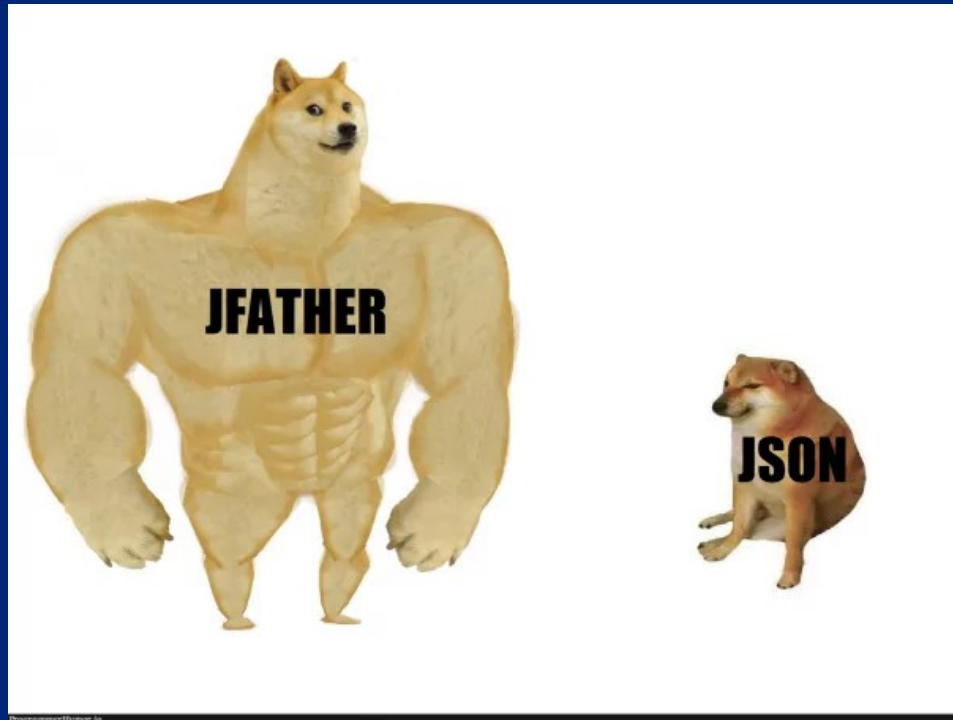


JavaScript (4)



Marina Ribaudò, marina.ribaudò@unige.it

JSON

2

- Acronimo di **JavaScript Object Notation**
- Rappresentazione di un oggetto mediante una stringa
- Formato interoperabile per serializzare oggetti, non è limitato a JavaScript (es. esistono serializzatori JSON per gli oggetti .NET)

JSON

3

- Si tratta di un formato usato per **il trasferimento dei dati attraverso la rete**
- I file JSON hanno estensione **.json**
- MIME type **"application/json"**

JSON: tipi di dato

4

- JSON supporta i seguenti **tipi di dato**
 - string, number, object, array
- E i **letterali**
 - true, false, null
- Le stringhe devono essere delimitate da “...” mentre in JavaScript si può anche usare ‘...’ e `...`

// Invalid JSON in RFC

{ "username": 'ribba' } // solo doppi apici

{ "price": 0xFF } // si possono usare solo cifre decimali

JSON: array e oggetti

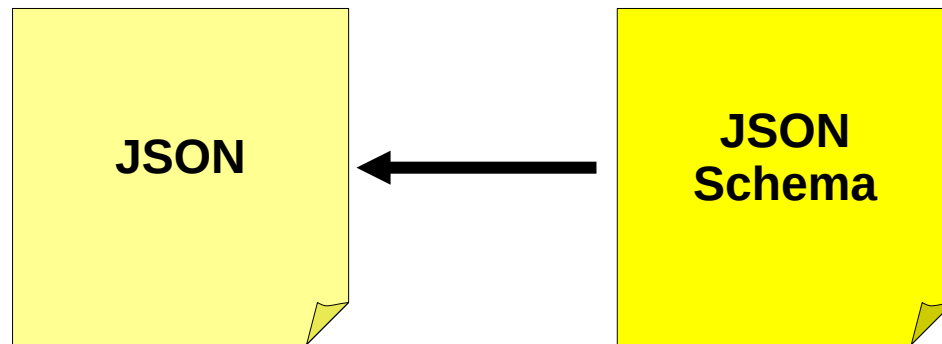
5

- I **dati** sono espressi come
 - Coppie “nome” : “valore”
 - Separate da virgola ,
- Gli **oggetti** sono racchiusi in {...}
- {"nome":"Mario", "cognome":"Rossi"}
- Gli **array** sono racchiusi in [...]
- [
 {"nome":"Mario", "cognome":"Rossi"},
 {"nome":"Anna", "cognome":"Bruni"},
 {"nome":"Pietro", "cognome":"Bianchi"}
]

JSON Schema

6

- JSON Schema è una **specifica**, scritta in JSON, che definisce la struttura, i vincoli e il formato dei dati JSON



- Serve per **convalidare** e **documentare** i dati JSON
- “Simile” a uno schema di database

JSON Schema

7

- Permette di specificare
 - il tipo e il formato dei dati
 - i dati obbligatori
- Esistono servizi online per creare JSON schema e per convalidare file JSON

JSON Schema

8

- <https://www.liquid-technologies.com/online-json-to-schema-converter>
- INPUT

```
[{"firstname":"a","lastname":"a","email":"a@a.it"}]
```


JSON Schema

9

- <https://www.liquid-technologies.com/online-json-to-schema-converter>
- OUTPUT

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "array",
  "items": [
    {
      "type": "object",
      "properties": {
        "firstname": {
          "type": "string"
        },
        "lastname": {
          "type": "string"
        },
        "email": {
          "type": "string"
        }
      },
      "required": [
        "firstname",
        "lastname",
        "email"
      ]
    }
  ]
}
```

JSON Schema

10

- <https://www.liquid-technologies.com/online-json-to-schema-converter>
- OUTPUT

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "array",
  "items": [
    {
      "type": "object",
      "properties": {
        "firstname": {
          "type": "string"
        },
        "lastname": {
```

Esiste la proprietà **pattern** per specificare il formato del dato

```
{
  "type": "string",
  "pattern": "^[A-Z]{6}[0-9]{2}[A-Z][0-9]{2}[A-Z][0-9]{3}[A-Z]$"
}
```

```
    }
  ]
}
```

JSON Validation

11

- <https://jsonlint.com/>
- INPUT: [{firstname:"a","lastname":"a","email":"a@a.it"}]
- OUTPUT

Invalid JSON!

Error: Parse error on line 1:

```
[{firstname:"a","lastn  
--^
```

Expecting 'STRING', '}', got 'undefined'

JSON Validation

12

- <https://jsonlint.com/>
- INPUT: [{"firstname":"a","lastname":"a","email":"a@a.it"}]
- OUTPUT

Invalid JSON!

Error: Parse error on line 1:

...a","email":"a@a.it"}]

-----^

Expecting ',', ']', got 'EOF'

JSON e JavaScript

13

- **JSON.parse(str)**
trasforma una stringa JSON in un oggetto
- **JSON.stringify(value)**
trasforma un valore JavaScript in una stringa JSON
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON