基礎プログラミングI

第7回ハッシュ・ヘッダ付きCSV

ハッシュとは

身の回りにあるものや人には様々な性質が備わっている。以下のキャラクタに対する属性を管理する場合を考える。

[0]	
	
名前	ドナポン
属性	甘
最大MP	250
初期MP	20

[1]	
	
名前	ラジャイン
属性	土
最大MP	450
初期MP	2

[2]	
	
名前	リッタン
属性	沼
最大MP	150
初期MP	30

[3]	
	
名前	音助
属性	波
最大MP	100
初期MP	80

ハッシュとは

最初のキャラクタの場合、

- **名前→ドナポン**
- **属性→甘**
- **最大MP→250**
- **初期MP→20**

という関係性を持つ。このように、特定のキーワードに対応する値を持つ関係性の集合を**ハッシュ**という。

ハッシュとは

各関係の左辺にある元となるキーワードのことを**キー (key)**、右辺にある対応値のことを**バリュー (value)**という。

キー	バリュー
“名前”	“ドナポン”
“属性”	“甘”
“最大MP”	250
“初期MP”	20

Rubyでのハッシュ表現

Rubyではこの対応関係を

{キー$_1$ => バリュー$_1$, キー$_2$ => バリュー$_2$, …}

の形式で表す。

Rubyでのハッシュ表現

先のキャラクタ例は以下のように表現する。

{“名前” => “ドナポン”, “属性” => 甘, “最大MP” => 250, “初期MP” => 20}

Rubyでのハッシュ表現

これを変数aに代入するには、

```
a = {“名前” => “ ドナポン”, “属性” => 甘, “最大MP” => 250, “初期MP” => 20}
```

とする。

Rubyでのハッシュ表現

変数aから「名前」に対応する値を取り出すには、

a[“名前”]

とする。

Rubyでのハッシュ表現

逆に値を設定するときは

a[“最大MP”] = 350

とする。

ハッシュオブジェクト

右のような新キャラクタのハッシュ値を利用者が自由に設定できるようにする場合を考える。

[3]	
	
名前	???
属性	???
最大MP	???
初期MP	???

ハッシュオブジェクト

この場合、キーとバリューのペアを一つも持っていない**空のハッシュ値**を入れてから代入させる。

```
newchar = {}              # {} が空のハッシュ値
print "名前は何にしますか?: "
newchar["名前"] = gets.chomp
print "属性は何にしますか?: "
newchar["属性"] = gets.chomp
print "最大MPはいくつにしますか?: "
newchar["最大MP"] = gets.to_i
print "初期MPはいくつにしますか?: "
newchar["初期MP"] = gets.to_i
```

ハッシュオブジェクト

空のハッシュ値はHash.newでも生成できる。

```
newchar = Hash.new
```

ハッシュ形式の値のことを**ハッシュオブジェクト**という。

ハッシュ値の集合

ハッシュ値を要素とする配列を定義してハッシュ値の集合を表すことができる。

[ハッシュ値$_1$, ハッシュ値$_2$, …]

ハッシュ値の集合

先の4つのキャラクタのハッシュ値の集合は

```
[
{"名前" => "ドナポン", "属性" => "甘", "最大MP" => 250, "初期MP" => 20},
{"名前" => "ラジャイン", "属性" => "土", "最大MP" => 450, "初期MP" => 2},
{"名前" => "リッタン", "属性" => "沼", "最大MP" => 150, "初期MP" => 30},
{"名前" => "音助", "属性" => "波", "最大MP" => 100, "初期MP" => 80}
]
```

で表せる。

ハッシュ値の集合

これを変数charaに代入するには、

```
chara = [
 {"名前" => " ドナポン", "属性" => "甘", "最大MP" => 250, "初期MP" => 20},
 {"名前" => "ラジャイン", "属性" => "土", "最大MP" => 450, "初期MP" => 2},
 {"名前" => "リッタン", "属性" => "沼", "最大MP" => 150, "初期MP" => 30},
 {"名前" => "音助", "属性" => "波", "最大MP" => 100, "初期MP" => 80}
]
```

とする。

ハッシュ値の集合

先頭要素[0] ({"名前" => " ドナポン", "属性" => "甘", "最大MP" => 250, "初期MP" => 20})を取り出すなら

chara[0]

とする。

ハッシュ値の集合

4番目[3]の名前を取り出すなら

chara[3]["名前"]

とする。

ハッシュ値の集合

4番目[3]の初期MPを上書きして
90に更新するなら

```
chara[3]["初期MP"] = 90
```

とする。

ヘッダ付きCSV

ヘッダ付きCSV

見出しとなるヘッダ行のついたCSVファイルをrubyで扱うことができる。

```
氏名,数学,英語
山田太郎,50,70
中町太郎,90,80
飯森花子,91,60
鶴岡一人,60,45
酒田三吉,52,82
三川一二三,12,98
```

ヘッダ付きCSV

ヘッダ行を持つCSVファイルはCSV.readを用い

```
CSV.read(CSVファイル名, headers: true)
```

とするとヘッダ構造を持った値になる。

ヘッダ付きCSV

score2h.csvファイルを読み込んで変数seisekiに代入するには以下のようにする。

```
seiseki = CSV.read(“score2h.csv”, headers: true)
```

ヘッダ付きCSV

先頭要素[0]を取り出す場合

```
seiseki[0]
```

先頭要素の“数学”の値を取り出す場合

```
seiseki[0][“数学”]
```

列の取り出し

seiseki[“数学”]とすると、数学の値のみを集めた配列

["50", "90", "91", "60", "52", "12"]

になる。

列の取り出し

数学の合計点を求めるには以下のようにすればよい。

```
sum = 0
seiseki["数学"].each do |pt|
  sum += pt.to_f
end
```

ヘッダの取り出し

ヘッダを取り出す場合

```
seiseki.header
```

まとめ

- 特定のキーワードに対応する値を持つ関係性の集合を**ハッシュ**という。
- 各関係の左辺にある元となるキーワードのことを**キー (key)**、右辺にある対応値のことを**バリュー (value)**という。
- ヘッダ行を持つCSVファイルを読み込むときはCSV.read(ファイル名, **headers: true**)とする。