## Как сдавать ревью

Дедлайн на первую посылку 12 мая 01:00. Первая посылка обязательно должна работать, иначе ревью не принимается! Заметьте, не “не падать”, а именно работать. pep8 - **обязательный** код стайл, приведенная команда не должна ничего вывести в вашей директории:

pycodestyle . --max-line-length=120 -r

В первой посылке обязательна только базовая функциональность, бонусы можно досылать в последующих итерациях. Только учтите, что они тоже подлежат ревью. Если они не успеют его успешно пройти, они не засчитаны. А в конце семестра, сами понимаете, число сдающих может вырасти вместе со временем ожидания комментариев от проверяющего) Так что, чем раньше вы сделаете бонусы, тем лучше.

Сам процесс сдачи устроен так:

* Придумываем задачу, подходящую под требования ниже
* Согласовываем её с семинаристом (@Onaga1958 в телеграмме, если что :) )
* Создаем пустой **приватный** репозиторий на github
* Добавляем туда Onaga1958 и ivanov-v-v: Settings->Manage Access->Invite a collaborator
* Коммитим в ветку **master** README.md с небольшим описанием
* Создаем ветку **dev** от **master**
* Пишем код в ветке **dev** и пушим в репозиторий
* Создаем pull request из **dev** в **master** и **не вливаем** его. В нем будет проходить обсуждение задачи.
* И дальше пока семинарист не принял ревью повторяем следующие шаги:
  + Ждем комментариев
  + Исправляем код, коммитим в **dev**, пушим. На исправление замечаний дается **пять дней**.
* Успеть пройти ревью **лучше** до начала зачетной сессии, скорость проверки на зачетной недели будет сильно ниже. **Нужно** успеть пройти ревью за день до окончания зачетной недели.

## Требования к задаче

* Веб-сервер. Лучше на flask, но можно и django, если очень хочется.
* Тесты (unittest или pytest)
* В README должна быть команда для запуска сервера и пример, объясняющий как пользоваться интерфейсом
* Она не должна быть слишком простой (веб-сервер для сложения целых чисел не подойдет)
* Но и не должна быть слишком сложной (писать свой vk прикольно, но за две недели вы не успеете)
* **Можно** использовать свой первый проект, но, конечно, не обязательно

## Примеры подходящих проектов (их выбирать нельзя)

* Пример относительно сложного проекта: псевдо юпитер ноутбук, который мы смотрели на семинаре. Там разве что тестов не хватает.
* Пример относительно простого проекта: сайт с курсами валют. В HTML помещаем формы, где можно выбрать валюту покупки, валюту продажи и дату. На сервере делаем HTTP запрос в API ЦБ.

## Бонусы

Тут четкого критерия нет, нужно к веб-серверу прикрутить что-нибудь прикольное. Примеры того, за что можно получить бонус:

* использовать базу данных
* реализовать какой-нибудь нетривиальный алгоритм на сервере (dfs считаем тривиальным)
* сделать классный, красивый html
* использовать matplotlib или plotly ([документация про интеграцию с flask](https://dash.plotly.com/integrating-dash))
* web scraping (см. 9-й семинар)
* pygame

Заметьте, сам факт использования например базы данных “просто чтобы было” не оценивается, нужно чтобы она действительно была нужна и использовалась по назначению. Если сомневаетесь достаточно ли прикольная ваша фича для бонуса, об этом всегда можно спросить семинариста)