



Lista de Exercícios 01

```
1  '''
2  Website list
3  '''
4  import logging
5  import requests
6
7
8  SITE_LIST = ['https://envato.com', 'http://amazon.co.uk',
9              'http://amazon.com', 'http://facebook.com', 'http://google.com',
10             'http://google.fr', 'http://google.es', 'http://google.co.uk',
11             'http://internet.org', 'http://gmail.com',
12             'http://stackoverflow.com', 'http://github.com',
13             'http://heroku.com', 'http://really-cool-available-domain.com',
14             'http://djangoproject.com', 'http://rubyonrails.org',
15             'http://basecamp.com', 'http://trello.com',
16             'http://yiiframework.com', 'http://shopify.com',
17             'http://another-really-interesting-domain.co',
18             'http://airbnb.com', 'http://instagram.com',
19             'http://snapchat.com', 'http://youtube.com', 'http://baidu.com',
20             'http://yahoo.com', 'http://live.com', 'http://linkedin.com',
21             'http://yandex.ru', 'http://netflix.com', 'http://wordpress.com',
22             'http://bing.com'
23         ]
24
25
26 def check_website(address, timeout=20):
27     '''
28     Check if a website is down. A website is considered down
29     if either the status_code >= 400 or if the timeout expires
30
31     Throw a WebsiteDownException if any of the website down conditions are met
32     '''
33     try:
34         response = requests.head(address, timeout=timeout)
35         if response.status_code >= 400:
36             logging.warning('Website %s returned status_code=%s', address,
37                             response.status_code)
38     except requests.exceptions.RequestException:
39         logging.warning('Timeout expired for website %s', address)
```

Figura 1: Módulo `websites.py`

```

1  '''
2  Sequential approach
3  '''
4
5  import time
6  from websites import SITE_LIST, check_website
7
8
9  def main():
10     '''
11     Main function
12     '''
13     start_time = time.time()
14     for address in SITE_LIST:
15         check_website(address)
16     end_time = time.time()
17     print('Time for sequential:', end_time - start_time, 'secs')
18
19
20 if __name__ == '__main__':
21     main()

```

Figura 2: Módulo `sequential.py`

### Exercício 1:

Observe os códigos das Figuras 1 e 2. O módulo `websites.py` possui uma lista de sites e uma função para verificar se um site está online ou offline. O módulo `sequential.py` percorre a lista de sites e faz a verificação de cada um. Implemente essa mesma verificação usando *threads* e processos. Dicas:

1. Utilize as estruturas **Queue** (para *threads* e **JoinableQueue** (para processos) para sincronizar processamento paralelo;
2. Tais estruturas possuem seguintes métodos:
  - put(item):** insere o item na fila (solicita e bloqueia);
  - get():** pega o primeiro item da fila (solicita e bloqueia);
  - task\_done():** libera a fila do bloqueio;
  - empty():** Retorna **True** se a fila estiver vazia;
  - join():** Espera a finalização de todas as *threads* e processos que estejam usando a fila.