

Guião 5 – Iteráveis, Iteradores e Geradores

Objectivos:

- Objetos Iteráveis e Iteradores
- Geradores

1. Utilizando *List Comprehension*, crie uma função leia o conteúdo de um ficheiro para uma lista (linha a linha).

2. Repita o ponto anterior, mas filtrando (removendo) as linhas que começam por com '#'.

3. O Utilizando os geradores (como nas *List Comprehension*), crie uma função que leia de um ficheiro linha a linha e uma outra função que escreva noutro, mas filtrando os '#'. Nota: o método `writelines` aceita iteráveis.

4. Considere a seguinte classe incompleta. Esta classe já é iterável porque implementa `__iter__` que retorna um iterador. O objeto iterador é `self`, ou seja, o próprio objeto. Complete a classe de forma a implementar o iterador `__next__()` que retorne iterativamente as potências de 2 até um máximo de `self.max` elementos.

Nota: o `raise StopIteration` termina a sequência de iteração.

```
class PowTwo:
    """Classe para implementar um iterador
    de potências de 2"""

    def __init__(self, max = 0):
        self.max = max

    def __iter__(self):
        self.n = 0
        return self
```

Teste com:

```
print ([ l for l in PowTwo(10) ])
```

que dever retornar: [1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024]

5. Na sucessão de Fibonacci o elemento seguinte resulta da soma dos dois anteriores. Para $F_0=0$ e $F_1=1$, o termo $F_n = F_{n-1} + F_{n-2}$.

Desenvolva uma classe que permita a iteração dos N primeiros termos da sucessão de Fibonacci.

6. Implemente uma função geradora que devolva infinitamente a sucessão de valores de Fibonacci.

6. Desenvolva uma função geradora ou expressão geradora onde que dada uma entrada iterável, devolve (filtra) os valores que são ímpares.

7. Finalize o guião e envie via moodle.