



# Ира

👤 Owner	
🔍 Verification	
🏷️ Tags	
🕒 Last edited time	@May 8, 2023 11:33 PM

## AnilistPython

*Библиотека про аниме и мангу.*

Основная [документация](#) на PyPI. [Код](#) и его описание на GitHub.

Советую также ознакомиться с [JSON-файлами](#), которые есть в открытом доступе на GitHub.



Проект состоит из **двух письменных частей** и **одной устной части**:

- 📄 Оформленная тетрадка с описанием библиотеки;
- 📄 Тетрадка с решением индивидуальных задач и предложенным заданием по теме (или заданиями);
- 🗣️ Презентация описания своей библиотеки перед остальными.

За каждую из частей ставится **отдельная оценка**. Также можно получить **дополнительные баллы**, которые могут улучшить оценку / принести дополнительную пятерку (в зависимости от их количества).

### Дедлайны:

к **10:00 15.05 (пн)** каждый должен закончить оформлять тетрадку с описанием и быть готовым выступить с презентацией

к **12:00 19.05 (пт)** у меня в гугл форме должны оказаться все тетради с решением индивидуальных заданий

### Регламент презентаций:

Никакой дополнительной презентации **не нужно**, только готовая тетрадка с описанием, по которой вы должны быть готовы рассказать про свою тему.

**Время:** 10 минут на презентацию + 5 минут на вопросы

Таким образом за пн и пт все докладываются!

## Общая задача:

1. Ознакомиться с документацией твоей библиотеки / твоих библиотек.
2. Создать тетрадку в Google Colab. В начале написать несколько общих слов: зачем нужна твоя библиотека и что вообще она умеет делать?  
**NB!** Не копируй мое описание)
3. В этой же тетрадке описать функции, которые способна выполнять твоя библиотека (если не сказано иного, то **все**). Вот советы, которые помогут:
  - а. Дели на смысловые блоки (если у тебя несколько разных библиотек, не смешивай их; если одна, то придумай в каком порядке рассказывать и как делить на части).
  - б. Опиши синтаксис каждой функции из тетрадки: твоя функция принимает какие-то аргументы внутрь? если да, то какие, какого типа и сколько? как вызвать эту функцию? возвращает ли она какой-то тип данных? если да, то какой?
  - в. Опиши смысл каждой функции из тетрадки (по-русски!): что она делает? зачем она нужна вообще?  
**NB!** Что не стоит делать, так это переводить дословно англоязычную документацию... Постарайся своими словами, потому что, если я поймаю кого-то на бездумном переводе, кара будет мгновенная 😊

- d. Для каждой функции приведи пример ее выполнения.  
**NB!** Как и в предыдущем пункте, старайся придумывать примеры отличные от тех, что в документации...
  - e. Если ты не можешь что-то понять ни из документации, ни из примеров, гугли!
4. Как оформлять тетрадку?
- a. Вот тут мы обсуждали оформление тетрадок в Google Colab.
  - b. Вот тут есть подробный гайд по *markdown*.
  - c. Для примера можешь взять мои тетрадки: эту, эту или, например, эту.

**Важно!** Ответы на дальнейшие вопросы тоже **обязательно** должны оказаться в твоей тетрадке, какими бы простыми и болтологическими они ни казались. Сразу предупреждаю, что если я их там не увижу, то буду спрашивать устно и придется выдумывать на ходу.

5. Представь себе, что ты создатель\_ница этой библиотеки / этих библиотек и хочешь выбить грант за твою работу, представляя перед потенциальными спонсорами.
- Как тебе кажется, кому и чем эта библиотека может быть полезна / интересна? Если у тебя несколько библиотек расскажи про **уникальные** плюсы каждой (не надо этих общих слов про “ну, она прикольная...”).
- (Можешь получить допбаллы, если проведешь ресерч о том, какие есть аналогичные библиотеки, если таковые существуют, и расскажешь, чем твоя / твои лучше. В этом тебе может помочь гугл (лучше искать инфу по-английски) и [GitHub Topics!](#))
6. Представь себе, что ты критик\_есса и хочешь указать на все слабые места библиотеки / библиотек.
- Чего не хватает в функционале (что бы ты добавил\_а, если мог\_ла)? Что может быть неудобно пользователю? Какие проблемы / ошибки тебе встретились, если таковые были?
7. Приложи ссылки на документацию / другие ресурсы, которые были полезны.

### Индивидуальные задания:

1. Представь, что ты учительница в школе / университете, где все школьники / студенты обязаны показать свое знание этой библиотеки. Придумай задание, в котором требуется использовать твою библиотеку, подробно опиши его условия и напиши “каноническое” решение.  
 (Можно получить допбаллы за придумку большего количества заданий!)
2. Дан **CSV-файл** с цитатами из аниме, где есть сама фраза; имя героя, который ее произнес, и из какого она аниме. Напиши функцию `get_character_quote(name)`, которая получала бы на вход имя персонажа и печатала (именно `print` — не `return`): его изображение, полное имя (на латинице + японское в скобках), из какого он аниме (по-английски + на ромадзи в скобках) и цитату / цитаты (если есть: пронумерованные, каждая с новой строки).  
 Например, так должен выглядеть вывод для `'Himura Kenshin'`:



```
Name: Kenshin Himura (緋村剣心)
Anime: Rurouni Kenshin (Rurouni Kenshin: Meiji Kenkaku Romantan)
Quote(s):
1. Whatever you lose, you'll find it again. But what you throw away you'll never get back.
2. You can die anytime, but living takes true courage.
3. You'll only realize that you truly love someone if they already caused you enormous pain. Your enemies can never hurt you the way
```

**P.S.** Мы уже однажды выводили картинки по URL, но [здесь](#) есть еще разные варианты кода. Только помни, что обычно картинки мы “печатаем” при помощи `img.show()`, где вместо `img` должно стоять название переменной, в которую мы записали открытую картинку.

3. У этой библиотеки открытая база данных, поэтому мы можем немного “докрутить” поиск. Тут хранится JSON-файл, в котором записан словарь, где каждому жанру соответствует список ID. Напиши функцию `search_by_genres(*genres)`, которая принимает на вход сколько-угодно жанров и ищет по файлу ID подходящих аниме на их пересечении. Например, если была вызвана `search_by_genres('psychological', 'romance')`, то функция должна найти ID всех психологическо-романтических аниме. Дальше отсортируй полученные ID по средней оценке ( `'average_score'` ) от самой высокой до самой низкой и верни пользователю **английское название** аниме с самой высокой оценкой.

```
>>> search_by_genres('psychological', 'romance', 'supernatural')
<<< Fruits Basket The Final Season

>>> search_by_genres('sports', 'supernatural')
<<< Hikaru no Go
```

**NB!** Для решения этого задания тебе скорее всего понадобится конструкция с исключениями `try-except`. Про них можно почитать [тут](#).