



po-6. クラス, コンストラクタ, メソッド

(Python プログラミング体験学習, 全9回)

<https://www.kkaneko.jp/cc/po/index.html>

金子邦彦





アウトライン

- 6-1 クラス定義, コンストラクタ
- 6-2 メソッド
- 6-3 メソッドに関する実習

- **メソッド**を Python で扱えるようになる.
- **メソッド**内での**属性**や**メソッド**への**アクセス**について理解する.



6-1. 復習



オブジェクト

- プログラミングでのオブジェクトは、コンピュータでの操作や処理の対象となるもののこと
- オブジェクトは1つあるいは複数のデータを持つことができる



メソッド

- プログラミングのメソッドは、オブジェクトに属する操作や処理のこと

hero.moveDown()

hero は**オブジェクト**

moveDown() は**メソッド**

間を「.」で区切っている

- 引数（ひきすう）とは、メソッドに渡す値のこと



6-1. クラス定義, コンストラクタ



問いかけ

- Python で、**オブジェクトの生成**はどのように行うのでしょうか？

Python のオブジェクトの生成



- 次の2つの**オブジェクト**を生成する Python プログラム

| | | | |
|---|---|---|---------|
| a | 1 | 2 | "red" |
| b | 2 | 4 | "green" |
| | x | y | color |

- クラス Ball のオブジェクト生成を行うプログラム

```
a = Ball(1, 2, "red")  
b = Ball(2, 4, "green")
```


クラス定義



クラス定義の中には、**コンストラクタ**の定義、その他**メソッド**の定義を含める。

```
1 class Ball:
2     def __init__(self, x, y, color):
3         self.x = x
4         self.y = y
5         self.color = color
6     def move(self, xx, yy):
7         self.x = self.x + xx
8         self.y = self.y + yy
9     def right(self, a):
10        self.move(a, 0)
11    def left(self, a):
12        self.move(-a, 0)
```

コンストラクタ

その他
メソッド

オブジェクトの生成を行う**メソッド**のことを
コンストラクタという



実習の指示

- 資料： **11**～**16**
- 次のことを理解しマスターする
 - クラス定義
 - コンストラクタの定義
 - オブジェクト生成

実習



① **ウェブブラウザ**を起動する

② **Python Tutor** を使いたいのので、次の URL を開く

<http://www.pythontutor.com/>

※ Internet Explorer でうまく動かない場合がある

→ うまく動かないときは Google Chrome を試してください

※ 途中で 「Server Busy . . .」 というメッセージが出る
ことがある。

→ 混雑している。 少し（数秒から数十秒）待つと自動で表示
が変わる（変わらない場合には、操作をもう一度行ってみ
る）

※ 日本語モードはない。英語で使う



③ 「Python Tutor」をクリック

← → ↻ ⓘ 保護されていない通信 | pythontutor.com ☆ APP 2

VISUALIZE CODE AND GET LIVE HELP

Learn Python, Java, C, C++, JavaScript, and Ruby

[Python Tutor](#) (created by [Philip Guo](#)) helps people overcome a fundamental barrier to learning programming: understanding what happens as the computer runs each line of code.

Write code in your web browser, see it visualized step by step, and get live help from volunteers.

Related services: [Java Tutor](#), [C Tutor](#), [C++ Tutor](#), [JavaScript Tutor](#), [Ruby Tutor](#)

Over five million people in more than 180 countries have used Python Tutor to visualize over 100 million pieces of code, often as a supplement to textbooks, lectures, and online tutorials.

[Visualize your code and get live help now](#)



Get live help!

Start private chat

These Python Tutor users are asking for help right now. Please volunteer to help!
user_041 from Singapore, Singapore needs help with Python3 - 2 people chatting - requested 12 minutes ago)
user_91c from Vélizy-Villacoublay, France needs help with Python3 - [click to help](#) minutes ago)
user_985 from Seattle, Washington, US needs help with Python3 - [click to help](#) (-

「Python 3.6」になっている

Write code in Python 3.6

```
1 |
```

エディタ

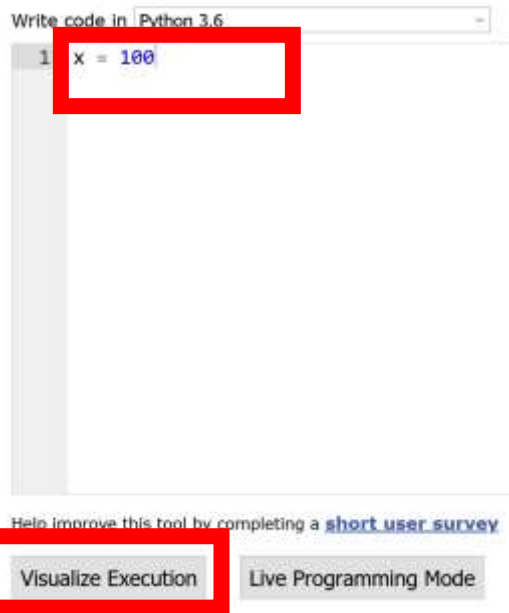
Visualize Execution

実行のためのボタン

Live Programming Mode

hide exited frames [default] inline primitives but don't nest objects [default]
draw pointers as arrows [default]

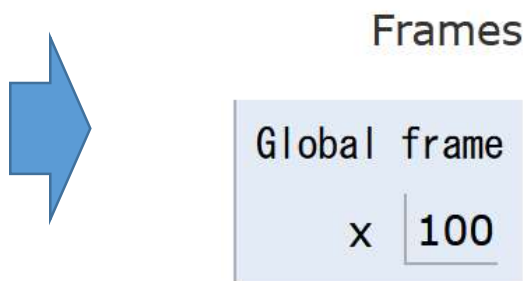
Python Tutor でのプログラム実行手順



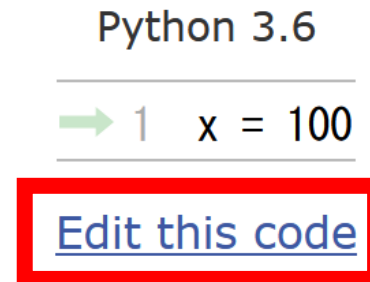
(1) 「**Visualize Execution**」をクリック。



(2) 「**Last**」をクリック。



(3) 結果を確認する。



(4) 「**Edit this code**」をクリックして戻る

④ 次のプログラムを実行し，結果を確認しなさい



クラス定義

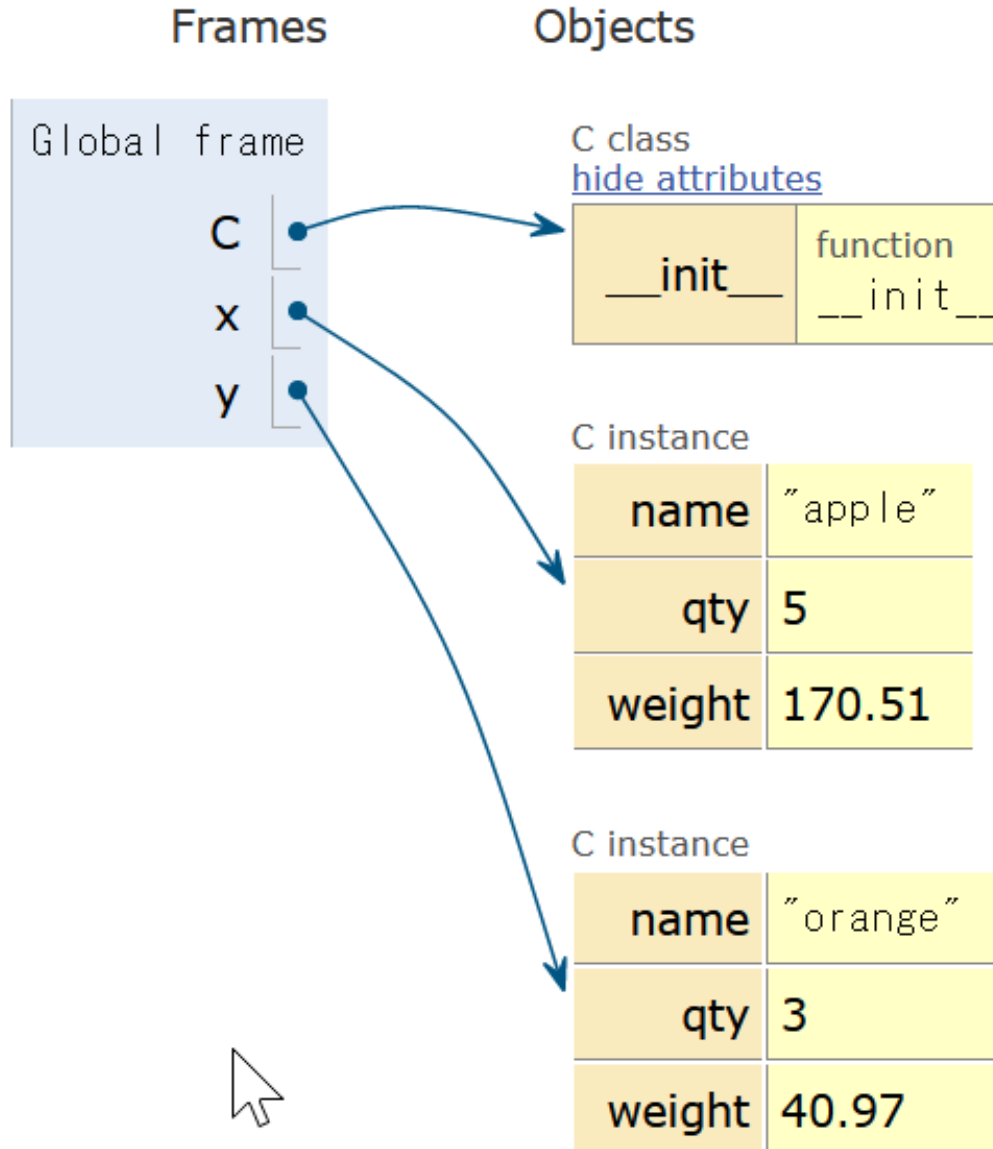
```
1 class C:
2     def __init__(self, qty, weight, name):
3         self.qty = qty
4         self.weight = weight
5         self.name = name
6
7 x = C(5, 107.51, "apple")
8 y = C(3, 40.97, "orange")
9 print(x.qty)
```

コンストラクタによる
オブジェクト生成

結果は次ページに記載

Print output (drag lower right corner to resize)

5



2つのオブジェクト
x, y が生成される



6-2. メソッド



問いかけ

- 「メソッド」とは何でしょうか？



メソッドと クラス

- プログラミングでの**メソッド**とは、オブジェクトに関する操作や処理のこと
- **メソッド**は、**クラス**に属する
- **メソッド**内のプログラムは、その**メソッド**が所属する**クラス**の**属性**や**メソッド**へのアクセス権がある

メソッドの定義



クラス名 **C**

属性 **qty, weight, name**

メソッド **total**

コンストラクタの定義

```
1 class C:
2     def __init__(self, qty, weight, name):
3         self.qty = qty
4         self.weight = weight
5         self.name = name
6     def total(self):
7         return self.qty * self.weight
```

メソッドの定義

クラス定義



属性やメソッド のアクセス

- 「オブジェクト名」 + 「.」で属性やメソッドにアクセスする

属性アクセス



| | | | |
|---|---|--------|---------|
| x | 5 | 170.51 | "apple" |
|---|---|--------|---------|

qty weight name

「オブジェクト名」 + 「.」で属性やメソッドにアクセスする

```
1 class C:
2     def __init__(self, qty, weight, name):
3         self.qty = qty
4         self.weight = weight
5         self.name = name
6     def total(self):
7         return self.qty * self.weight
8
9 x = C(5, 107.51, "apple")
10 y = C(3, 40.97, "orange")
11 print(x.qty)
12 print(x.total())
```

メソッドアクセス



「オブジェクト名」 + 「.」で属性やメソッドにアクセスする

```
1 class C:
2     def __init__(self, qty, weight, name):
3         self.qty = qty
4         self.weight = weight
5         self.name = name
6     def total(self):
7         return self.qty * self.weight
8
9 x = C(5, 107.51, "apple")
10 y = C(3, 40.97, "orange")
11 print(x.qty)
12 print(x.total())
```



実習の指示

- 資料：25, 26
- 次のことを理解しマスターする
 - メソッド定義
 - 「オブジェクト名」 + 「.」で属性やメソッドにアクセス

次のようにプログラムを書き換えて，実行し，結果を確認
しなさい



```
1 class C:
2     def __init__(self, qty, weight, name):
3         self.qty = qty
4         self.weight = weight
5         self.name = name
6     def total(self):
7         return self.qty * self.weight
8
9 x = C(5, 107.51, "apple")
10 y = C(3, 40.97, "orange")
11 print(x.qty)
12 print(x.total())
```

メソッド定義

「.」でメソッドにアクセス

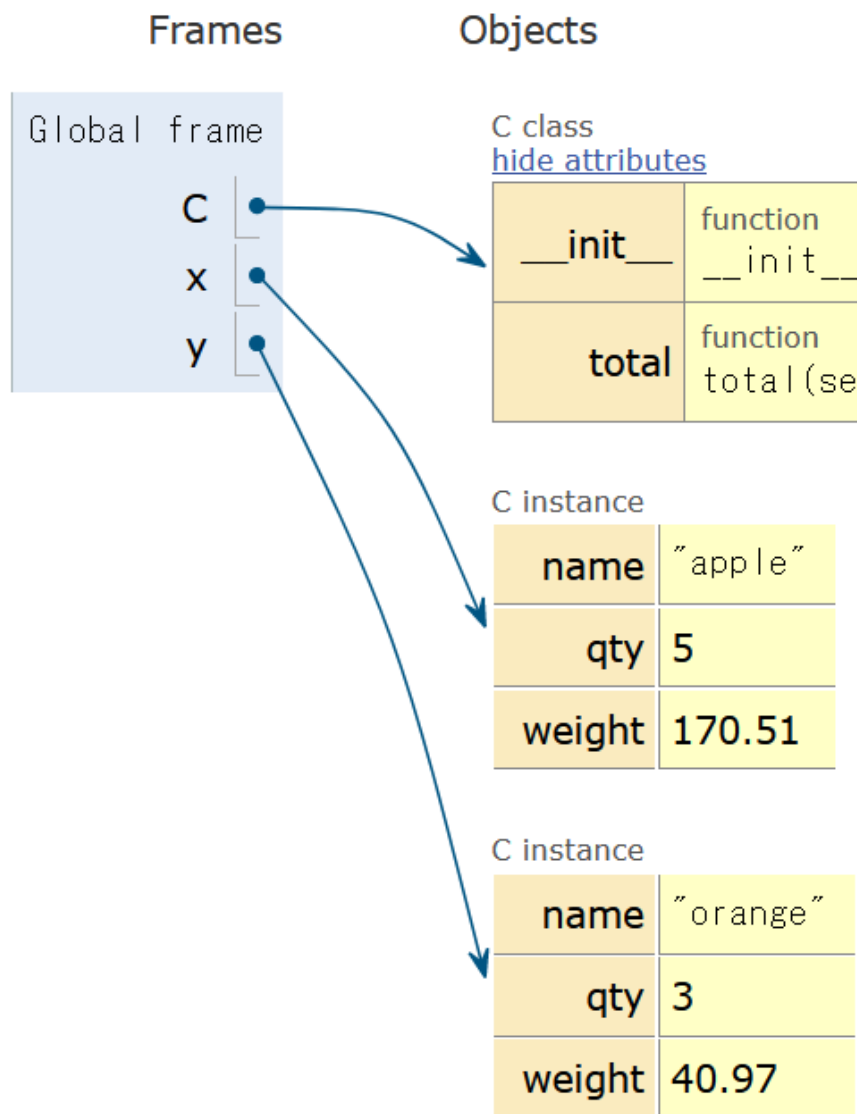
結果は次ページに記載

Print output (drag lower right corner to resize)

```
5  
852.55
```



表示を確認





問いかけ

- 「**メソッド**」とは何でしょうか？
プログラミングでの**メソッド**とは、オブジェクトに関する操作や処理のこと
- 「**オブジェクト名**」 + 「**.**」で**属性**や**メソッド**にアクセスする



キーワード

- **class** **クラス定義**
- **return** **メソッドやメソッド**
の**評価値**を呼び出し側に返す

まとめ



- Python の**クラス定義**では、**クラス名**の指定、**コンストラクタ**の定義、**メソッド**の定義を行う
- 「**オブジェクト名**」 + 「**.**」で**属性**や**メソッド**にアクセスする

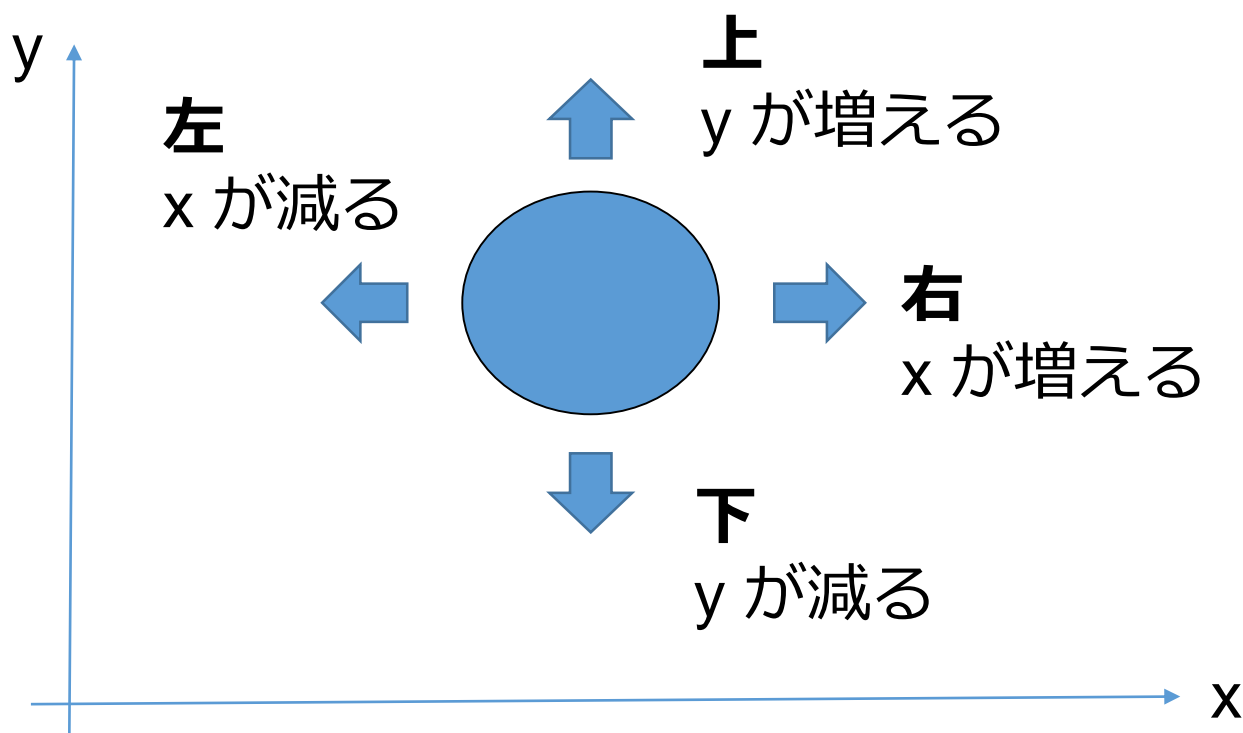


6-3. メソッドに関する実習

Python のメソッド



- 次の**メソッド**を考える.
 - Ball クラスに属する**メソッド** move
 - この**メソッド**は, **オブジェクト**の属性 x, y を増減させるためのもの



実習の指示

- 資料：33～36
- 次のことを理解しマスターする
 - メソッド定義
 - メソッドへのアクセス



① 次のプログラムを実行し，結果を確認しなさい

```
1 class Ball:
2     def __init__(self, x, y, color):
3         self.x = x
4         self.y = y
5         self.color = color
6     def move(self, xx, yy):
7         self.x = self.x + xx
8         self.y = self.y + yy
9     def right(self, a):
10        self.move(a, 0)
11    def left(self, a):
12        self.move(-a, 0)
13
14 a = Ball(1, 2, "red")
15 b = Ball(2, 4, "green")
16 a.move(10, 20)
```

結果は次ページに記載

Frames

Objects



Global frame

| | |
|-------------|---|
| Ball | • |
| a | • |
| b | • |

Ball class
hide attributes

| | |
|-----------------|---|
| __init__ | function __init__(self, x, y, color) |
| move | function move(self, xx, yy) |

Ball instance

| | |
|--------------|-----------|
| color | "red" |
| x | 11 |
| y | 22 |

Ball instance

| | |
|--------------|----------|
| color | "green" |
| x | 2 |
| y | 4 |

② 次のようにプログラムを書き換えて、実行し、結果を確認しなさい



```
1 class Ball:
2     def __init__(self, x, y, color):
3         self.x = x
4         self.y = y
5         self.color = color
6     def move(self, xx, yy):
7         self.x = self.x + xx
8         self.y = self.y + yy
9     def right(self, a):
10        self.move(a, 0)
11    def left(self, a):
12        self.move(-a, 0)
13
14    a = Ball(1, 2, "red")
15    b = Ball(2, 4, "green")
16    a.move(10, 20)
```

メソッド定義

結果は次ページに記載

「.」でメソッドにアクセス



Frames

Objects

Global frame

| | |
|-------------|---|
| Ball | • |
| a | • |
| b | • |

Ball class

hide attributes

| | |
|-----------------|---|
| __init__ | function __init__(self, x, y, color) |
| left | function left(self, a) |
| move | function move(self, xx, yy) |
| right | function right(self, a) |

Ball instance

| | |
|--------------|-------|
| color | "red" |
| x | -9 |
| y | 2 |

Ball instance

| | |
|--------------|---------|
| color | "green" |
| x | 2 |
| y | 4 |



全体まとめ



- プログラミングでの**メソッド**とは、**オブジェクトに関する操作や処理**のこと
- **メソッド**は、**クラス**に属する
- 「**オブジェクト名**」 + 「**.**」で**属性**や**メソッド**にアクセスする