オブジェクト指向　ピカチュウ

◆クラス Pokemon からインスタンス pokemon を作り、attack()を実行する

```
class Pokemon:
    name='ピカチュウ'
    hp=20
    atk=10
    ①
    def attack(self):
        print(f'{self.name}の攻撃！10 万ボルト')

pokemon=Pokemon()
pokemon.attack()
```

① の位置に、新しく「speed」を加えても何の変化も、エラーも起こらない。

◆oop_pokemon　実行ファイル

```
from game import Game
from pokemon import Pokemon

game=Game()
pokemon= Pokemon()
game.battle(pokemon)
```

◆game ファイル

ゲームの管理

```
class Game:
    def battle(self, pokemon):
        print(f'{pokemon.name}が現われた！{pokemon.name}の HP は{pokomon.hp}だ！')
        pokemon.attack()
```

◆pokemon ファイル

pokemon の管理

```
class Pokemon:
    name='ピカチュウ'
    hp=20
    atk=10

    def attack(self):
        print(f'{self.name}の攻撃！10 万ボルト')
```

■Lesson2　拡張

◆pokemon ファイル 2

```
class Pokemon:
    def __init__(self, name, hp, atk):
        self.name = name
        self.hp = hp
        self.atk = atk

    def attack(self):
        print(f'{self.name}の攻撃！10 万ボルト')
```

◆oop_pokemon ファイル 2

```
from game import Game
from pokemon import Pokemon

game=Game()
pokemon= Pokemon('ピカチュウ',20,10)
game.battle(pokemon)
```

■Lesson3　継承

◆pokemon ファイル３

```
class Pokemon:
    def __init__(self, name, hp, atk):
        self.name = name
        self.hp = hp
        self.atk = atk

    def attack(self):
        print(f'{self.name}の攻撃！', end='')
        self.attack_message(self):

    def attack_message(self):
        pass

class Pikachu(Pokemon):
    def __init__(self):
        super().__init__('ピカチュウ', 20,10)

    def attack_message(self):
        print('10 万ボルト！')

class Hitokage(Pokemon):
    def __init__(self):
        super().__init__('ヒトカゲ', 18, 5)  ・・・オーバーライド

    def attack_message(self):
        print('火の粉')
```

end=’’で改行をなくしている

pass は何もしない

`super().__init__('ヒトカゲ', 18, 5)`、親クラスの init メソッドを呼び出した

◆game ファイル 3

ポケモン 2 体に対応させる

```
class Game:
    def battle(self, pokemon1,pokemon2):
        print(f'{pokemon1.name}が現われた！{pokemon1.name}の HP は{pokomon1.hp}だ！')
        print(f'{pokemon2.name}が現われた！{pokemon2.name}の HP は{pokomon2.hp}だ！')
        pokemon1.attack()
        pokemon2.attack()
```

この方法だと、条件分岐はいらない

◆oop_pokemon ファイル 3

```
from game import Game
from pokemon import Pokemon

game=Game()
pikachu = Pikachu()
hitokage = Hitokage()
game.battle(pikachu, hitokage)
```

呼び出すインスタンスによって振る舞いが変わる。
このことを「多態性（ポリモーフィズム)」と呼ぶ。

■Lesson4　カプセル化