第7回　Python における実装（クラス）　2021年6月1日

オブジェクト指向型言語

東京都立大学　システムデザイン学部
情報科学科　西内信之

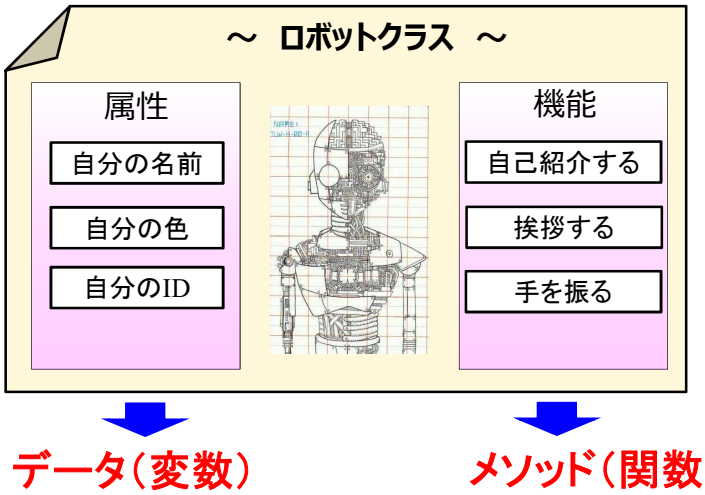
【授業計画・内容授業方法】(変更後：予定)

1. イントロダクション（ガイダンス）
2. オブジェクト指向の歴史・特徴・概念
3. クラス、継承、ポリモーフィズム
4. Python における実装（基礎、変数、条件、繰り返し）
5. Python における実装（データの型）
6. Python における実装（関数）
7. **Python における実装（クラス）**
8. Python における実装（継承、クラス変数とインスタンス変数）
9. 中間試験（レポート）
10. Python における実装（標準ライブラリ、外部ライブラリ）
11. Python における実装（tkinter）
12. Pythonで作るゲーム
13. オブジェクト指向開発と設計1
14. オブジェクト指向開発と設計2
15. 期末試験（レポート）・まとめ

1. Pythonにおける実装（クラス）

1.1 クラスとは

データと処理するプログラムをひとまとまりにして扱う機能のこと



関数：いくつかの処理を
まとめたもの

```
def xxxx():

    計算をする
    文字列を表示する
    ファイルに書き込む
        .
        .
        .
```

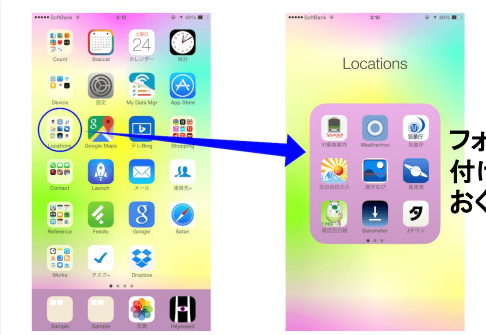
クラス：いくつかの関数と
変数をまとめたもの

```
class Aaaa:
    変数A
    変数B ・・・
    def xxxx():

    def yyyy():
```

1.2　クラスを使うと何がうれしいのか

(1) プログラムを整理できる



フォルダごとに名前を
付けてひとまとめにして
おくと後から探しやすい

全体のプログラムがスマートフォン
クラスがアプリのフォルダ
関数がアプリ

(2) 量産ができる

クラスは設計図

変数
耐久力:1000
スピード:600

関数
・ビームを撃つ。
・ミサイルを
　発射する。

ステージ1に100機

ステージ2に200機

量産するときに
変数を変えることもできる

1.3 クラスを作る

クラス

クラスの書式

```
class クラス名:
    変数の定義
    関数の定義
```

データ

メソッド

はじめにclassというキーワード、その後に作りたいクラスの名前。
その下に、変数の定義、関数の定義。（変数だけ、関数だけでもよい）

クラスの作成

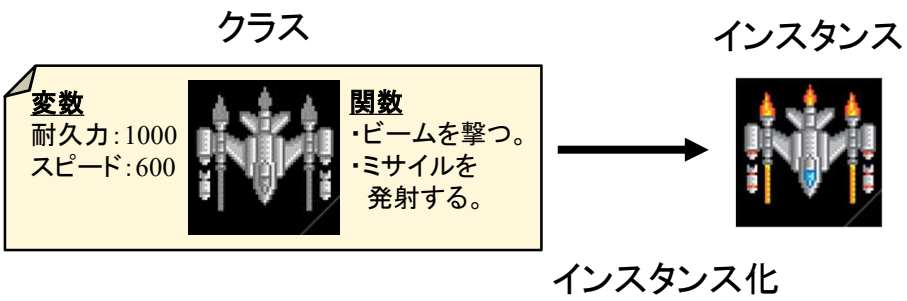
```
class Fruit:
    color="red"
    def taste(self):
        print("すっぱい")
        print("おいしい")
```

データ

メソッド

クラス名は慣習的に大文字で始めます。

1.4　インスタンス化



インスタンス化

```
変数 ＝ クラス名()
```

ここでの変数がインスタンス。

インスタンスの作成

```
class Fruit:
    color="red"
    def taste(self):
        print("すっぱい")
        print("おいしい")
```

```
apple=Fruit()
```
← インスタンス化

Apple がインスタンス。

クラスの変数の表示、メソッド（関数）の呼び出し

```
class Fruit:
    color="red"
    def taste(self):
        print("すっぱい")
        print("おいしい")
```

```
apple=Fruit()
```

変数の表示

```
apple.color
```

インスタンス名．変数

```
'red'
```

メソッドの呼び出し

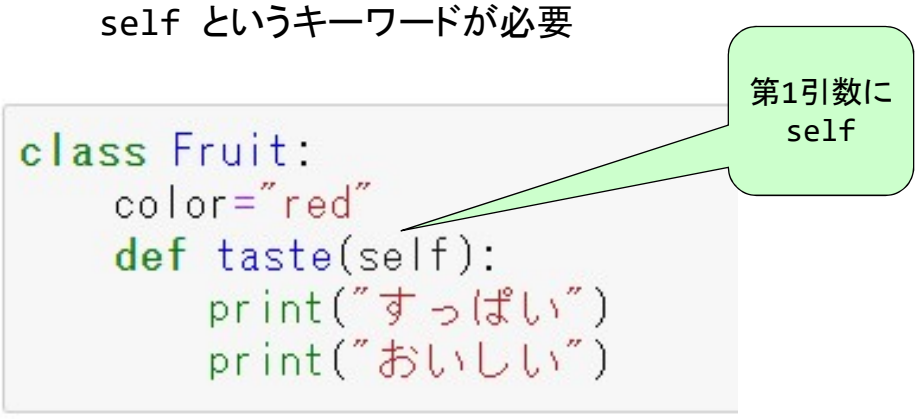
```
apple.taste()
```

インスタンス名．メソッド名（）

```
すっぱい
おいしい
```

1.5　メソッドの引数 self

クラスの中のメソッド（関数）の第1引数に
self というキーワードが必要



self を書かなかったら

```
class Fruit:
    color="red"
    def taste():
        print("すっぱい")
        print("おいしい")
```

```
apple=Fruit()
```

```
apple.color
```

```
'red'
```

```
apple.taste()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-15-200d19f2223c> in <module>
----> 1 apple.taste()

TypeError: taste() takes 0 positional arguments but 1 was given
```

エラー:「taste関数は引数をとらない(定義していない)のに、
引数が1つ渡されてきましたよ」

「関数を呼び出すときに、必ず1つ引数が自動的に渡される」というルールがある。

self　クラス自身を示す

selfからはクラス内で定義した
変数や関数を呼び出せる

selfは呼び出し側では指定不要。
呼び出し側は第2引数以降を指定。
第2引数以降は省略可能。

self とは？

```
class SampleClass:
    # メソッド1「引数messageの値を表示する」
    def s_method1(self, message):
        print(message)

    # メソッド2「メソッド1を呼び出す」
    def s_method2(self):
        self.s_method1("Hello World!")
```

```
ins = SampleClass()                    # インスタンス化
ins.s_method2()                        # メソッドの呼び出し


# 結果 ⇒ Hello World!
```

メソッド s_method1 では、
画面に引数 message の値を出力する処理を定義している。

メソッド s_method2 では、
先ほど定義した s_method1 の呼び出し。
引数 message には文字列 “Hello World!” を渡してる。

ポイント！！
メソッドの呼び出し部分
「self.s_method1(“Hello World!”)」となってます!!

先に解説した通り、メソッドの呼び出し時には
インスタンスを指定する必要がある。
インスタンス名 .メソッド名()

クラスの中ではまだインスタンスが生成されていません。
そのために必要なのが仮のインスタンス名の役割を担う self 。
この self はこのクラス自身を表していると説明できます。

1.6 クラスでコンストラクタを使う

2つのアンダースコアから始まり、
2つのアンダースコアで終わる
「特殊メソッド」と呼ばれる

```
class クラス名:
     def __init__( self, 引数 ) :
          self.初期設定したい変数＝引数
          最初に行いたい処理
```

インスタンス生成時に同時にインスタンス変数を設定できる。

インスタンス生成時に最初に1度だけ実行される、
「**初期化用**の特殊な関数」。

インスタンス生成時、第1引数を指定する必要はなく、
第2引数から記述。

```
class Example():
    def __init__(self, a,b):
        self.num1 = a
        self.num2 = b

    def print_tot(self):
        tot = self.num1 + self.num2
        print(tot)

myinstance = Example(1,2)
myinstance.print_tot()
```

```
class Example():
    def __init__(self, a,b):
        self.num1 = a
        self.num2 = b

    def print_tot(self):
        tot = self.num1+self.num2
        print(tot)
```

```
myinstance=Example(1,2)
myinstance.print_tot()
```

```
3
```

```
myinstance2=Example(4,5)
myinstance2.print_tot()
```

```
9
```

命名規約

命名規則とは、コードを書く時の変数名や関数名の付け方などのルール。命名規則に沿ってコードを書くことで、読みやすい、理解しやすいコードを書くことができる。

Python言語における変数、定数、関数、クラス、メソッドなどの命名規則(PEP-8)

関数名・変数名・メソッド名	全て小文字。複数の単語を使う場合は、単語の区切りにアンダースコア(_)を使う。	my_function
クラス名	大文字始まり。複数の単語を使う場合は、2番目以降の単語も大文字始まり。	MyClass

練習問題（宿題として提出）　コンストラクタ

Staffクラスを作成して、
基本給（5,000円）とボーナス（下の表）
の合計を返り値とする salaryメソッド を作成します。
ボーナス（bonus）は、コントラクタの第2引数とします
（bonusをインスタンス変数とする）。
以下のアルバイトの名前のインスタンスを作成し
salaryメソッドを実行せよ。

アルバイトの名前（インスタンス）	ボーナス（インスタンス変数）
suga	10,000
abe	20,000
mori	15,000

ファイル名のつけ方

39492959_0601. ipynb
学修番号_日付. ipynb

演習問題（宿題として提出）：

Healthクラスを作成して、
身長（cm）と体重（kg）から計算されるBMI値
を返り値とする cal_bmiメソッド を作成します。
身長と体重は、コントラクタの引数とします。
以下の名前のインスタンスを作成しcal_bmiメソッドを
実行せよ。

$$BMI = 体重kg \div (身長m)^2$$

名前（インスタンス）	身長（height）	体重（weight）
suga	160 cm	65 kg
abe	178 cm	75 kg
mori	170 cm	80 kg

宿題

練習問題～演習問題を1つのファイルに。
Kibacoから. ipynbファイルを提出。
テキストボックスに感想。

締め切り：6月7日

ファイル名のつけ方

39492959_0601. ipynb

学修番号_日付. ipynb