

# Извлечение данных из веб-ресурсов

## Получение информации с веб-страницы с адресом url

```
In import requests

req = requests.get(URL)
print(req.text)      # вывод содержимого веб-страницы
print(req.status_code) # вывод кода возврата
```

## Поиск первого вхождения подстроки, соответствующей регулярному выражению pattern, в строку string

```
In import re
print(re.search(pattern, string).group())
```

## Разделение строки string на подстроки, границы которых определяются регулярным выражением pattern

```
In import re
print(re.split(pattern, string, maxsplit=num_split))

# maxsplit – максимальное число делений, по умолчанию maxsplit = 0
```

## Поиск подстроки по шаблону pattern в строке string и замена её на подстроку repl

```
In import re
print(re.sub(pattern, repl, string))
```

## Поиск всех подстрок по шаблону pattern в строке string

```
In import re
print(re.findall(pattern, string))
```

## Формирование древовидной структуры веб-страницы

```
In from bs4 import BeautifulSoup
soup = BeautifulSoup(req.text, parser)
```

## Поиск первого тега tag

```
In # Возвращает строку с тегом, атрибутами и содержимым
# attrs – словарь атрибутов тега

tag_content = soup.find(tag, attrs={"attr_name": "attr_value"})
print(tag_content.text) # контент без тега
```

## Выполнение операций со всеми тегами tag

```
In # attrs – словарь атрибутов тега

for tag_content in soup.find_all(tag, attrs={"attr_name": "attr_value"}):
    # do something
```

# Словарь

**Web mining**

процесс поиска веб-ресурсов, необходимых для исследования, и извлечения информации из них

**HTML**

язык разметки гипертекста, который используется для создания веб-страниц

**HTTP**

транспортный протокол передачи данных в сети интернет

**HTTPS**

защищённая версия протокола HTTP

**HTML-тег**

элемент языка разметки гипертекста

**Атрибут тега**

свойство, дающее дополнительные возможности при работе с тегами