



# Python 講座

【第 1 回 #1】演習解答例

# 演習 #1 ・ 解答例

```
sandbox.py - D:/Storage/!U
File Edit Format Run Options W
print('C: \ちくわ大明神\
B: \誰だ今の\')
```

← シングルクォーテーション 3 個で囲む

```
sandbox.py - D:/Storage/!Uncategorized/sa
File Edit Format Run Options Window Help
print('C: \ちくわ大明神\
\nB: \誰だ今の\')
```

¥n で改行する →

シングルクォーテーション本体は  
¥'  
と書いてあげる

# 演習 #2 ・ 解答例

```
*sandbox.py - D:/Storage/!Uncatec
File Edit Format Run Options Window
x = int(input('1個目の値は? '))
y = int(input('2個目の値は? '))
answer = x + y
print('和は {} です'.format(answer))
```

input() の結果は  
int() で整数にしてあげる

文字列とともに表示するには  
format() を使う

※Advanced:

実数に対応させるには、float() を int() の代わりに使う

# 演習 #3 ・ 解答例 (1/2)

```
sandbox.py - D:/Storage/!Uncategorized
File Edit Format Run Options Window Help
favorites = ['狐', '音ゲー', 'C++', '英語']
number = int(input('何番目? (1-4): '))
print(favorites[number - 1])
|
```

- 3 行目で number-1 にする理由は？
  - 先頭にアクセスするには [0]、末尾は [3]。入力をそのままインデックスに使うと、[4] という範囲外のものを見に行こうとしてしまうため。

# 演習 #3 ・ 解答例 (2/2)

- 余力のある人の 3 行目の例

```
print(  
    '{ }が{ }番目に好き'.format(favorites[number - 1], number))↵
```

```
print(str(favorites[number - 1]) +  
    'が{ }番目に好き'.format(number))↵
```

- 後者で `str` 関数を使う理由は？
  - 「俺は数字が好きだ！」っていう人の `favorites` は、`favorites = [64, 256, 572, 65535]` かもしれない。
  - はじめに文字列に変換しないと、数値と文字列を足してしまうことに！