第3回
変数とデータ型2

（リスト、辞書）

目次

リスト

- リストは複数のデータを順番に入れておくことのできるデータ型です。
  - 文字列型では文字を順番に入れておくことができましたが、リストでは、どのようなデータ形式でも要素として持つことができます。
  - 文字列型と同様に、要素の順番はインデックスで指定ができます。

インデックス	0	1	2	3	4	5
要素	"太郎"	60	165	"男性"	1996	7

  - 文字列型と同様に＋や＊、inなどの演算子やメソッドが使えます。

- リスト用に、新しくソースファイルを作成する。
  - Spyderのメニューから[ファイル]-[新規ファイル]を選択して、新しいソースファイルを作成します。

リストを定義する

- リストを定義するには角括弧[]を使う
  - 構文：変数名 = [要素, 要素, 要素, ・・・]
  - インデックスを指定してリストの要素を取り出す際にも、文字列と同じように[]を使う
- 例：

インデックス	0	1	2	3	4	5
要素	"太郎"	60	165	"男性"	1996	7

```
 8 a = ["太郎", 60, 165, "男性", 1996, 7]
 9 print(a)
10 print(a[3])
```

何が出力される？

  - インデックスとしてマイナスの整数を指定すると、最後から数えた要素を取り出せます。

```
10 print(a[-1])
```

何が出力される？

スライス

- スライスという機能を使うと、リスト内の複数の要素をインデックスの範囲を指定して取り出すことができます。
  - スライスの範囲指定は、開始インデックスと、終了インデックスに1を加えた整数をコロン(:)で区切って指定します。
    構文:変数名[開始インデックス:終了インデックス＋1]
  - 例:リストの一部を取り出して、別のリストを作る例

インデックス	0	1	2	3		
要素	"太郎"	60	165	"男性"		

ここだけ取り出して別のリストbを作る

```
 8 a = ["太郎", 60, 165, "男性", 1996, 7]
 9 print(a)
10 print(a[-1])
11 b = a[1:3]
12 print(b)
```

  - 開始インデックスが先頭の場合、開始インデックスの指定を省略できます。同様に、終了インデックスが末尾の場合も省略できます。

```
11 b = a[:3]
12 print(b)
```

```
13 b = a[3:]
14 print(b)
```

何が出力される？

リストへの要素の追加

- 定義済みリストに要素を追加する様々な方法が用意されています。
  - ＋演算子を使った方法
    - 文字列の連結と同様に、リストにおける＋演算子は連結を表します。
    - 連結できる対象はリスト同士です。下記の左の例では、エラーになります。

×

```
16 a = [1, 2, 3, 4]
17 b = a + 1
18 print(b)
```

○

```
16 a = [1, 2, 3, 4]
17 b = a + [1]
18 print(b)
```

```
[1, 2, 3, 4, 1]
```

○

```
16 a = [1, 2, 3, 4]
17 b = a + [1,-1]
18 print(b)
```

```
[1, 2, 3, 4, 1, -1]
```

  - appendメソッドを使うと、自分自身に要素を追加できます。
    - 構文:リスト.append(追加要素)
    - 追加要素として、リストを指定した場合には、リスト全体が要素として追加されるので注意が必要です。リストを要素としてではなく、中身を追加したい場合は、extendメソッドを使います。

```
16 a = [1, 2, 3, 4]
17 a.append(5)
18 print(a)
```

```
[1, 2, 3, 4, 5]
```

```
16 a = [1, 2, 3, 4]
17 a.append([5, 6])
18 print(a)
```

```
[1, 2, 3, 4, [5, 6]]
```

extendを使うと？

リストの要素の上書きと削除

- 要素の上書き
  - 定義済みのリストの要素を上書きするには、インデックスを指定して代入します。
  - 例：

```
16 a = [1, 2, 3, 4]
17 a[1] = 100
18 print(a)
```

```
[1, 100, 3, 4]
```

- 要素の削除
  - リストの要素を削除するには、del文を使います。
    構文：del リスト[インデックス]
  - 例：

```
16 a = [1, 2, 3, 4]
17 del a[1]
18 print(a)
```

```
[1, 3, 4]
```


リストの要素の検索その他

- 文字列と同様に、要素の包含チェックとしてin演算子、検索にindexメソッドが使えます。
  - 例：

```
16 a = [300, 100, 400, 200]
17 b = 300 in a
18 print(b)
19 c = a.index(300)
20 print(c)
```

何が出力される？

- max関数とmin関数を使うと、要素の最大値、最小値を求めることができます。
  - 例：

```
16 a = [300, 100, 400, 200]
17 b = min(a)
18 print(b)
```

```
16 a = [300, 100, 400, 200]
17 b = max(a)
18 print(b)
```

- sort関数を使うと要素を小さい順に並べ替えできます。reverse関数を使うと要素の並びを逆にできます。
  - 例：

```
16 a = [300, 100, 400, 200]
17 a.sort()
18 print(a)
```

```
16 a = [300, 100, 400, 200]
17 a.reverse()
18 print(a)
```

ディクショナリ(辞書)

- ディクショナリは複数のデータをキーと呼ばれる見出しを使って管理できるデータ型です。
  - インデックスで要素を指定するのではなく、キーと呼ばれる数値や文字列で要素を指定できるのが特徴です。

キー	値(バリュー)
住所	静岡県静岡市
氏名	山田太郎
年齢	26

- ディクショナリ用に、新しくソースファイルを作成する。
  - Spyderのメニューから[ファイル]-[新規ファイル]を選択して、あたらしいソースファイルを作成します。

ディクショナリを定義する

- ディクショナリを定義するには波括弧{}を使う
  - 構文:変数名 = {キー1: 値1, キー2: 値2, ・・・}
  - キーと値の組みはコロン(:)で区切って記述します。
  - 値を取り出す際は[]の中に取り出したい値のキーを指定します。
  - 例:

キー	値(バリュー)
住所	静岡県静岡市
氏名	山田太郎
年齢	26

```
8 a = {"住所":"静岡県静岡市", "氏名":"山田太郎", "年齢":26}
9 print(a["氏名"])
```

何が出力される?

  - 存在しないキーを指定すると、エラーになります。

キーと値の追加、削除

- キーと値の追加
  - ディクショナリに新たなキーと値を追加するには、新しいキーを指定して代入を行います。
  - 例：
```
a = {"住所":"静岡県静岡市", "氏名":"山田太郎", "年齢":26}
a["体重"] = 60
print(a)
```

- キーを使って値を削除する
  - リストと同様にdel文を使います。
  - 例：
```
a = {"住所":"静岡県静岡市", "氏名":"山田太郎", "年齢":26}
del a["年齢"]
print(a)
```


キーの存在チェックと一覧

- 特定のキーが存在しているか確認するにはin演算子を使います。
  - 例：
```
a = {"住所":"静岡県静岡市", "氏名":"山田太郎", "年齢":26}
b = "年齢" in a
print(b)
```

True

- ディクショナリ内のキーの一覧を取得するにはkeysメソッドを使います。
  - ただし、取得した一覧はイテレータという特殊な形式なので、例えばリスト形式で取得したければ、list関数でリストに変換します。
  - 例：
```
a = {"住所":"静岡県静岡市", "氏名":"山田太郎", "年齢":26}
b = a.keys()
c = list(b)
print(c)
```

課題2-1：BMIの計算

- BMIを計算する
  - BMIとは、Body Math Indexの略で、体重と身長から太っているか痩せているかを判断する指標です。
    - BMIが25.0以上：太っている
    - BMIが18.5以上25.0未満：普通
    - BMIが18.5未満：痩せている
  - BMIの計算式は　BMI = 体重（kg）÷身長（m）÷身長（m）です。
    - 下記の囲みを追加して、さらに[ここを考える]を完成させて、BMIを計算してください。

```
 8 taijyu = 60
 9 print(taijyu + 5)
10 shincho = 1.6
11 bmi = [ここを考える]
12 print(bmi)
```

    - 身長や体重を自分や架空の数値に置き換えて、計算してみてください。

課題2-2：数値型の課題

- A案、B案、C案があり、それぞれの案を支持する人数を集計したところ、A案が142名、B案が46名、C案が78名となりました。
- A案を支持する人数は、全体の何パーセントか計算して、結果を表示してください。

```
#A案、B案、C案の人数
A = 142
B = 46
C = 78

#A案を支持するパーセントは？
```

課題2-3：文字列型の課題

- 課題2-2の結果のパーセントの数値の右に「パーセント」と表示してください。

```
#A案、B案、C案の人数
A = 142
B = 46
C = 78

#A案を支持するパーセントは？
```


課題2-4：リストの課題

- リストxの末尾に、リストyの内容を追加してください。
- 追加後、リストxの要素の並び順を逆にしてください。
- 最後に、リストxの内容をprint関数で表示してください。

```
x = [1, 2, 3, 4, 5]
y = [6, 7]
```

課題2-5：ディクショナリの課題

- input関数は、キーボードから入力した文字を取得できる関数です。下記の例では、入力した文字がkeyに代入されます。
- keyとして「車名」や「メーカー」「年式」が入力されたと仮定して、carディクショナリから対応する値を表示してください。

```
car = {"車名": "プリウス", "メーカー": "トヨタ", "年式": 2021}
key = input("知りたいキー名を入力してEnterキーを押してください。")
```


応用問題

```
kanji = ["静岡", "浜松", "沼津"]
yomi = {"静岡": "しずおか", "浜松": "はままつ", "沼津": "ぬまづ"}
n = input("0～2までの数字を入力してEnterキーを押してください。")
```

- 上記コードが与えられたとします。nに0が入力されたならば「静岡（しずおか）」、1ならば「浜松（はままつ）」、2ならば「沼津（ぬまづ）」と出力してください。
  - 解説とヒント：
    - input関数は、キーボードから文字を入力できる関数です。上記の例では、入力された文字がnに代入されます。
    - nに代入されるのは数値ではなく文字なので、リストのインデックスとして用いるためには数値に変換する必要があります（int関数を使う）。

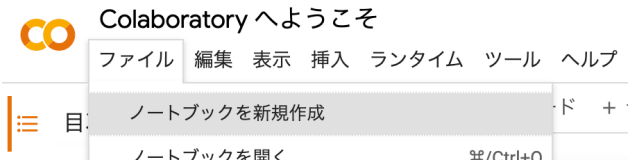
インストールレスのPython環境の紹介（Google Colaboratory）

- Webブラウザを使ってPythonプログラムの作成と実行ができます
  - 要：Googleアカウント
- Edge/Chrome/Safariを起動する
  - google colab で検索する
  - 下記リンクを開く

https://colab.research.google.com › notebooks › intro

Colaboratory - Google Colab

  - 「ファイル」「ノートブックを新規作成」



ノートブック

- Google Colaboratoryでは、ノートブックという単位でプログラムを記述します。
- セルという枠にコードを入力します。
  - コード以外にもテキスト（メモ書き）を入力するセルを追加できます。
- コードを実行するには、セル左の ▶ ボタンか、下記のキーを押します。
  - 実行：Ctrl+Enter
  - 実行と新たなセルの追加：Shift+Enter