**Python追加　配列(一次元配列/二次元配列**

1. **Pythonの配列(リスト)**

**配列の例**

d = [ 12, 45, 8, 25]

sd = [“abc”, “de”, “fg”]

dd = [ [1, 10, 100], [2, 20, 200], [3, 30, 300]]

**配列の操作例**

d = []・・・　配列の宣言/ 空の配列の作成

d = [ 12, 45, 8, 25] ・・・　配列の宣言と定義

d.append(60) ・・・　配列への要素の追加 結果　d = [ 12, 45, 8, 25, 60]

d = [0] \* 5・・・　配列への同一要素の定義 結果　d = [ 0, 0, 0, 0, 0]

**2. 一次元配列**

課題01: 乱数を使って配列を定義。

打ち込み/動作確認

import random

d = []

for i in range(10):

d.append(random.randint(1, 100))

print(d)

左のプログラムを打ち込んで実行して、動作を確認してください。

**3．二次元配列**

操作例

dd = [ [1, 10, 100], [2, 20, 200], [3, 30, 300]]

dd = [ [1, 10, 100],

[2, 20, 200], どちらも同じプログラム

[3, 30, 300]]

|  |  |  |
| --- | --- | --- |
| dd[0][0] | dd[0][1] | dd[0][2] |
| dd[1][0] | dd[1][1] | dd[1][2] |
| dd[2][0] | dd[2][1] | dd[2][2] |

**二次元配列の添え字(インデックス)**

dd[0,1] は10, dd[1,2]は200

dd.append([4, 40, 400]) 結果　[ [1, 10, 100], [2, 20, 200], [3, 30, 300], [4, 40, 400]]

課題02: 二次元配列の定義

打ち込み/動作確認

dd = []

for i in range(3):

t = []

for j in range(3):

t.append(i \* 10 + j)

dd.append(t)

print(dd)

(実行結界)

[[0, 1, 2], [10, 11, 12], [20, 21, 22]]

左のプログラムを打ち込んで実行して、動作を確認してください。

開発

**3．実習問題**

課題03: range()を使って配列dを逆にしたリストを作成して表示するプログラムを作成しなさい。

(実行例)

[75, 70, 13, 11, 67] ・・・　dの表示

[67, 11, 13, 70, 75] ・・・　dを逆にしたリストの表示

課題04　以下の考え方を使って円周率の近似値を計算しなさい。(モンテカルロ法)

　半径 rの1/4円の面積は r \* r \* π / 4 = r2 \* 0.785 (π = 0.785 \* 4)となる

dd = [[x0, y0], [x1,y1],・・・[x999, y999]] ( x <=100, y <= 100)の1000個のランダムなxy座標の配列を用意しておいて、各点が1/4円の中に入るか確認する。  
　1/4円に入るかは原点からの距離が100以内であるかで判定でき、

　　距離は　Pythonで math.sqrt(dd[n][0]\*\*2 + dd[n][1]\*\*2) (n は0以上999以下である)

この1/4円内に入っている、[x,y]の割合の4倍がπになることからπの近似値を計算するプログラムを作成する。

import random

import math

dd = []

for i in range(1000):

dd.append([random.randint(1, 100), random.randint(1,100)])

以上はddに1～100までのx,yを1000からなる二次元リストを作るプログラム  
この後に、πを求めるプログラムを追加。