**bat-файл, котрый коммитит и пушит все изменения в текущей директории**

Вот.bat-файл, который настраивает кодировку UTF-8, запускает скрипт, коммитит и пушит все изменения в текущей директории, включая новые и изменённые файлы:

@echo off

chcp 65001 > nul

echo ============================================

echo 🚀 Запуск скрипта обновления таблицы...

python generate\_colab\_table.py

if %errorlevel% neq 0 (

echo ❌ Ошибка при запуске скрипта. Проверь generate\_colab\_table.py

pause

exit /b

)

echo ============================================

echo 📚 Добавление файлов в Git...

git add .

echo ============================================

echo 📝 Создание коммита...

git commit -m "Автообновление: README.md и другие изменения"

echo ============================================

echo ☁️ Отправка изменений на GitHub...

git push origin master

echo ============================================

echo ✅ Всё готово! Все изменения загружены на GitHub.

pause

🔧 **Что делает этот скрипт:**

* git add . — добавляет **все изменённые и новые файлы** в текущей папке;
* git push origin master — пушит именно в ветку master;
* Работает из той же директории, что и .bat-файл и твои проектные файлы.

**ругается на отсутствие generate\_colab\_table.py**

**✅ Обновлённый generate\_colab\_table.py для GitPyProba:**

import os

from urllib.parse import quote

# Настройки

GITHUB\_USER = "jkrndkot"

GITHUB\_REPO = "GitPyProba"

CHEATSHEETS\_DIR = "cheatsheets"

# Получаем список всех .ipynb файлов

notebooks = [f for f in os.listdir(CHEATSHEETS\_DIR) if f.endswith(".ipynb")]

notebooks.sort()

def filename\_to\_title(filename):

# Убираем расширение и заменяем подчеркивания на пробелы

name = filename.replace(".ipynb", "")

name = name.replace("\_", " ")

return name

# Генерируем таблицу

lines = []

lines.append("| Тема | Открыть в Colab |")

lines.append("|------|------------------|")

for nb in notebooks:

title = filename\_to\_title(nb)

local\_link = f"{CHEATSHEETS\_DIR}/{quote(nb)}"

colab\_url = f"https://colab.research.google.com/github/{GITHUB\_USER}/{GITHUB\_REPO}/blob/master/{CHEATSHEETS\_DIR}/{quote(nb)}"

colab\_badge = f"[![Colab](https://colab.research.google.com/assets/colab-badge.svg)]({colab\_url})"

lines.append(f"| [{title}]({local\_link}) | {colab\_badge} |")

# Автоматическая замена таблицы в README.md между маркерами

README\_PATH = "README.md"

START\_MARK = "<!-- START CHEATSHEETS TABLE -->"

END\_MARK = "<!-- END CHEATSHEETS TABLE -->"

with open(README\_PATH, encoding="utf-8") as f:

readme = f.read()

if START\_MARK in readme and END\_MARK in readme:

before = readme.split(START\_MARK)[0]

after = readme.split(END\_MARK)[1]

table\_md = "\n".join(lines)

new\_readme = before + START\_MARK + "\n" + table\_md + "\n" + END\_MARK + after

with open(README\_PATH, "w", encoding="utf-8") as f:

f.write(new\_readme)

print("Таблица успешно обновлена в README.md между маркерами.")

else:

print("Маркеры таблицы не найдены в README.md. Изменения не внесены.")

**Изменения:**

* GITHUB\_USER = "jkrndkot" — твой пользователь;
* GITHUB\_REPO = "GitPyProba" — актуальный репозиторий;
* blob/master/... — корректная ветка (у тебя master, не main).

💡 Убедись, что:

* Папка cheatsheets и файл README.md существуют;
* В README.md есть маркеры:

<!-- START CHEATSHEETS TABLE -->

...

<!-- END CHEATSHEETS TABLE -->

**а у меня нет README.md файла**

Отлично, тогда я предложу тебе **адаптированную версию скрипта**, которая:

1. Если README.md **есть и содержит маркеры** — обновляет таблицу между ними;
2. Если README.md **есть, но без маркеров** — добавляет таблицу в конец файла;
3. Если README.md **вообще нет** — **создаёт новый** README.md с таблицей.

**✅ generate\_colab\_table.py, учитывающий отсутствие README.md: принято за основу!**

python

КопироватьРедактировать

import os

from urllib.parse import quote

# Настройки

GITHUB\_USER = "jkrndkot"

GITHUB\_REPO = "GitPyProba"

CHEATSHEETS\_DIR = "cheatsheets"

# Получаем список всех .ipynb файлов

if not os.path.isdir(CHEATSHEETS\_DIR):

print(f"❌ Папка '{CHEATSHEETS\_DIR}' не найдена.")

exit(1)

notebooks = [f for f in os.listdir(CHEATSHEETS\_DIR) if f.endswith(".ipynb")]

notebooks.sort()

def filename\_to\_title(filename):

name = filename.replace(".ipynb", "")

name = name.replace("\_", " ")

return name

# Генерируем таблицу

lines = []

lines.append("<!-- START CHEATSHEETS TABLE -->")

lines.append("| Тема | Открыть в Colab |")

lines.append("|------|------------------|")

for nb in notebooks:

title = filename\_to\_title(nb)

local\_link = f"{CHEATSHEETS\_DIR}/{quote(nb)}"

colab\_url = f"https://colab.research.google.com/github/{GITHUB\_USER}/{GITHUB\_REPO}/blob/master/{CHEATSHEETS\_DIR}/{quote(nb)}"

colab\_badge = f"[![Colab](https://colab.research.google.com/assets/colab-badge.svg)]({colab\_url})"

lines.append(f"| [{title}]({local\_link}) | {colab\_badge} |")

lines.append("<!-- END CHEATSHEETS TABLE -->")

table\_md = "\n".join(lines)

README\_PATH = "README.md"

if os.path.exists(README\_PATH):

with open(README\_PATH, encoding="utf-8") as f:

readme = f.read()

if "<!-- START CHEATSHEETS TABLE -->" in readme and "<!-- END CHEATSHEETS TABLE -->" in readme:

before = readme.split("<!-- START CHEATSHEETS TABLE -->")[0]

after = readme.split("<!-- END CHEATSHEETS TABLE -->")[1]

new\_readme = before + table\_md + after

print("🔄 Таблица обновлена в существующем README.md между маркерами.")

else:

new\_readme = readme + "\n\n" + table\_md

print("➕ Маркеры не найдены, таблица добавлена в конец README.md.")

else:

new\_readme = "# Шпаргалки\n\n" + table\_md

print("📄 README.md не найден. Создан новый файл с таблицей.")

with open(README\_PATH, "w", encoding="utf-8") as f:

f.write(new\_readme)

Теперь этот скрипт **не будет ругаться на отсутствие README.md** и создаст его сам, если надо.

Хочешь, чтобы он ещё добавлял дату последнего обновления в заголовок или под таблицей?

**🚀 Запуск скрипта обновления таблицы... ❌ Папка 'cheatsheets' не найдена. ❌ Ошибка при запуске скрипта. Проверь generate\_colab\_table.py Press any key to continue . . .**

**Мы остановились на том что скрипт ищет папку, которая в этом репозитории отсутствует. Я так понимаю, что мне нужно эту папку создать и перенести туда ipynb фалы. Так?**

Да, ты всё правильно понял 💡

🔧 Чтобы скрипт работал корректно:

1. **Создай папку cheatsheets** в корне репозитория (там же, где README.md, .bat и generate\_colab\_table.py).
2. **Перемести туда все .ipynb файлы**, которые ты хочешь отобразить в таблице.
3. Убедись, что структура выглядит примерно так:

КопироватьРедактировать

GitPyProba/

├── cheatsheets/

│ ├── intro.ipynb

│ ├── loops.ipynb

│ └── ...

├── generate\_colab\_table.py

├── run\_update.bat

└── README.md

📌 Только после этого скрипт сможет:

* Найти .ipynb файлы,
* Построить таблицу,
* Вставить или создать README.md.

Если хочешь, могу предложить альтернативу: настроить скрипт на любую другую папку или даже текущую директорию. Но вариант с cheatsheets/ — самый аккуратный и удобный. Продолжаем?

**всё супер. а файлы, находящиеся в репозитории, но вне папки cheatsheets/ участвуют в commit and push?**

Ответ: **Да**, участвуют.

В твоём .bat-файле используется команда:

git add .

Она добавляет **все изменения** (новые файлы, изменения, удаления) **во всей папке проекта**, включая:

* README.md;
* .ipynb из cheatsheets/;
* любые другие файлы или папки, находящиеся рядом — если они **не перечислены в .gitignore**.

🔒 Исключения:

* Если какие-то файлы явно исключены в .gitignore, они не попадут в коммит.
* Если ты хочешь ограничить область (например, только README.md и cheatsheets/), придётся указать это вручную в git add.

Кстати, можно настроить .bat, чтобы пушил только определённые файлы/папки.