

FORMATO JSON

JSON (JavaScript Object Notation) è un semplice formato per lo scambio di dati:

- per le persone è facile da leggere e scrivere
- per le macchine risulta facile da generare e analizzarne la sintassi.
- Si basa su un sottoinsieme del Linguaggio di Programmazione JavaScript (Standard ECMA-262 Terza Edizione - Dicembre 1999).
- è completamente indipendente dal linguaggio di programmazione, ma utilizza convenzioni conosciute dai programmatori di linguaggi della famiglia del C, come C, C++, C#, Java, JavaScript, Perl, Python, e molti altri.

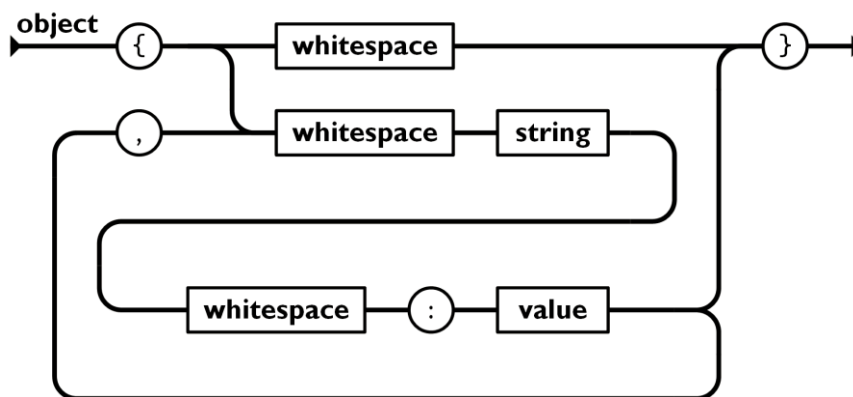
JSON è basato su due strutture:

1. **OGGETTI**: insieme di **coppie nome/valore**. In diversi linguaggi, questo è realizzato come un oggetto, un record, un struct, un dizionario, una tabella hash, un elenco di chiavi o un array associativo.
2. **ARRAY**: **elenco ordinato di valori**. Nella maggior parte dei linguaggi questo si realizza con un array, un vettore, un elenco o una sequenza.

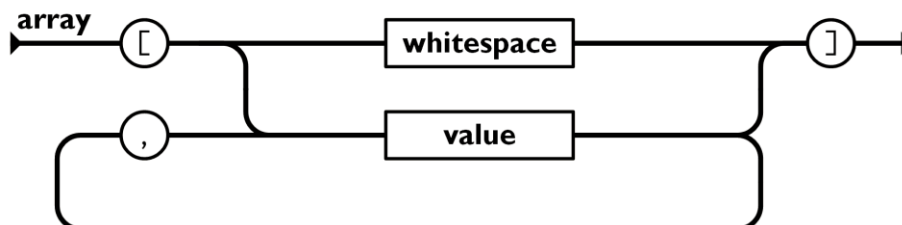
Queste sono strutture di dati universali: tutti i linguaggi di programmazione moderni li supportano in entrambe le forme. E' sensato che un formato di dati che è interscambiabile con linguaggi di programmazione debba essere basato su queste strutture.

In JSON, assumono queste forme:

- Un **oggetto** è una serie non ordinata di nomi/valori. Un oggetto inizia con **{** e finisce con **}**. Ogni nome è seguito da **:** due punti e la coppia di nome/valore sono separata da **,** virgola. (non c'è una virgola dopo l'ultimo elemento).



- Un **array** è una raccolta ordinata di valori. Un array comincia con **[** e finisce con **]**. I valori sono separati da **,** virgola (non c'è una virgola dopo l'ultimo elemento). Possono essere di tipo misto.



- Un **valore** può essere una stringa tra virgolette, o un numero, o vero **true** o falso **false** o nullo **null**, o un oggetto o un array.
- **Queste strutture possono essere annidate.**

Per validare il file <https://jsonlint.com/>

```

public class MyJson {
    String myString = "my string";
    int myInt = 5;
    double[] myArrayOfDoubles = new double[] { 3.14, 2.72 };
    MyOtherJson objectInObject = new MyOtherJson();
}

```

Questa è la relativa rappresentazione JSON di un oggetto della classe:

```

{
  "myString" : "my string",
  "myInt" : 5,
  "myArrayOfDoubles" : [ 3.14, 2.72 ],
  "objectInObject" : {}
}

```

Array di oggetti

```

[
  {
    "precision": "zip",
    "Latitude": 37.7668,
    "Longitude": -122.3959,
    "Address": "",
    "City": "SAN FRANCISCO",
    "State": "CA",
    "Zip": "94107",
    "Country": "US"
  },
  {
    "precision": "zip",
    "Latitude": 37.371991,
    "Longitude": -122.026020,
    "Address": "",
    "City": "SUNNYVALE",
    "State": "CA",
    "Zip": "94085",
    "Country": "US"
  }
]

```

Gli array possono anche contenere elementi con tipi misti, ad esempio:

```

[
  "red",
  51,
  true,
  null,
  {
    "state": "complete"
  }
]

```