

9. クラス

□ 概要

- Pythonのクラスについて学びます。

□ 学習内容

- 9.1 クラスの定義
- 9.2 クラスの利用
- 9.3 クラスメソッド、スタティックメソッド
- 9.4 継承



9.1 クラスの定義

□ クラスとは

- 何か「モノ」を作るための「設計図」や「型」の様なもの。
- 設計図であるクラスには、属性(プロパティ)と操作(メソッド)を定義する。

9.1 クラスの定義

□ クラス定義

- classキーワードの後にクラス名を指定する。

class クラス名:

 コンストラクタやメソッドの定義

9.1 クラスの定義

□ コンストラクタ

- クラスに属性がある場合は、コンストラクタと呼ばれるメソッドで属性の定義と初期化を行う。
- self以外の引数は省略可。複数指定も可。
- 属性は「self.属性名」の形式で定義を行う。

```
class クラス名:  
    def __init__(self, 引数1, 引数2,...):  
        self.属性名1 = 引数1  
        self.属性名2 = 引数2
```

9.1 クラスの定義

□ メソッドとは

- クラスがどういう「動き」を持つのかを定義したもの。コンストラクタもメソッドのひとつ。
- self以外の引数は省略可能。複数指定することもできる。
- 戻り値も省略可能。

```
class クラス名:  
    def メソッド名(self, 引数1, 引数2,...):  
        処理  
        return 戻り値
```



9.2 クラスの利用

□ インスタンスの作成

- 作成したクラスを元にして、実際に作った「モノ」をインスタンスという。

変数名 = クラス名(引数1に渡す値, 引数2に渡す値,...)

- インスタンスを作成することで、クラスで定義した属性やメソッドを利用することができる。

変数名.属性名

変数名.メソッド名(引数1に渡す値, 引数2に渡す値,...)

9.3 クラスメソッド、スタティックメソッド

□ クラスメソッド

- メソッドに「@classmethod」を付与することで、クラスメソッドとなる。

```
class クラス名:  
    @classmethod  
    def メソッド名(cls, 引数1, 引数2,...)  
        処理
```

9.3 クラスメソッド、スタティックメソッド

□ クラスメソッドの呼び出し

- インスタンスを使用せずに、クラス名を指定するだけで呼び出すことができる。

クラス名.クラスメソッド名(引数1に渡す値, 引数2に渡す値,...)

9.3 クラスメソッド、スタティックメソッド

□スタティックメソッド

- メソッドに、「@staticmethod」を付与することで、スタティックメソッドとなる。

```
class クラス名:  
    @staticmethod  
    def メソッド名(引数1, 引数2,...)  
        処理
```

9.3 クラスメソッド、スタティックメソッド

□ スタティックメソッドの呼び出し

- クラスメソッドと同じく、インスタンスを使用せずに、クラス名を指定するだけで呼び出すことができる。

クラス名.スタティックメソッド名(引数1に渡す値, 引数2に渡す値,...)

9.3 クラスメソッド、スタティックメソッド

□ クラスメソッドとスタティックメソッドの違い

	クラスメソッド	スタティックメソッド
インスタンス変数にアクセスできるか	できない	できない
クラス変数にアクセスできるか	できる	できない (メソッド内で自クラスを宣言すればアクセス可)

9.4 継承

- 基本となるクラスを元にして、基本となるクラスの属性やメソッドを受け継いだ新しいクラスを作ることが継承という。
- 継承元となるクラスの事を、スーパークラス、継承元を受け継いで作ったクラスの事をサブクラスという。
- スーパークラスとサブクラスは、親子関係に例えることもある。（スーパークラスが親。サブクラスが子。）
- スーパークラスは、複数のサブクラスの継承元になることができる。
- サブクラスは、複数のスーパークラスを継承元として作成でき、これを多重継承という。（多重継承については後述）

9.4 継承

- 継承を利用したクラスの定義
 - クラス名の後の()にスーパークラス名を指定する。

```
class クラス名(スーパークラス名):  
    コンストラクタやメソッドの定義
```

9.4 継承

□ サブクラスのコンストラクタ

- スーパークラスのコンストラクタの呼び出しや、サブクラスのみが持つ属性を定義することができる。
- スーパークラスのコンストラクタの呼び出しは、「`super().__init__(スーパークラスのコンストラクタに渡す引数(複数指定可))`」と記述する。

```
class クラス名(スーパークラス名):  
    def __init__(self, 引数1, 引数2, 引数3):  
        super().__init__(引数1, 引数2)  
        self.サブクラスの属性 = 引数3
```

9.4 継承

□ 継承を利用したメソッドの定義

- サブクラスはスーパークラスの属性だけでなく、メソッドも受け継いでいるため、スーパークラスのメソッドをそのまま使用する場合は、サブクラスに定義しなおす必要はない。
- 以下のような場合にサブクラスにメソッドを定義する。
 - サブクラスでスーパークラスのメソッドを再定義する。
 - スーパークラスのメソッドに、サブクラスで処理を付け足し、拡張する。
 - サブクラス独自のメソッドを追加する。

9.4 継承

□ オーバーライド

- スーパークラスで定義したメソッドを、サブクラスで再定義すること。
- オーバーライドをする際は、スーパークラスで定義したメソッドと同じメソッド名で定義をする。

□ メソッドの拡張

- スーパークラスのメソッドに、サブクラスで処理を付け足すこと。
- オーバーライド同様、スーパークラスで定義したメソッドと同じメソッド名で定義をする。



9.4 継承

□ サブクラスのメソッドの呼び出し

- サブクラスのメソッドは、サブクラスのインスタンスを作成して呼び出すことができる。

変数名 = サブクラス名(引数1に渡す値, 引数2に渡す値, ...)

変数名.サブクラスのメソッド名()

9.4 継承

□ 多重継承

- サブクラスを定義する際、継承元であるスーパークラスを複数指定する。

```
class クラス名(スーパークラス名1,スーパークラス名2,...):  
    コンストラクタやメソッドの定義
```