

# Тестовое задание для Python-разработчика

Задача 1: Парсинг данных с Avito   
Требования:   
1. Написать скрипт на Python для парсинга объявлений о недвижимости в ЮФО с Avito.   
2. Данные должны сохраняться в XML-файлы, разбитые по 2000 объявлений в каждом.   
3. Поля для парсинга:   
 - Заголовок объявления   
 - Цена   
 - Адрес   
 - Площадь (м²)   
 - Ссылка на объявление   
 - Дата публикации   
4. Обеспечить обработку ошибок (например, недоступность страницы).   
Что оцениваем:  
- Использование библиотек (BeautifulSoup, Scrapy, requests и т.д.).   
- Оптимизацию работы с большими объемами данных.   
- Чистоту кода и документацию.

Задача 2: Интеграция XML с веб-интерфейсом

\*\*Требования:\*\*   
1. Написать скрипт для загрузки XML-файлов через веб-интерфейс (эмулировать POST-запрос).   
2. Реализовать проверку на дубликаты объявлений (по ссылке или заголовку + адресу).   
3. Добавить простой HTML-интерфейс с кнопкой для загрузки XML и отображением статуса (успех/ошибка).   
Пример структуры HTML:   
```html  
<form action="/upload" method="post" enctype="multipart/form-data">  
 <input type="file" name="xml\_file">  
 <button type="submit">Загрузить</button>  
</form>  
<div id="status"></div>  
```  
Что оцениваем:   
- Умение работать с HTML/CSS/JavaScript.   
- Понимание HTTP-запросов и интеграции с бэкендом.   
- Решение для проверки уникальности данных.   
  
Задача 3: Работа с Git   
1. Создать репозиторий на GitHub/GitLab, разместить решение.   
2. Имитировать процесс code review:   
 - Создать ветку feature/avito-parser.   
 - Сделать 3+ коммита с осмысленными сообщениями.   
 - Оформить пулл-реквест в main, описать изменения.   
3. В README.md кратко объяснить:   
 - Как запустить проект.   
 - Как вы проверили уникальность объявлений.   
 - Какие проблемы могли возникнуть при парсинге Avito и как их решить.   
  
Что оцениваем:   
- Следование git-flow.   
- Умение документировать код и процессы.   
  
Задача 4: Легаси-код   
Вам предоставлен PHP-код (условный пример):   
```php  
function loadXML($file) {  
 $xml = simplexml\_load\_file($file);  
 foreach ($xml->ad as $ad) {  
 // ... сохранение в базу без проверки дубликатов  
 }  
}  
```  
Вопросы:   
1. Какие проблемы вы видите в этом коде?   
2. Как бы вы модифицировали его для работы с Python-скриптом?   
3. Предложите решение для увеличения производительности при загрузке 10k+ объявлений.