

# データサイエンティスト育成コース



## Pythonコーディングテスト練習問題

- こちらの練習問題は、入学試験で出題される問題の類題となっております。
- 実際の試験は、HackerRankというサービスで実施されます。
- 試験の採点は、関数をPythonで作成し、ユニットテストにパスすることで加点されます

### Q1 文字列の結合

以下のPythonコードを実行した場合の結果を選びなさい。

1. 4
2. 4 おはよう
3. 9 おはようございます
4. 9

```
In [ ]: a = 'おはよう'
        b = 'ございます'
        c = len(a) + len(b)
        d = a + b
        print(c, d)
```

### Q2 データ型

a = [1,1,2,2,3,4,5]というリストを集合(set)型にした場合、以下のうちどの結果になりますか？

1. {1, 1, 2, 2, 3, 4, 5}
2. {1, 2, 3, 4, 5}
3. {1, 2, 3}
4. 上記いずれでもない

### Q3 文字列の偶奇の違いにより処理を変える

以下のような名前の配列を与えられた時、名前が奇数の長さのときは「Good night」、偶数のときは「Good morning」という挨拶の配列を返す関数を作ってください。 例:

例)

入力: ['Myung', 'Petrucci', 'LaBrie', 'Rudess']

出力: ['Good night', 'Good morning', 'Good morning', 'Good morning']

```
In [ ]: def greeting(names):
        result = []
        # ここにコードを書いてください。
        # 戻り値はリストにしてください。
        return result
```

```
In [ ]: names = ["Myung", "Petrucci", "LaBrie", "Rudess"]
        greeting(names=names)
```

## Q4 偶数だけを取り出す関数

与えられた配列 a から偶数だけ取り出し、偶数だけのリストを返す関数を書いてください。

例)

入力: [1,2,3,4,5,6,7,8,9,10]

出力: [2,4,6,8,10]

```
In [ ]: def get_even_number(numbers):  
        result = []  
        # ここにコードを書いてください。  
        # 戻り値は配列にしてください。  
        return result
```

```
In [ ]: numbers = [1,2,3,4,5,6,7,8,9,10]  
        get_even_number(numbers=numbers)
```

## Q5 面積が一番大きい長方形のインデックスを返してください。

複数の長方形の短辺と長辺を与えたとき、一番面積が大きい長方形のインデックスを返してください。

インデックスとは行番号（Pythonは0から始まるのを忘れなく！）のことです。

また、各要素の左側は短辺、右側は長辺です（以下のrectsのリストの1行目ですと、2.1が短辺、4.1が長辺となります）。

```
In [ ]: rects = [  
        [2.1, 4.1],  
        [0.5, 1.1],  
        [2.5, 3.2]  
        ]
```

```
In [ ]: def get_largest_rect(rects):  
        #returnの後のNoneは消して、ここにコードを書いてください。  
        #戻り値は、数字(インデックス)であることに注意しましょう。(面積ではありません)  
        return None
```

```
In [ ]: get_largest_rect(rects=rects)
```

## Q6 コインが表が出る確率を求めよう

表が出る確率が10%（ということは裏が出る確率は90%）のコインがあります。

コインをN回投げたときに、表がx回出る確率を求めてください。

答えは、小数点第5位を四捨五入して、第4位まで答えてください。例：0.12345 であれば、0.1235にしてください。

**ヒント1: 確率を求める数式は以下のとおりです。**

$$N! / ((N-x)! * x!) * (0.3)^{x} * (0.7)^{(N-x)}$$

**ヒント2: 階乗を求めるときは以下のように書きます**

```
num = 1  
for i in range(N, 0, -1):  
    num = num * i
```

**ヒント3: 小数点第5位を四捨五入する場合は、以下のように書きます。**

```
round(0.12345, 4)
```

```
In [ ]: def coin_toss(N, x):  
        #returnの後のNoneは消して、ここにコードを書いてください。  
        #戻り値は、確率であることに注意しましょう。  
        return None
```

```
In [ ]: coin_toss(N=10, x=4)
```

## Q7: 数字を並び替えよう

ソートアルゴリズムはいくつかありますが、その中で**バブルソート** (<http://wa3.i-3-i.info/word14886.html>)を実装してみてください。

アルゴリズムは簡単で隣り合う数字を比べて左の数字より右の数字が大きければ入れ替えていき、並び替えるまで繰り返します。

例は以下のとおりです。

アルゴリズムの動き方) 入力を a = [5,4,1,3,2] とします。最終的に欲しい結果は、[1,2,3,4,5]です。

では、アルゴリズムの動きを見ていきましょう。

- 1 巡目では、まず、5,4 のペアを比べて評価します。
2. すると、左の数字のほうが右の数字よりも大きいので入れ替えます。この時点で[4,5,1,3,2]になります。
3. 続いて、[5,1]を比べます。ここでも入れ替えます。上記の1から3のステップを繰り返し1順目が終了します。すると、[4,1,3,2,5]になります。

2巡目は、[4,1]をまず比較して、[1,4]に入れ替えます。

同じように、左から右に配列の要素を見ていきながら繰り返し2巡目が終了します。

```
In [ ]: num_list = [5,4,3,2,1]
```

```
In [ ]: def bubble_sort(num_list):  
        # バブルソートを実装してリストの形で出力してください。  
        return num_list
```

```
In [ ]: bubble_sort(num_list=num_list)
```