В этом уроке мы с вами порешаем задачки и поотвечаем на вопросы про API и дополнительные функции и методы, которые вы успели изучить в лекциях

⭐️Задание со звёздочкой!⭐️

Считайте данные из Google Sheets, которые находятся по этой [ссылке](https://docs.google.com/spreadsheets/d/1e3PRXmMNXo9AZZnaol1aHgYxtH6KYaCDymEQlK6jaO8/export?format=csv) при помощи библиотеки requests. Посчитайте процент мужчин среди всех посетителей сайта, округлите ответ до целых.

Воспользуйтесь этим кодом, чтобы перевести данные из реквеста в пандас

from io import BytesIO

import requests

req = requests.get(your\_link)

data = req.content

df = pd.read\_csv(BytesIO(data))

[Подсказки](https://www.notion.so/7-1-PYTHON-ef010af43d6f4d8392c1f85985570a38)

Скачайте файл с данными посещаемости сайта Яндекс.Метрика [отсюда](https://yadi.sk/d/NU25PVVkxIEEtw) (сохранен в переменную df). При помощи функции np.where создайте серию, в которую будет записываться 1, если посещения были сделаны роботами с браузера, содержащего Chrome в названии, и 0 в других случаях. Поместите получившуюся серию в переменную chrome\_robots.

Может понадобиться:

* [str.contains()](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.Series.str.contains.html)

[Подсказки](https://www.notion.so/7-2-PYTHON-5efee62fe86c4fbe8657bdbec0f8d568)

⭐️Задание со звёздочкой!⭐️

У вас есть [датасет](https://stepik.org/media/attachments/lesson/361619/sample_ads.csv) с данными о кликах и показах по объявлению, в котором есть пропуски. Вычислите количество строк без пропусков в колонке click\_type. Запишите ответ в переменную not\_nan\_count

* ad\_id – id объявления
* user\_id – id пользователя
* click\_type – тип клика (клик по заголовку объявления, клик по центру)

Данные сохранены в переменную df.

[Подсказки](https://www.notion.so/7-3-PYTHON-9612affae3374fc3837934c668fe1f4f)

⭐️Задание со звёздочкой!⭐️

В списке lst находятся различные элементы. Создайте список new\_list и сделайте следующее:

* при помощи конструкции try except и цикла попробуйте преобразовать каждое значение в листе в int
* если это получилось – добавьте преобразованное значение  в список new\_list

[Подсказки](https://www.notion.so/7-4-PYTHON-49f2687dcc8b4983bbf98a8526eb91e4)

⭐️Задание со звёздочкой! ⭐️

А теперь попробуем воспользоваться знаниями использования Telegram API. Отправьте себе в личку любое послание, а в ответ к этому шагу напишите ник бота, от имени которого будете отправлять сообщения.

[Подсказки](https://www.notion.so/7-5-PYTHON-cc7d1ca8f59242e88586ea41106d2fe5)

Настало время последнего мини-проекта в блоке Python! В нём вы поработаете с API и узнаете, как ведут себя пользователи на сайте Яндекс.Метрики.

Так как мы работаем с демо данными Яндекс.Метрики, нам позволено видеть данные только за последнюю неделю. Это требует обновления заданий и ответов на них каждую неделю, что не очень удобно и вам, и нам. Поэтому для удобства проверки мы решили немного модифицировать степы: теперь для того, чтобы ответить на вопросы, вам необходимо поработать с уже выгруженными датасетами из Яндекс.Метрики. Но вы можете самостоятельно повторить каждый из запросов в задании к реальному API Яндекс.Метрики, чтобы потренироваться и получить данные в таком же формате, как в прикрепленном к задаче датасете.

Удачи!

Для начала, давайте посмотрим на распределение посещений сайта по дням. Постройте график с распределением количества посещений от ботов и обычных пользователей сайта Яндекс.Метрика. Данные для построения графика хранятся [здесь](https://stepik.org/media/attachments/lesson/361618/step_1_lesson_7.csv).

[Подсказки](https://www.notion.so/7-1-PYTHON-82cda571e1ba4c8fa60a1e42a3cb7f5b)

Наименьшее число посещений пользователей-людей наблюдается 15 июня

Наибольшая разница между числом посещений людьми и ботами наблюдается 16 июня

15 июня сайт посетило наибольшее число ботов

15 июня ботов было примерно в два раза больше, чем 16 июня

15 июня сайт посетило наименьшее число ботов

Наименьшее число посещений наблюдается 17 июня только среди пользователей

17 июня сайт посетило наибольшее число ботов, по сравнению с 15 и 16 июня

17 июня ботов было примерно в два раза меньше, чем 15 июня

Определите процент посещений сайта Яндекс.Метрики с бразузера Google Chrome. Данные лежат [здесь](https://stepik.org/media/attachments/lesson/361618/step_2_lesson_7.csv).

Ответ округлите до целого.

[Подсказки](https://www.notion.so/7-2-PYTHON-94fe9f24df5c4c51a7c53e787f0af1ac)

Давайте выясним, с какого браузера приходит **больше** всего ботов. Посмотрите на [данные](https://stepik.org/media/attachments/lesson/361618/step_3_lesson_7.csv) и определите, у какого браузера самая **большая** **доля** посещений сайта ботами от общего числа посещений в этом браузере?

В качестве ответа укажите название браузера.

[Подсказки](https://www.notion.so/7-3-PYTHON-46900648f56d4f26b1a3f9caac6ce304)

Выгрузите [данные](https://stepik.org/media/attachments/lesson/361618/step_4_lesson_7.csv), содержащие информацию о дне, типе пользователя, браузере. Создайте колонку с метрикой доли заходов на сайт для каждого из этих разрезов.

В ответе укажите процент открытий сайта настоящими пользователями Safari (не ботами) с точностью до 1 знака после точки.

Hint: обратите внимание, здесь вам нужно найти долю заходов конкретных типов пользователей с конкретного браузера от вообще всех-всех заходов на сайт за этот день.

[Подсказки](https://www.notion.so/7-4-PYTHON-e1990eb2888646039f8de8d404ecd16a)

💀 Сложное задание! 💀

При помощи библиотеки gspread отправьте данные из предыдущего шага в Google SpreadSheet. Далее отправьте ссылку на получившийся файл в Google SpreadSheet (не забудьте открыть доступ по ссылке, это можно сделать вручную, когда данные уже зальются в таблицу) при помощи VK API себе в личку ВКонтакте.

В ответе напишите слово *Да*, если всё сделали :)

[Подсказки](https://www.notion.so/7-5-PYTHON-96d37f60654f4fb4a5535a79e457a0bb)

⭐️Задание со звёздочкой!⭐️

Другой аналитик уже агрегировал часть данных из метрики. Вам нужно объединить усилия, чтобы получить финальный отчет. Прочитайте дополнительные данные из Google SpreadSheet, объедините вашу таблицу из предыдущего шага с таблицей аналитика и посчитайте количество визитов на человека (отношение visits к users) в разрезе по браузеру и значениям Robots/People.

В ответе укажите, сколько визитов на уникального пользователя совершается ботами с браузера Google Chrome (округлите до целого)

Вот так можно прочитать данные другого аналитика:

from io import BytesIO

import requests

r = requests.get('https://docs.google.com/spreadsheets/d/e/2PACX-1vQWMsvBTVio9C7IOOxfFO9C15BRHyME-\_ENHqBodDOjuiHwk9fCuF5hUVmDs497PZOqPYK3exdSikOK/pub?gid=1006633900&single=true&output=csv')

data = r.content

df = pd.read\_csv(BytesIO(data))

[Подсказки](https://www.notion.so/7-6-PYTHON-5be1420805864cf7a85cd4941650612c)

В рамках дополнительного проекта попробуем освоить API, позволяющий получать данные о музейных коллекциях. Это задание может пригодиться для большего понимания того, какими способами можно собирать и обрабатывать данные, например, для собственного проекта или другой рабочей задачи. Также вы можете более глубоко изучить документацию, чтобы потом исследовать другие интересные для себя вопросы по полученным данным :)

Описание API

**Документация:** [https://metmuseum.github.io](https://metmuseum.github.io/)

**Ключ:**для использования выбранного API (на момент написания задания) ключ не требуется :)

Всего есть 4 основных раздела:

* **Объекты** (Objects)
* **Объект** (Object) – данные об объекте (например, о картине)
* **Департаменты** (Departments) –  данные о департаменте/разделе (например, "Современное искусство")
* **Поиск** (Search) – используется для поиска

В рамках задач поработаем с блоками Object и Search.

С более подробным описанием каждого доступного параметра можно ознакомиться в [документации](https://metmuseum.github.io/).

Для начала, импортируйте нужные библиотеки:

import requests

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

Сам запрос к API должен состоять из нескольких параметров. Пример обращения к поиску (Search):

r = requests.get('https://collectionapi.metmuseum.org/public/collection/v1/search?q=QUERY')

res = r.json()

где QUERY – ключевое слово.

Предположим, ваша задача – найти все произведения, связанные с подсолнухами. Выполните поиск по слову *"sunflowers".* Внимательно посмотрите на формат ответа и сохраните **id** полученных объектов в список sunflower\_ids.

В качестве ответа выберите id объектов (3), которые присутствуют в полученном списке:

[Подсказки](https://abrupt-moose-9e7.notion.site/7-1-PYTHON-f59ce56a23234e89bae98a2ab5d62d5d)

20149

437329

2032

16822570

436524

Отлично! На предыдущем шаге мы получили результаты поиска. Следущий этап – собрать информацию об этих объектах.

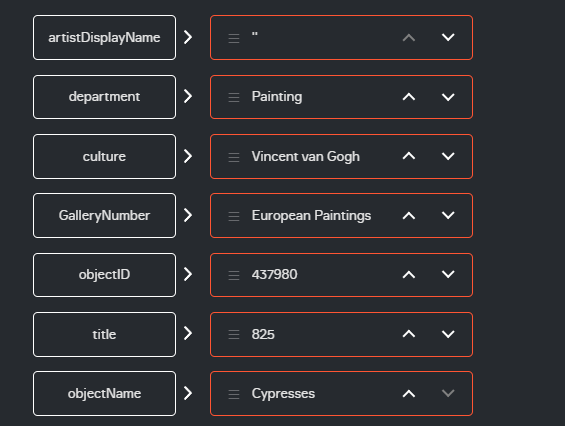
Пример текста запроса для получения информации об одном объекте (Object):

r\_test = requests.get(f'https://collectionapi.metmuseum.org/public/collection/v1/objects/OBJECT\_ID')

obj\_test = r\_test.json()

где OBJECT\_ID – id объекта.

Выполните запрос для объекта 437980. Для проверки соотнесите полученные параметры с их значениями:



Теперь напишите цикл, с помощью которого будет собрана информация об объектах, которые были получены на 2 шаге. Результаты запишите в датафрейм, а именно – следующие параметры:

1. **objectID** – id объекта
2. **title** – название
3. **artistDisplayName** – автор
4. **department** – департамент
5. **objectBeginDate** – дата (начало)
6. **objectEndDate** – дата (конец)
7. **period** – название периода
8. **objectName** – название/категория объекта
9. **culture** – культура

Сколько объектов относятся к японской культуре? (culture –Japan)

[Подсказки](https://abrupt-moose-9e7.notion.site/7-3-PYTHON-a763632d1a674015ae11a7076796ceae)

1

10

5

22

45

18

Теперь попробуем построить более сложный запрос с помощью секции "Search"! На этот раз параметры необходимо передать через "&", например, departmentId (id департамента – 1) и q (ключевое слово – cat):

r = requests.get('https://collectionapi.metmuseum.org/public/collection/v1/search?departmentId=1&q=cat')

cats = r.json()

Выполните поисковый запрос для департамента "Asian art" (departmentId = **6**) с тегом "cat" и запишите id объектов в cat\_ids. Используйте цикл из шага 4, чтобы получить данные о первой **1000** объектов из списка.

**Note:** выполнение цикла может занять некоторое время (~5 минут), поэтому **проверять** его работоспособность лучше **на** более **маленькой** **выборке**!

В качестве ответа укажите, в каком периоде (period) было выполнено наибольшее количество работ, представленных в музее?

[Подсказки](https://abrupt-moose-9e7.notion.site/7-4-PYTHON-b637e85407b0466dbd4415820b0cd639)

Timurid period (1370–1507)

Edo period (1615–1868)

Tang dynasty (618–907)

Meiji period (1868–1912)

Период чаще всего не указан