

階層構造と木構造

階層構造

- 親要素が何らかの子要素を持っているデータ構造
- 例
 - 本はいくつかの章で構成される
 - 章はいくつかの節で構成される

階層構造が使われてるデータ表現

- XML: eXtend Markup Language
- JSON: JavaScript Object Notion
- YAML: YAML Ain't a Markup Language / Yet Another Markup Language

JSON

- XMLよりも軽量
- Web系のAPIなので頻繁に利用
- 記述可能なデータ型
 - 文字列：ダブルクォーテーションで**囲む**
 - 数値：ダブルクォーテーションで**囲まない**
 - bool：true, false
 - 配列：後述

- Key-Value構造

- ハッシュテーブル同様に key によりラベル付けされたデータ (value) 構造を記述
- {}を使う

```
{  
  "first": "Taro",  
  "last": "Hirodai"  
}
```

- 配列
 - 原則同じ型の配列
 - []を使う

```
{  
  "first": "Taro",  
  "last": "Hirodai",  
  "score": [87, 92, 58, 50, 80]  
}
```

- 配列の要素にKey-Value

```
[
  {
    "first": "Taro",
    "last": "Hirodai",
    "score": [87, 92, 58, 50, 80]
  },
  {
    "first": "Jiro",
    "last": "Tanaka",
    "score": [54, 36, 64, 76, 44]
  }
]
```

- PythonからJSONを読み書きすることが可能 → ハッシュテーブルとリストに適切に割り当てられる

```
import json
data = json.loads('''
[
  {
    "first": "Taro",
    "last": "Hirodai",
    "score": [87, 92, 58, 50, 80]
  },
  {
    "first": "Jiro",
    "last": "Tanaka",
    "score": [54, 36, 64, 76, 44]
  }
]
''')
data[0]["first"]
```

木構造

- 階層構造の一種
- 親要素と子要素（葉）の属性が同じ
- 根が一つ
- 例
 - 章はいくつかの章で構成される？
 - 家系図（親と子は同じ「人間」属性）
 - 最初の人類（根が一つ）

木構造の実装

- 線形リスト的に子供へのリンク（ポインタ）を持つ
- ハッシュテーブルによる実装

木構造の処理

- 探索などについて再帰処理が基本
- 子を根とした木に対して処理を行う

```
import json
data = json.loads('''
{
  "value": 0,
  "leaf": [
    { "value": 1, "leaf": [
      { "value": 100, "leaf": [] },
      { "value": 3, "leaf": [] }
    ] },
    { "value": 15, "leaf": [] }
  ]
}
''')

def leafvalue(root):
    if len(root["leaf"]) == 0:
        print(root["value"])
        return
    for leaf in root["leaf"]:
        leafvalue(leaf)

leafvalue(data)
```