

# Поиск тегов: методы bs4 и рекомендации

## Метод `find_all()`

Метод `find_all()` всегда возвращает список.

### Поиск по названию тега

Используется параметр `name`. Название параметра можно не указывать:

```
# Название тега передаётся строкой.
result = soup.find_all('b')

# Названия тегов передаются списком.
result = soup.find_all(['title', 'b'])
```

### Поиск по атрибуту

Используется параметр `attrs`:

```
# Поиск в коде всех элементов с классом 'title'.
result = soup.find_all(attrs={'class': 'title'})
```

Любой неизвестный ключевой аргумент библиотека *bs4* воспринимает как атрибут тега:

```
# Поиск в коде всех элементов id='link2'.
result = soup.find_all(id='link2')
```

При поиске по классу аргумент прописывается как `class_`, с нижним подчёркиванием:

```
# Поиск в коде всех элементов class='antihero'.
result = soup.find_all(class_='antihero')
```

## Комбинированный поиск — по названию и атрибуту

```
# Поиск в коде всех элементов <a class='hero'>.
result = soup.find_all('a', attrs={'class': 'hero'})
```

## Метод find()

Метод `find()` возвращает только первый найденный элемент.

```
# Поиск в коде первого элемента <a class="hero">.
result = soup.find('a', attrs={'class': 'hero'})
```

## Поиск внутри найденного тега

```
# HTML-код для поиска:
# <p class="story">
# <a class="hero" href="http://example.com/ilya" id="link1">Илья Муромец</a>,
# <a class="hero" href="http://example.com/alesha" id="link2">Алёша Попович</a>,
# </p>

# Метод find_all.
all_stories = soup.find_all('p', class_='story')
first_story = all_stories[0]
link2 = first_story.find_all(id='link2')
# Найдёт:
# [<a class="hero" href="http://example.com/alesha" id="link2">Алёша Попович</a>]

# Метод find.
first_story = soup.find('p', class_='story')
link2 = first_story.find(id='link2')
# Найдёт:
# <a class="hero" href="http://example.com/alesha" id="link2">Алёша Попович</a>
```

## Как получить содержимое тегов

Текст — при помощи атрибута `text`:

```
# HTML-код для поиска:
# <a class="hero" href="http://example.com/alesha" id="link2">Алёша Попович</a>
link2 = soup.find(id='link2')
print(link2.text)
```

```
# Будет напечатано:  
# Алёша Попович
```

К любому содержимому атрибута тега можно обратиться как к ключу словаря:

```
# HTML-код для поиска:  
# <a class="hero" href="http://example.com/alesha" id="link2">Алёша Попович</a>  
link2 = soup.find(id='link2')  
print(link2['href'])  
  
# Будет напечатано:  
# http://example.com/alesha
```

## Рекомендации по поиску тегов

- Ищите теги в несколько шагов — от более крупных к более мелким.
- Не полагайтесь на относительную вложенность элементов и их расположение по порядку. Ищите нужный элемент, опираясь на теги и их атрибуты.
- При поиске по атрибутам в первую очередь используйте атрибут `id`, так как он должен быть уникальным на странице (но бывают исключения).
- Если значение атрибута состоит из нескольких слов, обращайтесь к одному из них. Выбирайте из слов то, которое, как вам кажется, изменится с меньшей вероятностью.

```
# HTML-код для поиска:  
# <a href="http://example.com/ilya" class="hero superhero" id="link1">Илья</a>,  
# <a href="http://example.com/alesha" class="superhero hero" id="link2">Алёша</a>  
  
super_hero = soup.find_all(class_='superhero')  
  
# Найдёт оба тега <a> и их содержимое.
```

- Не используйте классы, которые связаны со стилями, или которые обозначают что-то непонятное. Например, `class="col-md-3 mb-3 mt-1"` плохо подходит для поиска.