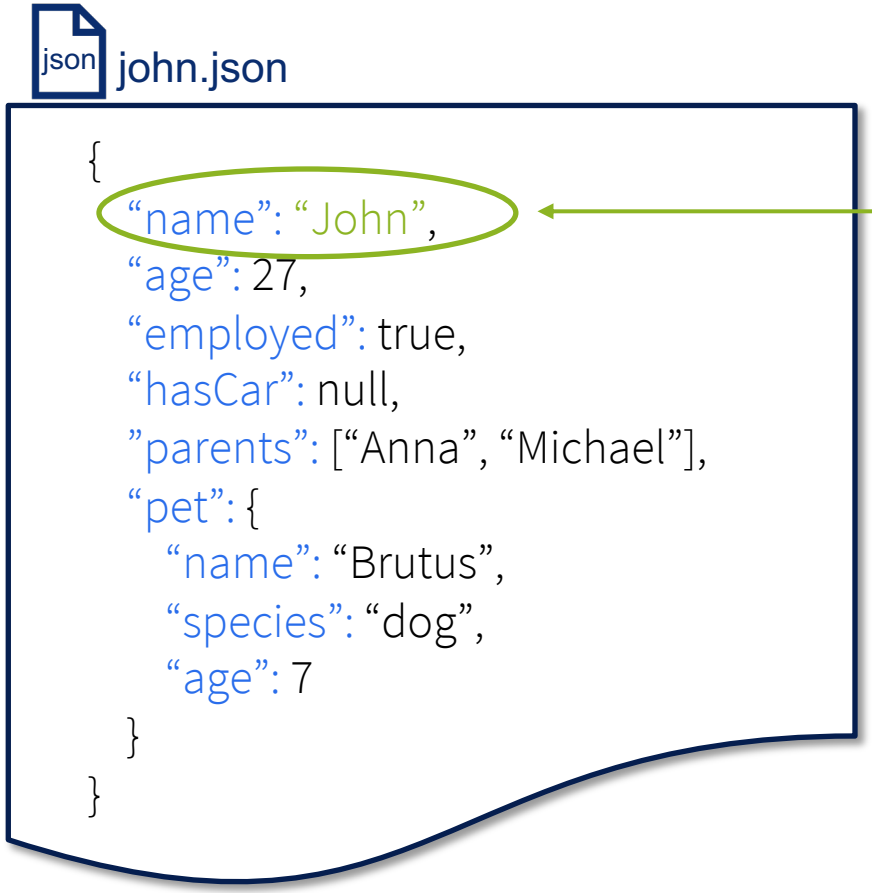
```
{
  “name”: “John”,
  “age”: 27,
  “employed”: true,
  “hasCar”: null,
  ”parents”: [“Anna”, “Michael”],
  “pet”: {
    “name”: “Brutus”,
    “species”: “dog”,
    “age”: 7
  }
}
```

Objects are enclosed in **curly braces**

The JSON object: Characteristics



The JSON object: Characteristics



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

keys are of datatype **string**

The JSON object: Characteristics

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: Characteristics

json 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: Characteristics

json john.json

```
{
  “nam
  “age”
  “emp
  “hasC
  ”pare
  “pet”:
    “name”: “Brutus”,
    “species”: “dog”,
    “age”: 7
  }
}
```

values **must be**

ng

null
array
object

strings are any kind of characters enclosed in “ “

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

The JSON object: Characteristics

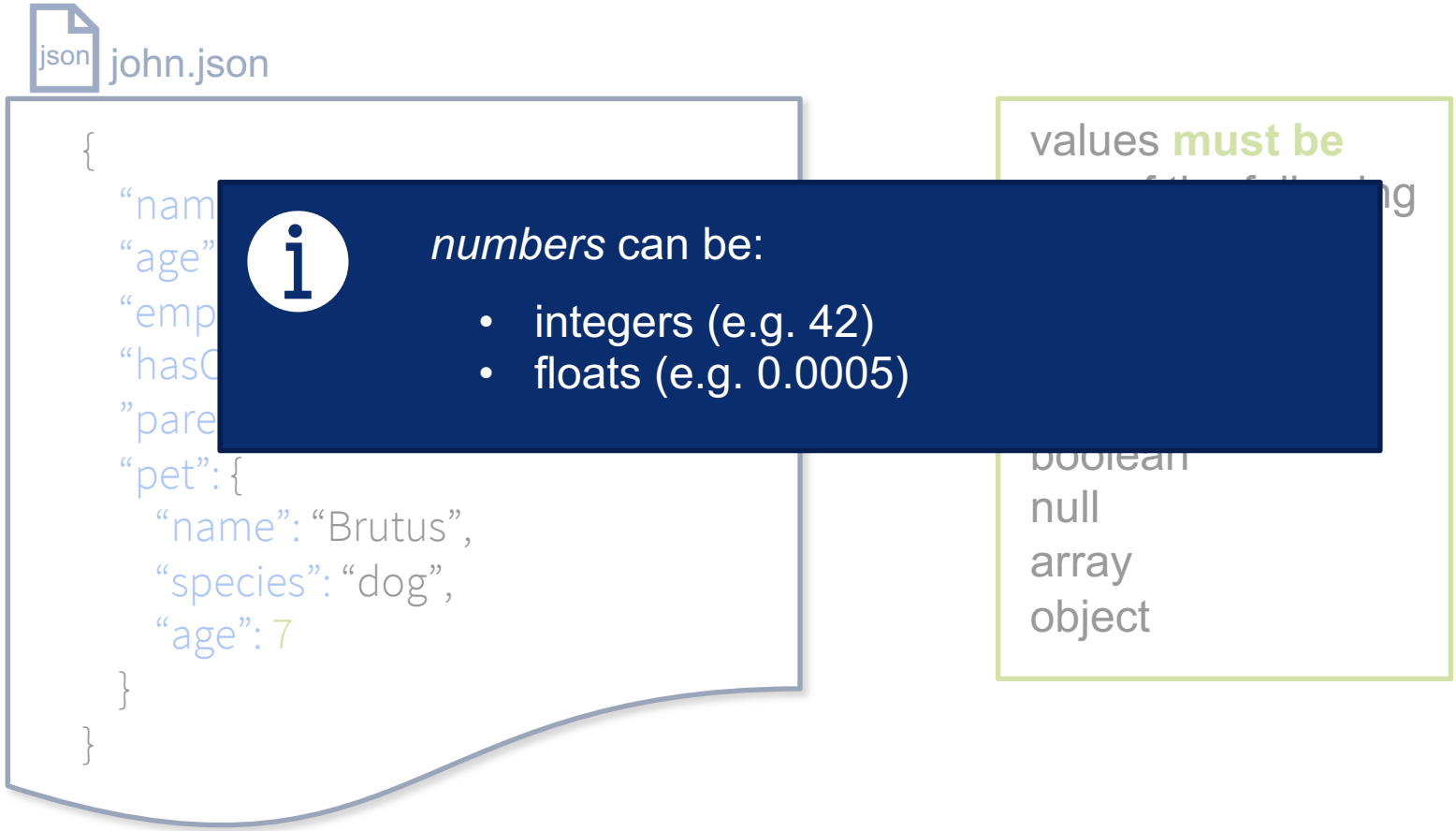
john.json

```
{
  “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: Characteristics



The JSON object: Characteristics

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: Characteristics

json john.json

```
{
  “nam
  “age”
  “emp
  “hasC
  ”pare
  “pet”: {
    “name”: “Brutus”,
    “species”: “dog”,
    “age”: 7
  }
}
```

values **must be**

ng

boolean
null
array
object

a *boolean* has one of two possible values

- true / false
- 1 / 0

The JSON object: Characteristics

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: Characteristics

json john.json

```
{
  “nam
  “age”
  “emp
  “hasC
  ”pare
  “pet”: {
    “name”: “Brutus”,
    “species”: “dog”,
    “age”: 7
  }
}
```

values **must be**

ng

boolean

null

array

object

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

The JSON object: Characteristics

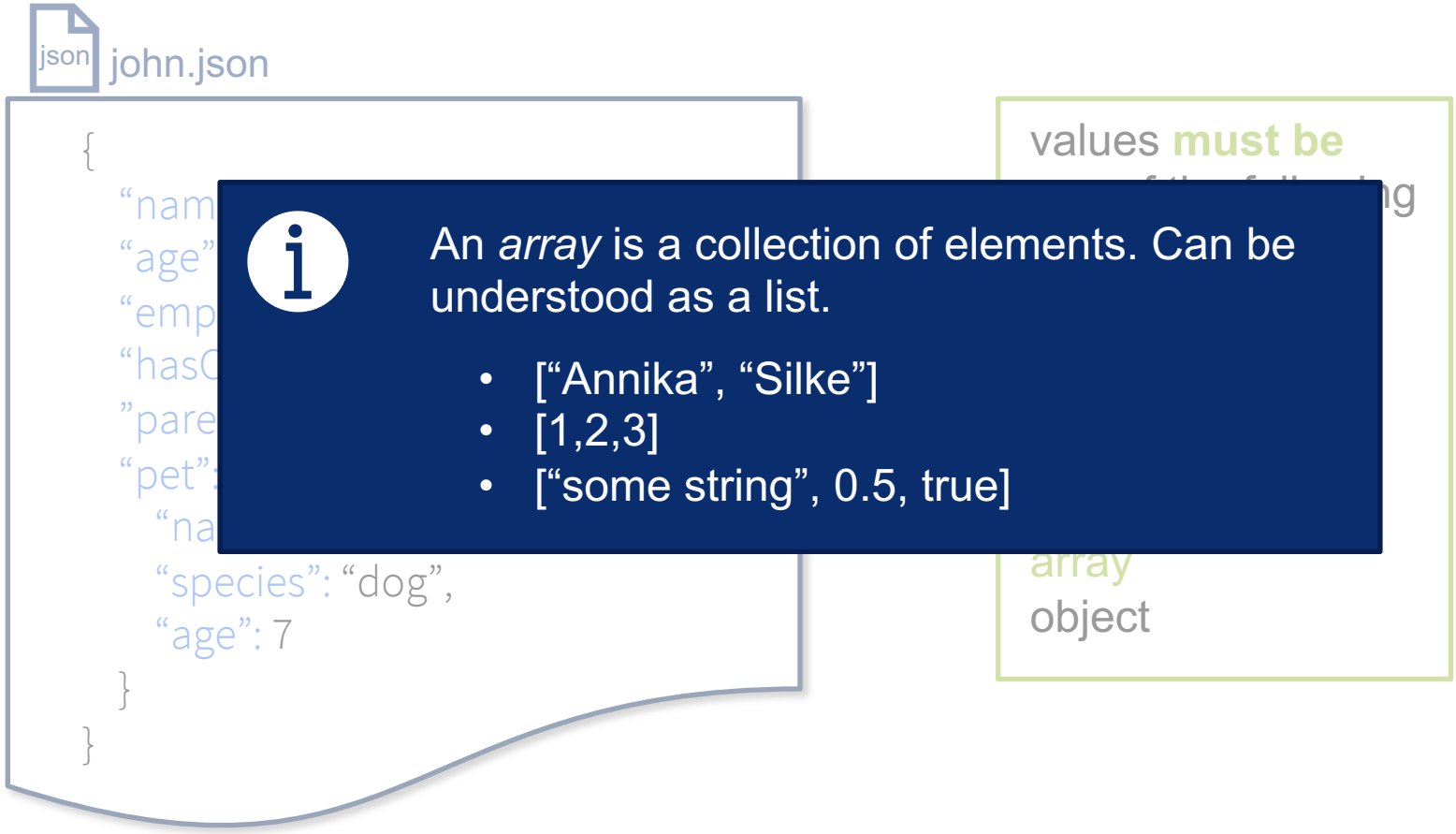
john.json

```
{
  “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: Characteristics



The JSON object: Characteristics

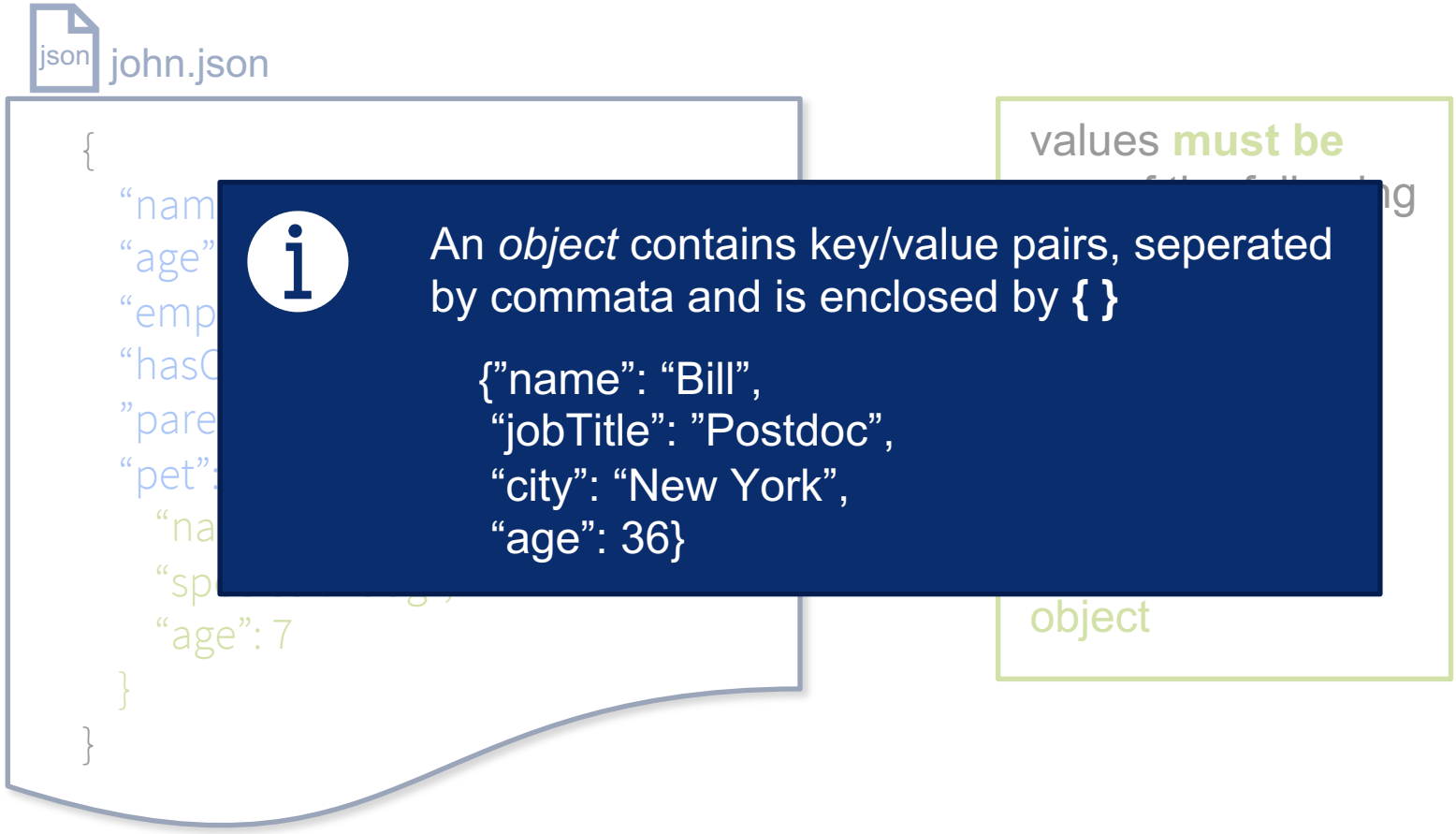
json john.json

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

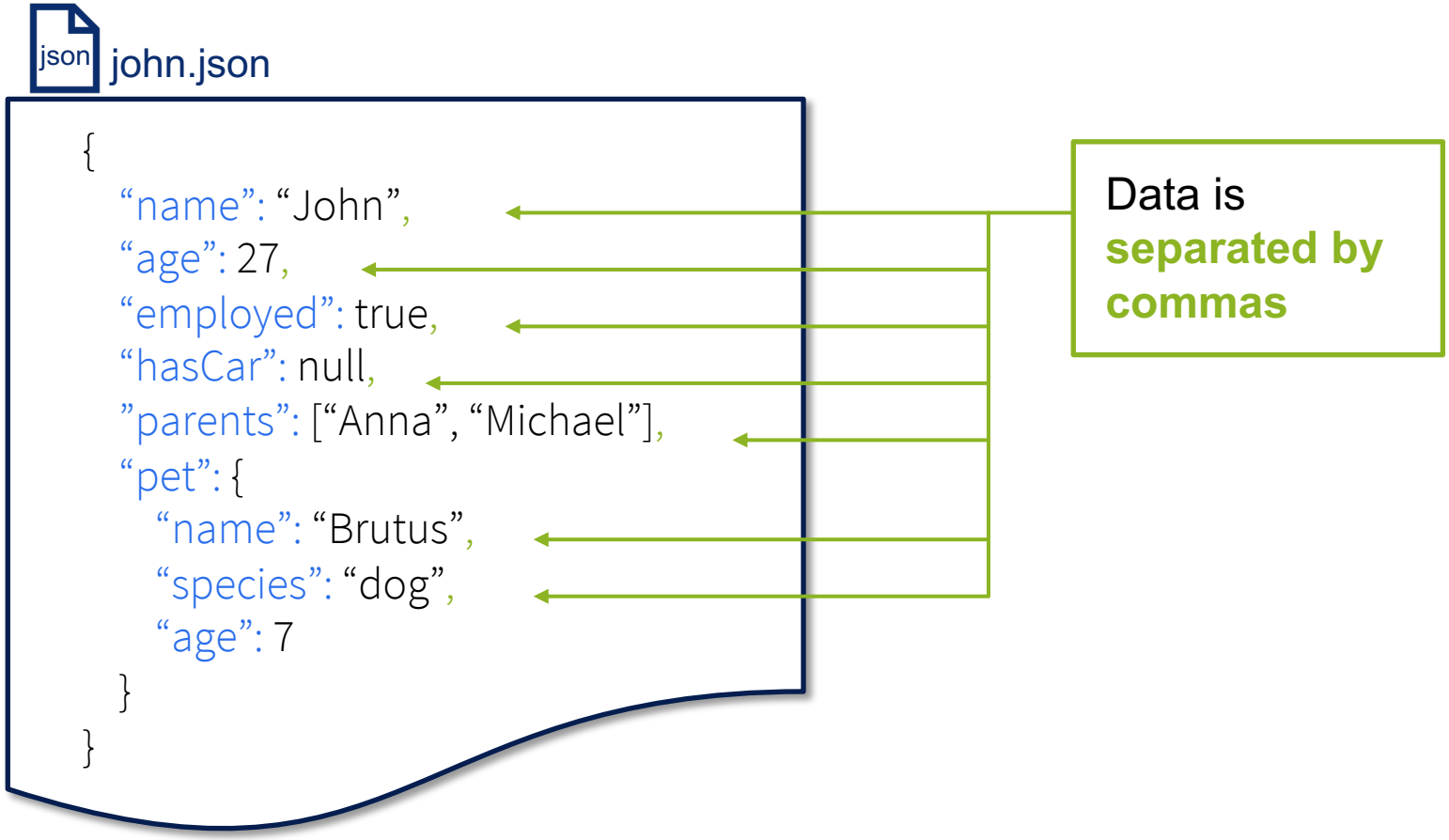
values **must be** one of the following data types:

string
number
boolean
null
array
object

The JSON object: Characteristics



The JSON object: Characteristics



The JSON object - indentation

json 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

json john.json

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

xml 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?