リスト・ループ⇒使ってみよう

リスト　[1, 3, 4, 7, 11]

空のリスト　[]

文字列のリスト ['a', 'bc']

リストを作って変数に代入

x = [1, 3, 4, 7, 11]

s = ['a', 'bc', 'def']

要素のアクセス
[<位置>]

たとえば
x[0] ～ 0番目の要素

問題

x = [1, 3, 4, 7, 11]
のとき　x[1] は？

s = ['a', 'bc', 'def']
のとき　s[0] は？
では　　s[3] は？

print(x[1]) など
試してみよう

リスト・ループ⇒使ってみよう

リストの切出し (スライス)
［〈位置〉：〈位置〉］

位置は要素と要素の間にある

```
x = [1, 3, 4, 7, 11]
    ↑  ↑  ↑  ↑  ↑  ↑
    0  1  2  3  4  5
```

x[1:3] は
⇒#1の位置 (1要素目の前) から
#3の位置 (3要素目の前) まで

```
x = [1, 3, 4, 7, 11]
    ↑  ↑  ↑  ↑  ↑  ↑
    0  1  2  3  4  5
```

s = ['ab','cd','ef'] のとき
s[1:3] は？

```
s = ['ab','cd','ef']
    ↑   ↑   ↑   ↑
    0   1   2   3
```

問題
x = [1, 3, 4, 7, 11] のとき
x[2:3] は？
x[2:2] は？
x[3:1] は？

print(x[1:3]) など
試してみよう

リスト・ループ⇒使ってみよう

端を書かないスライス

x[1:] は x[1:最後] と同じ

x = [1, 3, 4, 7, 11]
のとき x[1:] は？

逆向きの≪場所≫

x[-2] は x[後ろから2番目]

x = [1, 3, 4, 7, 11]
-5 -2 -3 -2 -1

x[-2] は 7

x[-4] は 3

逆向きの利用

最後の要素は？

x[-1]

最後の2要素のリストは？

x[-2:]

問題

x = [1, 3, 4, 7, 11] のとき
x[3:] は？
x[:3] は？
x[-2:] は？
x[:-2] は？

print(x[3:]) など
試してみよう

リスト・ループ⇒使ってみよう

スライスに第3パラメタが…

x[〇:〇:◎]

x=[1, 2, 3, 4, 5, 6] のとき

x[:] は？

x[::2] は？

間隔2で (1つおきに) 取る

x[::2] は [1, 3, 5]

x[::−2] は？

逆方向に間隔2で取る

x[::-2] は [6, 4, 2]

x[::−1] は？

print(x[::-1]) など
試してみよう

要素の値を書き換える

要素に代入する

x[3] = 8

⇒ x = [1, 2, 3, 8, 5, 6]

リストの後ろに追加する

x.append(9)

⇒ x=[1, 2, 3, 4, 5, 6, 9]

2つのリストを結合する

x.extend([7,8])

⇒ x = [1, 2, 3, 4, 5, 6, 7, 8]

又は x + [7,8]

リスト・ループ⇒使ってみよう

この他いろいろな操作がある

https://docs.python.jp/3/tutorial/introduction.html#lists
https://docs.python.jp/3/library/stdtypes.html#sequence-types-list-tuple-range
必要なときにマニュアルを見ればいい

便利な操作の例

u in s　　if 3 in [1,3,5]:
u not in s

s * n　　[1,3] * 3
⇒ [1,3,1,3,1,3]

len(s)　　len([1,3,5]) ⇒ 3

min(s)　　min([1,3,5]) ⇒ 1

max(s)　　max([1,3,5]) ⇒ 5

s.count(x)　s中のxの数
[1,3,5,3,5,3].count(5) ⇒ 2

list(range(n)) はリスト
[0, 1, 2, … , n-1] を作る

list(range(m, n)) はリスト
[m, m+1, … , n-1] を作る

list(range(m, n, p)) はリスト
[m, m+p, … , n-1以下]

長さ10で全部0のリスト
[0, 0, 0, … , 0]
を作ってみよう

リスト・ループ⇒使ってみよう

実は文字列は、文字を要素としたリストに見える

```
s = 'abcde'
print(s[2]) ?

t = 'pqr'
print(s+t) ?

u = s[1:3] + t[-2:]
print(u) ?
print(len(u)) ?

print(s[ : :-1]) ?

v = 'a' * 5
print(v) ?
```

```
print('b' in s)
if 'b' in s:
    print('found')

w = 'This is a pen.'
print(s.count('s'))
```

⇒ 結果は 2

print(s[::-1]) など
試してみよう

リスト・ループ⇒使ってみよう

ループに慣れよう

```
for u in [1,2,3]:
    print(u)

for u in ['Taro','Jiro']:
    print(u)

for u in 'abc':
    print(u)

for u in range(2,5):
    print(u)
```

10回繰り返すにはどうすればよいか？

上記を試してみよう

```
s = 'This is a pen.'
for u in range(len(s)):
    print(s[u])
```

```
s = 'This is a pen.'
for u in s:
    print(u)
```

試してみよう

```
count = 0
for u in s:
    if u==_____:
        count = _________
print(count)
```

テキストファイルの読み込み

知識　　テキストファイルとは

文字だけのファイル
　⇔文字以外も(何でもいい)ファイル

コンピュータの中は「何でも０/１」
　文字も0/1　'a'は 0110 0001
　　(英数字は7桁、仮名漢字は24桁)

文字だけ
　⇒ 英数字はA～Zとa～zの26x2文字
　7桁使う　0～1111111(127)
　　余っている？
　英数字の他に記号とか制御文字
　　空白、'!'、','、';'、':'など
　　改行、バックスペース、抹消など

テキストファイルのイメージは？

hello
Good day.

h e l l o NL G o o d _ d a
y . NL

改行文字で行が区切られる

文字以外でも何でも書けるファイル
　⇒「バイナリファイル」
　↓
　行区切りがない (改行文字がない)
　　⇒ 行の概念がない

テキストファイルの読み込み

ファイルのオープン・クローズ操作

ファイルは、オープンしてから使う

オープン open は利用の準備

```
f = open('testfile.txt', 'r')
```

'testfile.txt'：ファイルの名前

'r'：モード

→ ファイルの使い方を指定する

- 'r'　read　読出し用にopen
- 'w'　write　書込み用にopen (切詰め)
- 'a'　append　書込み用、既に存在する場合は末尾に追記する
- '+'　更新用に開く (読みr+・書きw+)

使い終えたらクローズするのが建前

クローズ close は終了・後始末

```
f.close()
```

→ f は前に open したファイル

プログラム終了時に自動的にクローズ

より安心な with open(...) as

```
with open('testfile.txt','r') as f:
    data = f.read()      読む
```

＝

```
f = open('testfile.txt','r')
data = f.read()      読む
f.close()
```

よく忘れる

テキストファイルの読み込み

テキストの読み込み方のいろいろ

`data = f.read()`

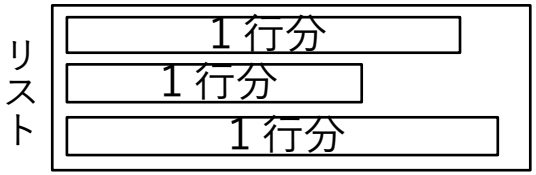
全体を文字列として読込む

⇩

自分で行ごとに分割

`lines = f.readlines()`

全体を１行ずつに分けたリストとして読込む



`line = f.readline()`

次の１行だけ読込む
ファイル全部を読むにはループを書く必要がある

```
with open('testfile.txt','r') as f:
    mylist = f.readlines()
for line in mylist:
    print(line)
```

リストから１つずつ取出す

とか、先頭の５行だけprintしたければ

```
for line in mylist[:5]:
    print(line)
```

試してみよう

Jupyter Notebook上でテキストファイルを作ろう　←　「新規」で「テキスト」
名前を「rename」で test.txt にしよう

test.txt の内容を書き込もう　たとえば

吾輩は猫である。
名前はまだ無い。
どこで生まれたかとんと見当もつかぬ。

左の要領で test.txt の内容を表示するプログラムを作って試そう

TAチェック