

めあて：関数を作ってみよう！

☆Keywords☆

- 関数とは？
- 引数(ひきすう)とは？
- 戻り値とは？

○例題 次のプログラムをコピペして実行してみよう。【主】 (各2点×4 = 8点)

問題	プログラム
<p>パターン①</p> <p>引数も戻り値もない関数</p> <p>print("こんにちは!")というプログラムに、say_hello という名前の関数を作成し、3回表示させる。</p>	<pre>def say_hello(): #関数の名前 print("こんにちは!") #関数が行う処理 (つまりただ「こんにちは」と表示させるだけの関数) say_hello() say_hello() say_hello()</pre>
<p>パターン②引数を使う関数</p> <p>print("こんにちは、" + name + "さん!")というプログラムに、greet という関数名をつけて、name を引数とし、「自分の名前」を表示させる。</p>	<pre>def greet(name): #関数の名前、「name」は引数 print("こんにちは、" + name + "さん!") #関数が行う処理 greet("りさ") #出力: こんにちは、りささん!関数の「name」に「りさ」が引き渡される (引数)。</pre>
<p>パターン③</p> <p>戻り値だけを使う関数</p>	<pre>import random #ランダム関数を使えるように設定する。 def omikuj(): #関数に名前を付ける。 kuji=["大吉","中吉","小吉","凶"] #くじの配列(リスト)を作る。 return random.choice(kuji) #指定したリストからランダムに1つ選んで返す。 kekka=omikuj() print("結果は",kekka,"です") #omikuj()を呼び出し、戻り値を変数 kekka に保存して表示する。</pre>
<p>パターン④</p>	<pre>def add(a, b):</pre>

引数も戻り値も使う関数	<pre># 「def」で関数を定義。addが関数の名前で、a,bは引数(関数に値を渡す入り口)関数はa,bという2つの値を受け取る、と約束している。 return a + b #計算して戻り値を返す。 result = add(2, 5) #関数を呼び出す。add(2,5)で関数を呼び出し、戻ってきた値(8)が変数resultに代入される。 print(result) #resultを表示する。</pre>
-------------	---

○残りの時間で以下の課題に取り組む。【思】各3点×4=12点(減点法)

提出締め切り：次の情報の授業まで。リンクをクラスルームから提出すること。

(1)引数なし・戻り値なしの関数

関数名：say_something()

自分の好きな四字熟語を表示させるプログラムを作成してください。

(2)引数だけを使う関数

関数名：greet_two(name1, name2) 引数を2つ使って、

「こんにちは、一郎さん！」 「こんにちは、次郎さん！」

と表示させるだけのプログラムを関数を利用して作成してください。

(3)戻り値だけを使う関数

関数名：suggest_genre() 配列(リスト)：

genres = ["アクション", "コメディ", "ファンタジー", "ホラー", "SF"]

という宣言をして、

おすすめの映画ジャンルは【ランダムに選択される要素】です！

と表示されるプログラムを作成してください。

(4)引数も戻り値も使う関数

関数名：kakezan(a, b)

上の関数を使って、 4×6 を計算させるプログラムを作成してください。

○発展問題 余裕がある生徒は挑戦してみよう！

(1) 4 の 2 乗を返す関数

関数名を `square(x)` とする。 ヒント 戻り値の設定 : `return x*x`

(2) 2 つの値、10 と 20 の平均を返す関数

関数名と引数を、`average(x, y)` と設定する。

(3) 消費税(10%)とし、120 円の商品を購入するときの値段を計算してくれるプログラムを作成する。

ヒント : 最初のプログラムは以下。

```
def Taxprice(price):
```

```
ans=price*1.1
```

```
return ans
```