



# S1: окружение, GitHub и Hello Bot

Python 3.11+ · VS Code · Git/GitHub · pyTelegramBotAPI · dotenv

Практический семинар для начинающих разработчиков

# План семинара

01

## Вступление и цели

Понимание задач семинара

03

## Настройка проекта

venv, зависимости, .gitignore, .env

05

## Hello Bot

Код и запуск бота

02

## Проверка и установка ПО

Python, Git, VS Code

04

## Работа с GitHub

Создание репозитория, первый push

06

## Итоги и ДЗ

Проверка результатов

 **Время:** 1 час 20 минут |  **Формат:** практический семинар

# Цели семинара

## Python окружение

Проверка Python 3.11+, создание виртуального окружения, установка зависимостей

## Настройка IDE

VS Code с расширениями Python, Pylance, GitHub Pull Requests

## Безопасность

Правильная работа с секретами через .env и .gitignore

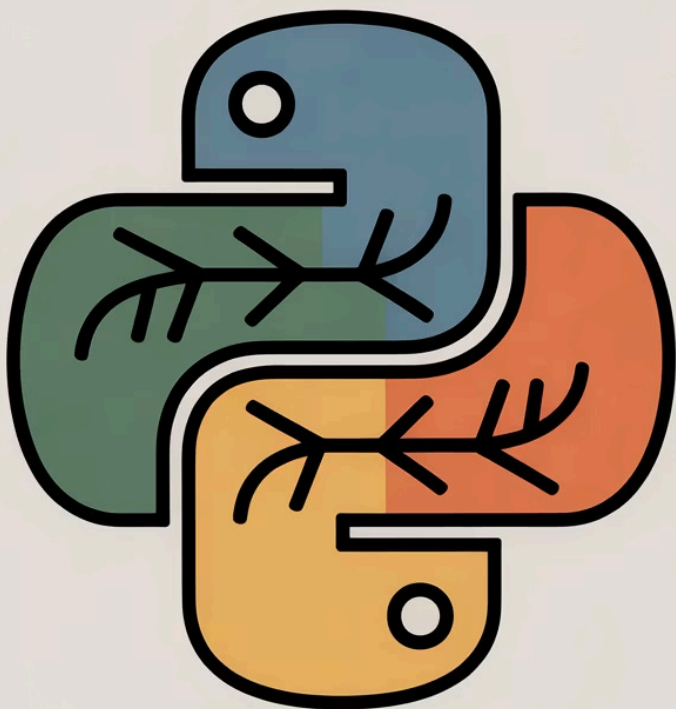
## Git и GitHub

Инициализация репозитория, первый коммит и push

## Рабочий бот

Запуск Telegram-бота с командами /start и /help

✅ К концу семинара у каждого будет локальный проект с настроенным окружением и работающий бот в Telegram!




## ✓ Проверка Python и Git

 Windows

```
py --version  
git --version
```

Ожидаемый результат:

```
Python 3.11.x  
git version 2.x.x
```

 macOS /  Linux

```
python3 --version  
git --version
```

Ожидаемый результат:

```
Python 3.11.x  
git version 2.x.x
```

⚠ Если Python отсутствует: Windows — скачать с [python.org](https://python.org) (отметить "Add to PATH"), macOS — установить официальный .pkg



# Установка VS Code + расширения



VS Code

Скачать с [code.visualstudio.com](https://code.visualstudio.com)



Python

Основное расширение для Python



Pylance

Автодополнение и проверка типов



GitHub

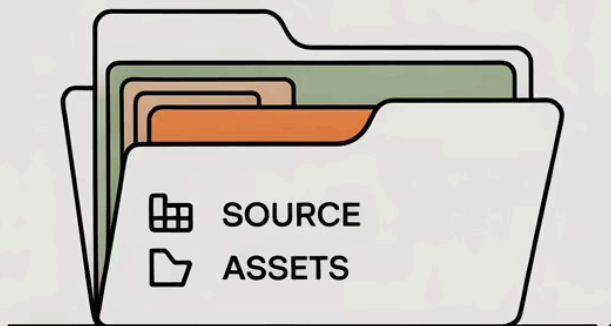
GitHub Pull Requests and Issues

## Настройка интерпретатора

После установки расширений: `Ctrl+Shift+P` → **"Python: Select Interpreter"**

Встроенный терминал: `Ctrl+`` (обратная кавычка)

# project\_code



## Структура проекта

Создать папку

Название: `tg-bot-simple`

Открыть в VS Code

File → Open Folder → выбрать созданную папку

Готовность к работе

Проводник VS Code показывает пустую папку проекта



Рекомендуется хранить проекты в отдельной папке, например `C:\Projects\` или `~/Projects/`



# Виртуальное окружение: Windows

Команды для выполнения:

```
py -3.11 -m venv .venv
```

Создание виртуального окружения

```
.\.venv\Scripts\Activate.ps1
```

Активация окружения

```
py -m pip install -U pip
```

Обновление pip

```
py -m pip install pyTelegramBotAPI python-dotenv
```

Установка зависимостей

❌ При ошибке активации: запустить PowerShell как администратор и выполнить Set-ExecutionPolicy RemoteSigned



# Виртуальное окружение: macOS/Linux

Команды для выполнения:

```
python3 -m venv .venv
```

Создание виртуального окружения

```
source .venv/bin/activate
```

Активация окружения

```
python -m pip install -U pip
```

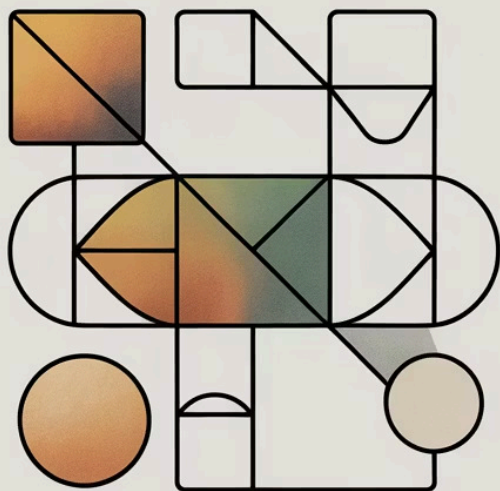
Обновление pip

```
pip install pyTelegramBotAPI python-dotenv
```

Установка зависимостей

После активации в терминале появится `.venv` перед командной строкой





# .gitignore и .env файлы

## Создать .gitignore

```
.venv/  
__pycache__/  
.env
```

Исключаем из Git:

- Виртуальное окружение
- Кэш Python
- Файл с секретами

## Создать .env

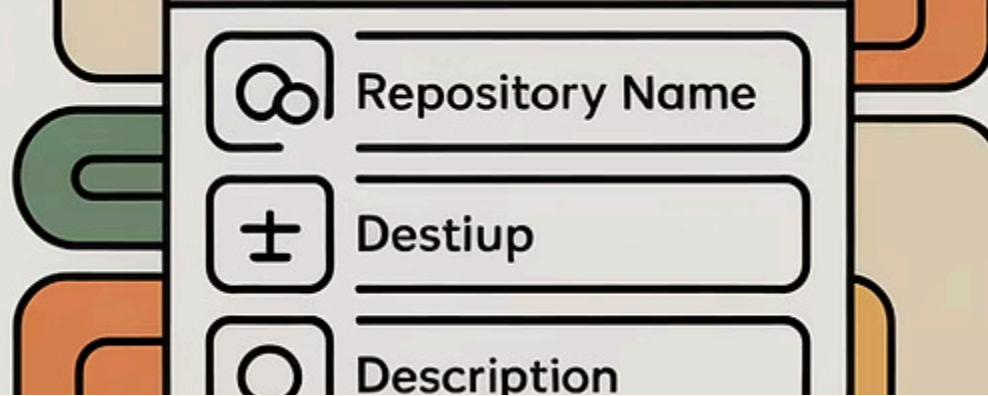
```
TOKEN=ваш_токен_бота
```

Получить токен:

1. Найти @BotFather в Telegram
2. Отправить /newbot
3. Следовать инструкциям
4. Скопировать токен в .env



**КРИТИЧЕСКИ ВАЖНО:** токен бота НИКОГДА не коммитим в Git! Всегда добавляйте .env в .gitignore



## GitHub: создание репозитория

### Вход в GitHub

Зайти на github.com, войти в аккаунт или зарегистрироваться

1

### Настройки

Название: tg-bot-simple  
Тип: Public  
НЕ добавлять README

2

3

4

### Новый репозиторий

Нажать зелёную кнопку "New" или "Create repository"

### Создание

Нажать "Create repository", скопировать URL репозитория

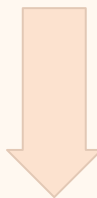


Создаём пустой репозиторий без README, чтобы избежать конфликтов при первом push



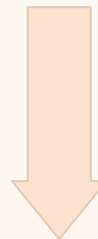
# Git: инициализация и первый КОММИТ

Последовательность команд:



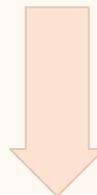
**git init**

Инициализация Git репозитория в текущей папке



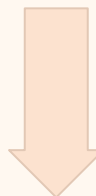
**git add .**

Добавление всех файлов в индекс (.gitignore автоматически исключит ненужные)



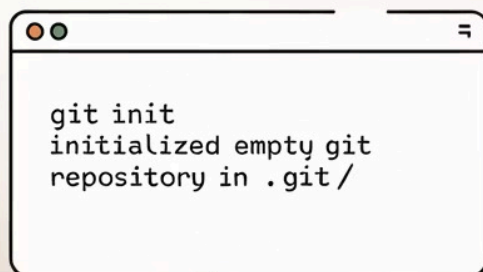
**git commit -m "init: scaffold venv, .env sample, deps"**

Создание первого коммита с осмысленным сообщением



**git branch -M main**

Переименование ветки в main (современный стандарт)



```
git init
initialized empty git
repository in .git/
```



# Привязка к GitHub и push

Финальные команды:

```
git remote add origin URL_вашего_репозитория
```

Связываем локальный репозиторий с GitHub

```
git push -u origin main
```

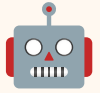
Отправляем код на GitHub (первый push)



## Проверка результата

- Обновить страницу репозитория на GitHub
- Убедиться, что файлы .gitignore появились
- Убедиться, что .env НЕ загрузился (и это правильно!)

Можно добавить README.md для описания проекта



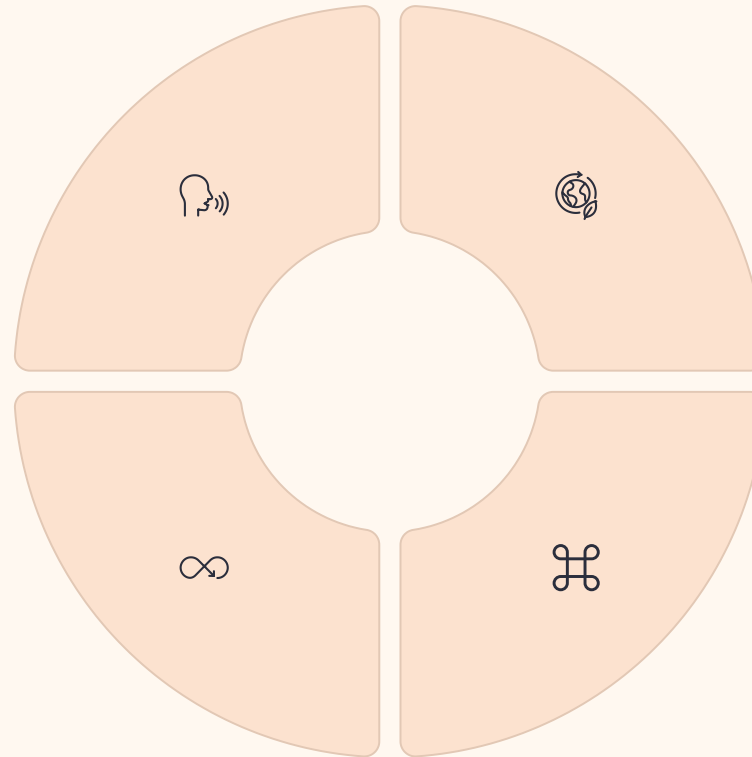
# Hello Bot: концепция

## pyTelegramBotAPI

Библиотека для работы с Telegram Bot API.  
Простая в использовании, хорошо  
документирована

## Polling

Бесконечный цикл опроса Telegram для  
получения новых сообщений



## python-dotenv

Загрузка переменных окружения из .env  
файла для безопасного хранения токенов

## Обработчики команд

Функции, которые реагируют на команды  
/start и /help от пользователей

# Код main.py (часть 1)

Создать файл main.py:

```
import os
from dotenv import load_dotenv
import telebot

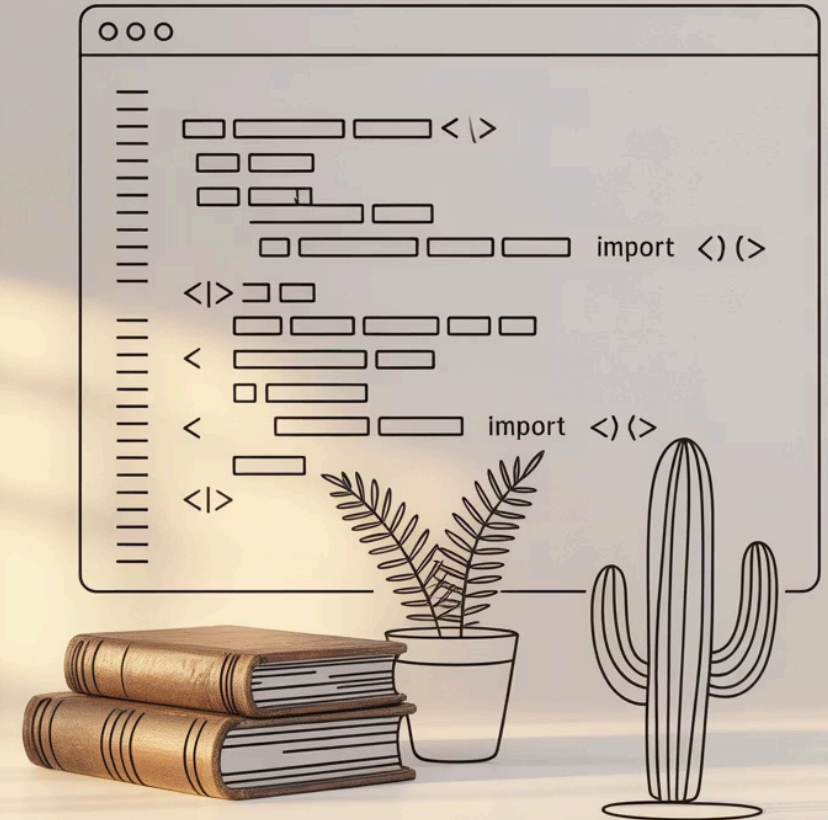
load_dotenv()

TOKEN = os.getenv("TOKEN")
if not TOKEN:
    raise RuntimeError("В .env нет TOKEN")

bot = telebot.TeleBot(TOKEN)
```

## Объяснение кода:

- `load_dotenv()` — загружает переменные из `.env`
- `os.getenv("TOKEN")` — получает токен из переменных окружения
- Проверка токена предотвращает ошибки при запуске
- `telebot.TeleBot(TOKEN)` — создаёт объект бота





## Код main.py (часть 2)

Добавить обработчики и запуск:

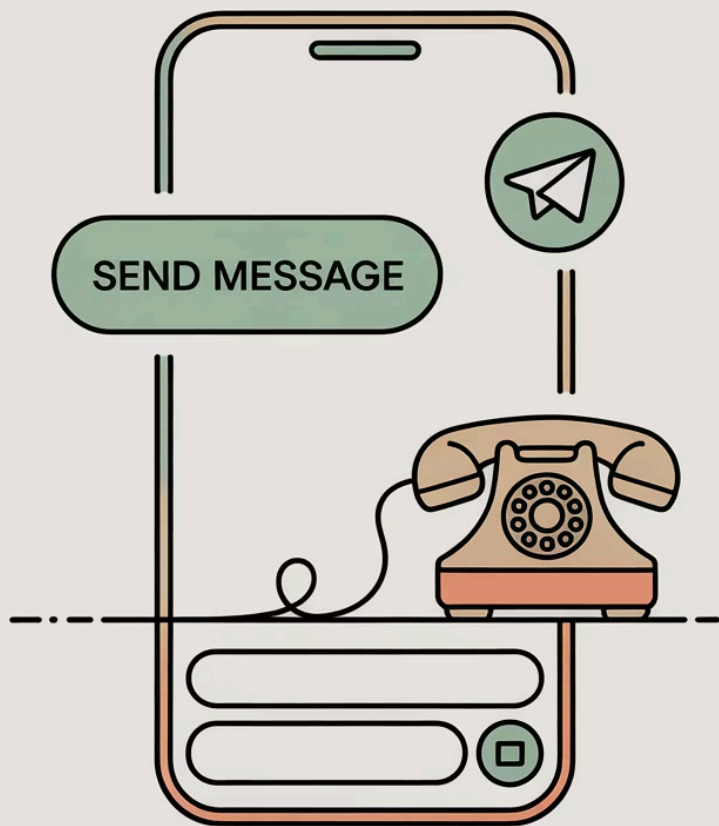
```
@bot.message_handler(commands=['start'])
def start(message):
    bot.reply_to(message, "Привет! Я твой первый бот! Напиши /help")

@bot.message_handler(commands=['help'])
def help_cmd(message):
    bot.reply_to(message, "/start — начать\n/help — помощь")

if __name__ == "__main__":
    bot.infinity_polling(skip_pending=True)
```

### Ключевые моменты:

- `@bot.message_handler(commands=[...])` — декоратор для команд
- `bot.reply_to()` — ответ на конкретное сообщение
- `infinity_polling()` — бесконечный опрос новых сообщений
- `skip_pending=True` — игнорирует старые сообщения при запуске



# Запуск и проверка в Telegram

## Запуск бота:

```
python main.py
```

Ожидаемый вывод:

```
Started polling...
```








## ✓ Чек-лист проверки:

1. Бот запустился без ошибок
2. В консоли появилось сообщение о polling
3. В Telegram найти бота по имени
4. Отправить команду /start
5. Получить приветственное сообщение
6. Отправить команду /help
7. Получить список команд

⚠ Частые ошибки: неправильный токен, бот не запущен, проблемы с сетью.  
Остановка: Ctrl+C



## Таблица прогресса группы

Студент	 Python	 VS Code	 Git	 venv	 .env	 push	 бот
Иванов И.	✓	✓	✓	✓	✓	✓	✓
Петров П.	✓	✓	⌚	⌚	⌚	⌚	⌚
...	...	...	...	...	...	...	...

Ассистенты отмечают готовность каждого этапа. Никто не остаётся позади!

**Легенда:** ✓ готово | ⌚ в процессе | ✗ проблема | ? помощь нужна



# Домашнее задание



## Обязательное

**Команда /about:** информация о боте (автор, версия, назначение)

**README.md:** описание проекта, инструкции по установке и запуску



## Опциональное

**Команда /ping:** простая проверка работоспособности бота

**Логирование:** запись действий бота в файл логов



## Критерии приёмки:

- Репозиторий на GitHub с README
- Бот отвечает на все заявленные команды
- Код структурирован и прокомментирован
- Соблюдена безопасность (токены не в коде)



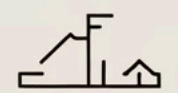
thon



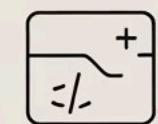
atabase



CSS



Python



Git



Database



## Полезные ссылки



### Python

[python.org](https://python.org) — официальный сайт

Документация, загрузка, обучающие материалы



### VS Code

[code.visualstudio.com](https://code.visualstudio.com) — редактор кода

Расширения, настройка, горячие клавиши



### Git

[git-scm.com](https://git-scm.com) — система версий


Скачать, документация, интерактивное обучение



### GitHub

[github.com](https://github.com) — хостинг репозиторий

Создание аккаунта, публичные и приватные репозитории

 **Telegram:** @BotFather для создания ботов



## 🎉 Поздравляем! Итоги семинара

100%

Готовность

Все студенты завершили основные этапы

5

Ключевых навыков

Python, Git, GitHub, VS Code, Telegram API

1

Рабочий бот

У каждого есть действующий Telegram-бот

### ✅ Что мы сделали:

- Настроили полноценное окружение разработки
- Изучили основы работы с Git и GitHub
- Создали и запустили первого Telegram-бота
- Освоили безопасную работу с токенами



# Appendix: Команды Windows

Полный список команд для Windows:

# Проверка установленного ПО

```
py --version
```

```
git --version
```

# Создание и активация виртуального окружения

```
py -3.11 -m venv .env
```

```
.\.env\Scripts\Activate.ps1
```

# Установка зависимостей

```
py -m pip install -U pip
```

```
py -m pip install pyTelegramBotAPI python-dotenv
```

# Git команды

```
git init
```

```
git add .
```

```
git commit -m "init: scaffold venv, .env sample, deps"
```

```
git branch -M main
```

```
git remote add origin URL_репозитория
```

```
git push -u origin main
```

# Запуск бота

```
python main.py
```



# Appendix: Команды macOS/Linux

Полный список команд для macOS/Linux:

# Проверка установленного ПО

```
python3 --version
```

```
git --version
```

# Создание и активация виртуального окружения

```
python3 -m venv .env
```

```
source .env/bin/activate
```

# Установка зависимостей

```
python -m pip install -U pip
```

```
pip install pyTelegramBotAPI python-dotenv
```

# Git команды

```
git init
```

```
git add .
```

```
git commit -m "init: scaffold venv, .env sample, deps"
```

```
git branch -M main
```

```
git remote add origin URL_репозитория
```

```
git push -u origin main
```

# Запуск бота

```
python main.py
```

# Appendix: Шаблон .gitignore

Полный шаблон .gitignore для Python проектов:

```
# Виртуальные окружения
```

```
.venv/
```

```
venv/
```

```
env/
```

```
# Кэш Python
```

```
__pycache__/
```

```
*.py[cod]
```

```
*$py.class
```

```
# Файлы с секретами
```

```
.env
```

```
.env.local
```

```
.env.*.local
```

```
# IDE файлы
```

```
.vscode/
```

```
.idea/
```

```
*.swp
```

```
*.swo
```

```
# Операционная система
```

```
.DS_Store
```

```
Thumbs.db
```

```
# Логи
```

```
*.log
```

```
logs/
```

```
# Временные файлы
```

```
*.tmp
```

```
*.bak
```



# Appendix: Полный код main.py

Готовый код для копирования:

```
import os
from dotenv import load_dotenv
import telebot

# Загрузка переменных окружения
load_dotenv()

# Получение токена бота
TOKEN = os.getenv("TOKEN")
if not TOKEN:
    raise RuntimeError("В .env файле нет TOKEN")

# Создание объекта бота
bot = telebot.TeleBot(TOKEN)

@bot.message_handler(commands=['start'])
def start(message):
    bot.reply_to(message, "Привет! Я твой первый бот! Напиши /help")

@bot.message_handler(commands=['help'])
def help_cmd(message):
    bot.reply_to(message, "/start — начать\n/help — помощь")

if __name__ == "__main__":
    print("Бот запускается...")
    bot.infinity_polling(skip_pending=True)
```