



バリューフィットを大切にするユーザベースの「The 7 Values」とは？

詳しくはこちら

⚠ この記事は最終更新日から1年以上が経過しています。



@shizen-shin

2020年06月07日に投稿

# JSONファイルとは？JSONファイルの構造と使い方を理解する



JSON

## JSONファイルとは？JSONファイルの構造と使い方を理解する

JSON（ジェイソン）ファイルについてのまとめ。

### JSONファイルとは

ある決められたフォーマットのテキストファイル。  
情報に名前（キー）をつけ、階層構造で記述できる。

- ・ JavaScript Object Notationの略

JS用に開発されたが、他の言語でも使える。

### 使われる場所

- ・ システムの設定ファイル
- ・ 情報データの保存

ユーザー情報の保存で使われることが多い。

## JSONファイルの記述ルール

- ・ 「"キー名":"値"」でワンセット
- ・ 文字列はダブルクォテーションで囲む（キー名も）
  - シングルクォテーション不可
- ・ 値は①数値、②文字列、③真偽値(小文字)のいずれか
- ・ コメントアウトはない



```
{"キー名":"値"}
```

この形がベース。

データは並列並べたり、入れ子にできる。

- ・ 例えば、名前(name)が轟さんを保存する場合は、

```
{"name":"轟"}
```

- ・ 27歳という年齢データも追加すると、

```
{"name":"轟", "age":"27"}
```

- ・ 新たに、19歳の空条さんを追加する。

```
{{"name":"轟", "age":"27"}, {"name":"空条", "age":"19"}}
```

情報をグループ化し、わかりやすく理解できる。

## 1つのキーに複数のデータを保存する

list型で記述する。

```
{"キー名": [値1, 値2, 値3, ...,]}
```

値は文字列でも数値でもOK。

JSONの記述も入る。

例えば、轟さんが複数の趣味を持っている場合、

```
{"name": "轟", "age": "27", "hobby": ["aaa", "bbb", "ccc"]}
```

## 改行

---

改行で記述されていることが多い。

▼ 

```
{"name": "轟", "age": "27"}
```

以下と同じ。

```
{  
  "name": "轟",  
  "age": "27"  
}
```

▼ 

```
{{"name": "轟", "age": "27"}, {"name": "空条", "age": "19"}}
```

以下と同じ

```
{  
  {  
    "name": "轟",  
    "age": "27"},  
  {  
    "name": "空条",  
    "age": "19"  
  }  
}
```

▼ {"name": "轟", "age": "27", "hobby": ["aaa", "bbb", "ccc"]}

以下と同じ

```
{
  "name": "轟",
  "age": "27",
  "hobby": ["aaa", "bbb", "ccc"]
}
```

こういった情報が保存されていて、その塊がいくつあるかが見やすくなる。

## 値に2次元配列を使う

---

2次元配列も使える。

"hobby":["aaa", "bbb", "ccc"]を2次元配列にした例。

listの中にlistを書く入れ子（ネスト）構造になる。

```
{
  "name": "轟",
  "age": "27",
  "hobby": [["aaa", "bbb", "ccc"], [111, 222, 333]]
}
```

以下と同じ。（値ごとに改行）

例 1

```
{
  "name": "轟",
  "age": "27",
  "hobby": [
    ["aaa", "bbb", "ccc"],
    [111, 222, 333]]
}
```

例 2

```
{
  "name": "轟",
  "age": "27",
  "hobby": [
    [
      "aaa",
      "bbb",
      "ccc"
    ],
    [
      111,
      222,
      333
    ]
  ]
}
```

あまり改行が多いと見にくい、

## 「"キー名":"値"」の入れ子（ネスト）

「"キー名":"値"」の値の部分に「"キー名":"値"」を入れることもできる。

`{"name":"轟", "age":"27"}` に子供のデータを追加する。

`"children":{"name":"Jon", "age":"5", "gender":"male"}`

```
{"name":"轟", "age":"27", "children":{"name":"Jon", "age":"5", "gender":"male"}}
```

改行あり

```
{
  "name": "轟",
  "age": "27",
  "children": {
    "name": "Jon",
    "age": "5",
    "gender": "male"
  }
}
```

```
}  
  
}
```

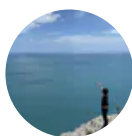
データの深度に制限はないため、"children"の値のなかに更に「"キー名":"値"」を入れ込むこともできる。



ユーザー登録して、Qiitaをもっと便利に使ってみませんか。

登録する

ログインする



@shizen-shin

メモとして活用してます

フォロー



🔗 この記事は以下の記事からリンクされています

 プログラミング初心者が挑んだlaravelの検索機能追加 からリンク 3 months ago

## コメント

この記事にコメントはありません。

あなたもコメントしてみませんか :)

[ユーザー登録](#)

すでにアカウントを持っている方は[ログイン](#)

## 記事投稿イベント開催中



### Azure IoTに関する記事を投稿しよう！

2021/09/10~2021/10/09

[イベント詳細を見る](#)

[👉 すべて見る](#)



How developers code is here.



## Qiita

[About](#) [利用規約](#) [プライバシー](#) [ガイドライン](#) [デザインガイドライン](#)

[リリース](#) [API](#) [ご意見](#) [ヘルプ](#) [広告掲載](#)

## Increments

[About](#) [採用情報](#) [ブログ](#) [Qiita Team](#) [Qiita Jobs](#) [Qiita Zine](#)

© 2011-2021 Increments Inc.