

kubo-aoi / ProjExD Public

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#)

main

...

[ProjExD](#) / [ex01](#) / [alphabet.py](#) / <> Jump to

kubo-aoi main #7

History

1 contributor

56 lines (45 sloc) 1.46 KB

...

```
1 import datetime
2 import random
3 import string
4
5 tcount = 10
6 kcount = 2
7 max = 5
8
9 def main():
10     start = datetime.datetime.now()
11     for i in range(max):
12         seikai = shutudai()
13         correct = kaitou(seikai)
14         if correct == 1:
15             break
16     end = datetime.datetime.now()
17     print(f"所要時間 : {(end-start).seconds}秒かかりました")
18
19
20 #問題を表示する関数
21 def shutudai():
22
23     eng = string.ascii_uppercase
24     teng = [random.choice(eng) for i in range(tcount)]
25     print(f"対象文字は : {teng}")
26
27     keng = [random.choice(teng) for i in range(kcount)]
28
29     for i in range(kcount):
30         teng.remove(keng[i])
31     heng = teng
32     print(f"表示文字は : {heng}")
33     print(f"欠損文字は : {keng}")
34     return keng
35
36
37 #解答を表示する関数
38 def kaitou(seikai):
```

```
39     ans = int(input("欠損文字はいくつあるでしょうか？ : "))
40     if ans != kcount:
41         print("不正解です、またチャレンジしてください")
42         return 0
43
44     else:
45         print("正解です！それでは具体的に欠損文字を1つずつ入力してください")
46         for i in range(kcount):
47             first = input(f"{i+1}文字目を入力してください : ")
48             if first not in seikai:
49                 print("不正解です、またチャレンジしてください")
50                 return 0
51             seikai.remove(first)
52         print("正解です")
53         return 1
54
55 if __name__ == "__main__":
56     main()
```