

Instituto Superior de Engenharia de Lisboa
Licenciatura em Engenharia Informática e Multimédia

Aplicação para Exercícios Python

Manual de Utilização

Setembro, 2020

Índice

1	Motor de Busca	2
2	Exercício Python	3
3	Gerar Versões	5
4	Diferenças de Versões	6
5	Visualizar Versões	8

1 Motor de Busca

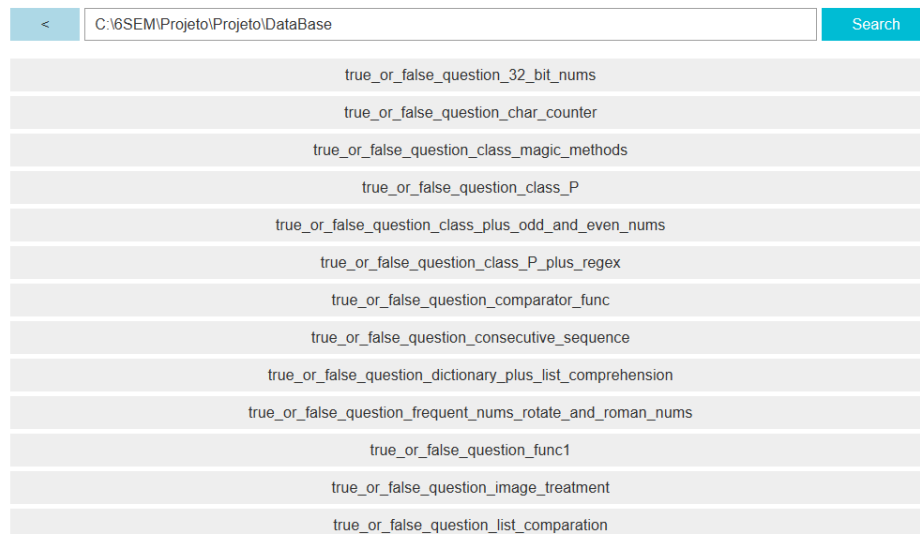


Figure 1: Motor de Busca

O Motor de Busca permite navegar pelas diretorias do seu computador. Pode chegar a uma pasta ao inserir a sua diretoria na caixa de texto e premir o botão "Search" ou chegar até ela clicando nas pastas das diretorias anteriores.

Sempre que pretender retroceder , tanto no motor de busca como ao longo da aplicação basta clicar no botão superior esquerdo com o simbolo "<".

2 Exercício Python

[<](#) [Generate Versions](#) [Versions Differences](#) [See Versions](#)

QUESTION: "true_or_false_question_working_with_lists"

Considere o programa, Python 3, que se segue. A função `random_string_generator()` gera uma string com caracteres aleatórios, o segundo ciclo *for* do código apresentado adiciona em cada iteração um valor inteiro a lista `a` e uma string a lista `b`, o terceiro ciclo *for* a cada iteração multiplica o elemento da lista `a` por `_33`.

```
from random import seed
from random import choice
from random import randint
from string import ascii_letters

seed(135)

a = []
b = []
def random_string_generator():
    l = ""
    for k in range(randint(1,15)):
        l += choice(ascii_letters)
    return l

for i in range(10):
    a.append(randint(0, 50))
    b.append(random_string_generator())

d = a

for i in range(20):
    a[i%len(a)] = a[i%len(a)] * 3
```

Acrescente a este programa o código que lhe permita indicar se as afirmações seguintes são verdadeiras ou falsas. Indique se é verdadeiro ou falso.

Answer 1:

A função `a.func1` e a função `a.func2` produzem o output diferente. **True**

A função `a.func1` e a função `a.func2` produzem o output igual. **False**

Answer 2:

A função `ab.func3` organiza os elementos da lista `ab` de forma 22. **True**

A função `ab.func3` organiza os elementos da lista `ab` de forma 222. **False**

Answer 3:

O elemento da lista `a`, no índice 3, é 33. **True**

O elemento da lista `a`, no índice 3, é 333. **False**

Answer 4:

A lista `a`, após o segundo ciclo de *for*, aumenta o valor de cada elemento 44 vezes. **True**

A lista `a`, após o segundo ciclo de *for*, aumenta o valor de cada elemento 444 vezes. **False**

Answer 5:

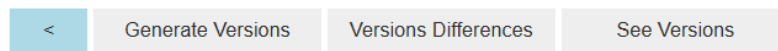
Os 5 `func5` elementos da lista `a`, podem ser apresentados da seguinte maneira 55. **True**

Os 5 `func5` elementos da lista `a`, podem ser apresentados da seguinte maneira 555. **False**

Figure 2: Exercício Python

Quando uma pasta que contém um exercício é selecionada a aplicação exibe uma página com o exercício completo, enunciado e alíneas. E na barra de cima as opções que poderá efetuar para aquele exercício, seja **gerar versões**, observar as **diferenças entre versões** ou as **versões criadas**.

3 Gerar Versões



QUESTION: "true" false_question_working_with_lists"

Considere o programa, Python 3, que se segue. A função `random\string_generator()` gera uma string com caracteres aleatórios, o segundo ciclo *for* do código apresentado adiciona em cada iteração um valor inteiro a lista `a` e uma string a lista `b`, o terceiro ciclo *for* a cada iteração multiplica o elemento da lista `a` por `_33`.

```
from random import seed
from random import choice
from random import randint
```

Figure 3: Gerar Versões

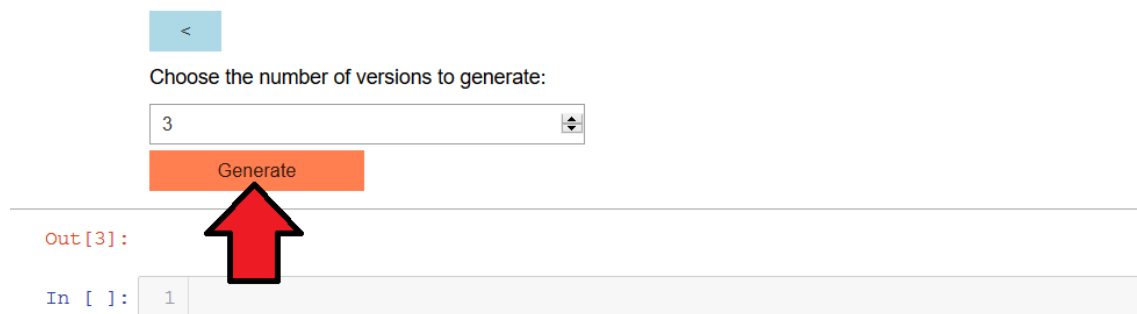


Figure 4: Gerar Versões 2

Para gerar versões do exercício selecione a opção "Generate Versions" (figura 3). Esta irá abrir uma nova janela onde pode escolher o número de versões que pretende, pode depois gerar as mesmas clicando no botão "Generate" (figura 4).

Quando a geração terminar a aplicação irá retornar á página do exercício.

4 Diferenças de Versões



Considere o programa, Python 3, que se segue. A função `random\string_generator()` gera uma string com caracteres aleatórios, o segundo ciclo *for* do código apresentado adiciona em cada iteração um valor inteiro a lista `a` e uma string a lista `b`, o terceiro ciclo *for* a cada iteração multiplica o elemento da lista `a` por `_33`.

```
from random import seed
from random import choice
from random import randint
from string import ascii_letters
```

Figure 5: Diferenças de Versões 1

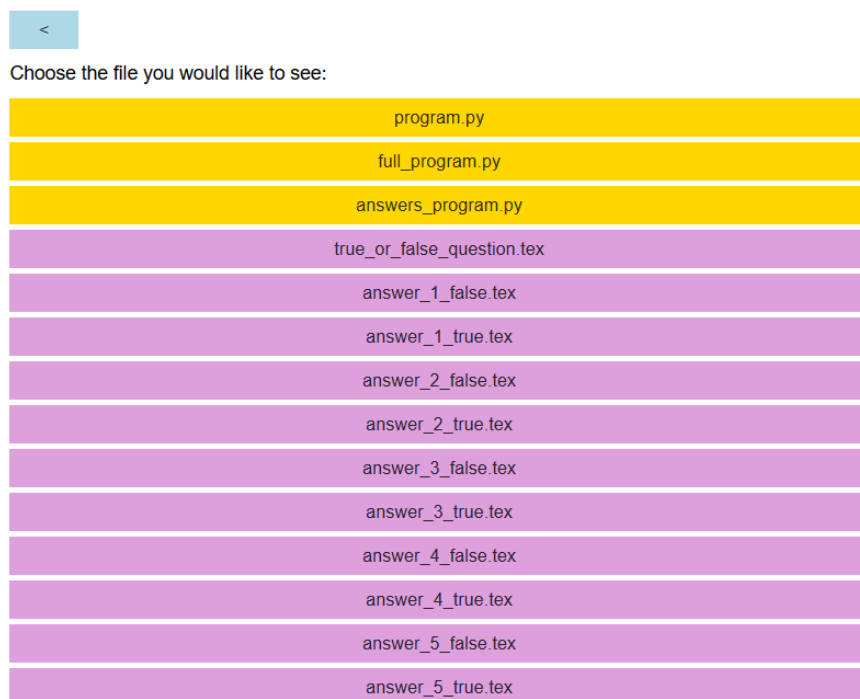


Figure 6: Diferenças de Versões 2



```
program.py

from random import seed
from random import choice
from random import randint
from string import ascii_letters

seed(1120117)

o = []
l = []
def random_string_generator():
    v = ""
    for b in range(randint(1,15)):
        v += choice(ascii_letters)
    return v

for t in range(18354):
    o.append(randint(0, 104))
    l.append(random_string_generator())

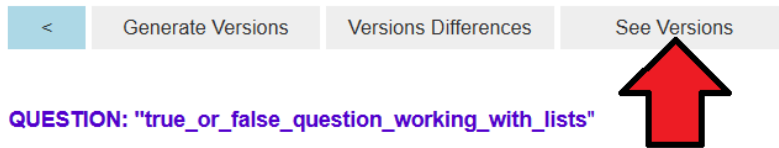
k = o

for t in range(55062):
    o[t%len(o)] = o[t%len(o)] * 3
```

Figure 7: Diferenças de Versões 3

Para observar as diferenças entre versões selecione a opção "Version Differences" (figura 5), está irá exibir uma página com todos os ficheiros que podem ser alterados por estas versões (figura 6). Escolha o script que pretende ver e aplicação irá mostrar o ficheiro escolhido com os tokens que sofrerão alteração a vermelho.

5 Visualizar Versões



Considere o programa, Python 3, que se segue. A função `random\string_generator()` gera uma string com caracteres aleatórios, o segundo ciclo *for* do código apresentado adiciona em cada iteração um valor inteiro a lista `a` e uma string a lista `b`, o terceiro ciclo *for* a cada iteração multiplica o elemento da lista `a` por `_33`.

```
from random import seed
from random import choice
from random import randint
from string import ascii_letters

seed(135)
```

Figure 8: Visualizar Versões 1



Figure 9: Visualizar Versões 2

QUESTION: "true_or_false_question_working_with_lists", version 1

Considere o programa, Python 3, que se segue. A função `random_string_generator()` gera uma string com caracteres aleatórios, o segundo ciclo *for* do código apresentado adiciona em cada iteração um valor inteiro a lista `o` e uma string a lista `l`, o terceiro ciclo *for* a cada iteração multiplica o elemento da lista `o` por 3.

```

from random import seed
from random import choice
from random import randint
from string import ascii_letters

seed(1120117)

o = []
l = []
def random_string_generator():
    v = ""
    for b in range(randint(1,15)):
        v += choice(ascii_letters)
    return v

for t in range(18354):
    o.append(randint(0, 104))
    l.append(random_string_generator())

k = o

for t in range(55062):
    o[t%len(o)] = o[t%len(o)] * 3

```

Acrescente a este programa o código que lhe permita indicar se as afirmações seguintes são verdadeiras ou falsas.

Indique se é verdadeiro ou falso.

Answer 1:

A função `o[::-1].sort()` e a função `o.sort(reverse=True)` produzem o output diferente. **True**

A função `o[::-1].sort()` e a função `o.sort(reverse=True)` produzem o output igual. **False**

Answer 2:

A função `l.sort(key=len, reverse=True)` organiza os elementos da lista `l` de forma crescente com base no tamanho da string. **True**

A função `l.sort(key=len, reverse=True)` organiza os elementos da lista `l` de forma decrescente com base no tamanho da string. **False**

Answer 3:

O elemento da lista `o`, no índice 26, é 756. **True**

O elemento da lista `o`, no índice 26, é 757. **False**

Answer 4:

A lista `o`, após o segundo ciclo de *for*, aumenta o valor de cada elemento 27 vezes. **True**

A lista `o`, após o segundo ciclo de *for*, aumenta o valor de cada elemento 28 vezes. **False**

Answer 5:

9

Os 3 últimos elementos da lista `o`, podem ser apresentados da seguinte maneira `o[-3:]`. **True**

Os 3 últimos elementos da lista `o`, podem ser apresentados da seguinte maneira `o[-3]`. **False**

Para visualizar as versões geradas selecione a opção "See Versions" (figura 8), a aplicação irá abrir uma nova janela com as versões criadas (figura 9), se pretende observar o enunciado e as alíneas de uma versão clique na mesma (figura 10).