

### クラス

・オブジェクトの「設計図」「雛形」「テンプレート」



### オブジェクト

・クラスからつくられる「実体」



※「データ」と「データに対する操作(メソッド)」を  
ひとまとめにした「部品」「パーツ」

### 配列

・「データ」や「オブジェクト」の保存用の器



※中に保存されている「データ」や  
「オブジェクト」を  
1つずつ簡単に扱うことができる

## 3-3 クラス・オブジェクト(インスタンス)・配列

---

### 考えてみよう...

- (1) 爆弾ゲーム(9-3)を拡張して「Ruby 君が弾を撃ち返す」ようにしたときに新たに追加した下記の A) B) C) それぞれの要素は、前ページの写真のそれぞれどれに該当するものでしょうか？

A)

```
class Tama  
  ~  
end
```

B)

```
tama = Tama.new(hero.centerx, hero.centery)
```

C)

```
tamas = []  
...  
tamas << tama  
...  
tamas.each do |tama|  
  ~  
end
```

### ついでの説明...

オブジェクト指向とは、プログラムの中で使われる「データ」と「機能(操作)」を「オブジェクト」という単位でまとめて考えるプログラムの考え方です。

プログラムの中で使われる「データ」と「そのデータを処理する機能(操作)」をひとまとめにして独立させ、「データを管理する側」と「そのデータを利用する側」といった立場を明確にわけること、プログラムの流れを考えやすくしたり、プログラムを部品として再利用しやすくしたりすることができます。

ある程度以上の複雑さを持ったプログラムを書こうとすると、プログラムの全体をつねに把握するのが困難になってくるため、このような考え方がとても重要になってきます。

- ・プログラムの変更箇所を局所化できる
- ・現実の世界をそのままに近い形でモデリングしやすい

