関数とは？

まず、関数について説明します。

関数とは、いろいろな「処理」をまとめて1つにしたものです。

なぜ関数があるのでしょうか？

料理で例えてみます。

例えば、いつも作るカレーがあるとします。

そのレシピを料理ロボットに記憶してもらいます。

またカレーが食べたくなったときに、

ボタン1つで作れる。

しかも、その料理ロボットは、自分も使えるし、家族も使える。

その料理ロボットが関数なのです。

関数の便利なところは色々あるのですが、3つあげてみます。

1. 同じものを2回書く必要がない

2. 1行で使い回しができる

3. 関数の中のコードを理解していなくても他の人も使うことができる

といった便利な点があります。

関数の種類

関数には2種類あります。

自分で作る関数と、Pythonがあらかじめ用意してくれている関数です。

Pythonがあらかじめ用意してくれている関数のことを「組み込み関数」といいます。

今まで使ってきたprint関数は、組み込み関数です。

print関数はたった1行で変数の中身を表示してくれますが、

print関数の中身は何行ものコードが書かれています。

もしprint関数がなければ、変数の中身を表示させたいときに、イチからそのコードを書くことになり、大変です。

しかし、print関数があることで、コードを書く必要がありませんし、たった1行で使い回すことができます。
また、print関数の中身でどんなことを書かれているか理解していなくても、print関数を使うことができます。

関数の定義、引数、戻り値

def 関数名():

実行する処理

関数を作ることを「関数を定義する」と言います。
Pythonで関数の定義の仕方を見ていきましょう。
Pythonでは関数の定義にdefを使います。
defの後に関数の名前である関数名を書き、
丸括弧を書きます。
丸括弧の中に記述するものを引数と言います。
関数は、引数を受け取ることができます。
受け取った引数は、関数内で使うことができます。

例えば、関数内にある数字と引数を掛け算することができます。

このように、関数に引数の値を渡すことで関数のできる処理の幅が広がります。

引数という言葉は、「引数を関数に渡す」と言ったり、「引数を関数が受け取る」と言ったりします。

引数は、必ず必要と言うわけではなく、省略することができます。

また、引数は、何個でも渡すことができるので、必要な分だけカンマで区切って入れることができます。

引数の丸括弧のあとに

コロンを書いて、

次の行に実行する処理を書きます。

実行する処理の行は右にインデントする必要があります。

関数の記述が

終わったら、

インデントを元に戻します。

そして、関数は、引数を受け取ることができる一方、

関数は処理結果を返すことができます。

これを戻り値といいます。

return文を使うことで、戻り値として関数の外に値を返す

ことができます。

それでは、実際に、コードを書いてみましょう。

引数なしの関数

それでは、実際に、コードを書いてみましょう。

```
def say_hello():
    print("Hello World")

say_hello()
```

引数なしの関数を見ていきましょう。

文字列の「Hello World」を表示させる関数を定義して実行

してみましょう。

まずは defを書きます。関数名を書きましょう。

関数名は、「say_hello」としましょう。

今回、引数はないので丸括弧のみを書きます。

コロンを書いて、次の行に実行する処理「print("Hello World")」を書きます。

これで関数を定義することができました。

関数を実行するには、関数名、丸括弧で実行できます。

関数名「say_hello」に引数なしの丸括弧。これで「Hello World」が表示されるはずです。

実行してみましょう。

実行結果：

```
Hello World
```

「Hello World」を表示することができました。

```
def say_hello():
    print("Hello World")
say_hello()
say_hello()
say_hello()
```

定義した関数は何度でも呼び出すことができます。

3回say_hello()を記述します。

「Hello World」は3回表示されるはずです。

実行してみましょう。

実行結果：

```
Hello World
Hello World
Hello World
```

「Hello World」を3回表示することができました。

引数ありの関数

```
def say_hello(greeting):
    print(greeting)

say_hello("Hello World")
```

次に、引数を使う場合の関数を見ていきましょう。

関数の引数を挨拶という意味のgreetingとしましょう。

受け取った引数をprint関数で表示させます。

関数の定義は終わりです。

関数にHello Worldという文字列を渡します。

では、実行してみましょう。

実行結果：

```
Hello World
```

「Hello World」が表示されました。

関数を変数へ代入

```
print(greeting)
hello = say_hello
hello("Good Morning")
```

Pythonは関数を変数に代入することができます。

先ほどと同様に定義した関数名を、そのまま変数に代入するだけです。

変数名は「hello」とします。

helloにsay_helloを代入します。

helloのあとに丸括弧、Good Morningの文字列を渡してみましょう。

実行してみます。

実行結果：

```
Good Morning
```

「Good Morning」が表示されました。

複数の引数がある関数

```
def add(num01,num02):
    print(num01 + num02)

add(6,2)
```

引数を2つ使ってみましょう。

足し算という意味の「add」という関数名を作ってみます。

num01とnum02を足してprint関数で表示してみましょう。

では、add関数に6と2を渡してみます。

結果は、5足す2に加えて、関数内の3を足すので、10となるはずです。

実行してみます。

実行結果：

8が表示されました。

return文

```
def add(num01,num02):
    return (num01 + num02)

add(6,2)
```

関数の結果は、return文で返すことができます。

printをreturnに変更してみましょう。

ただし、returnで結果を返しても、printで表示をさせていないので、確認することができません。

念のため、実行してみましょう。

何も表示されません。

```
def add(num01,num02):
    return (num01 + num02)

print(add(6,2))
```

関数を呼び出すaddの部分をprintで括ってみます。

実行してみます。

実行結果：

```
8
```

8が表示されました。

```
def add(num01,num02):
    return (num01 + num02)

add_result = add(6,2)
print(add_result)
```

結果を変数に格納して、print関数で表示させる方法もあります。「add_result」という変数に代入して表示させてみましょう。

表示させてみます。

実行結果：

```
8
```

8が表示されました。

確認問題

それでは最後に確認問題です。

3つの引数を受け取る関数を作り、9と4と2の平均を表示させてください。

一旦、動画を止めて記述してみてください。

答え合わせです。まず、defと書いて、変数名は何でもよいのですが、divという関数を作ります。

引数は3つなので、丸括弧の中に引数を3つ記述します。

そして、returnを書き、a,b,cを足します。足し算は先に計算をしたいので丸括弧で括ってそれから個数の3で割ります。

結果をdiv_resultに格納して、print関数で表示させてみましょう。

実行してみます。

```
def div(a ,b , c):
    return (a + b + c) / 3
```

```
div_result = div(9 ,4 ,2)
print(div_result)
```

実行結果：

```
5.0
```

5.0が表示されました。