

# PLAN SZKOLENIA: GitHub + Copilot w praktyce

## 🎯 Cel:

Uczestnicy po szkoleniu będą potrafili:

- Złożyć konto i korzystać z GitHub.
  - Utworzyć i sklonować repozytorium.
  - Pracować z gałęziami (branchami).
  - Zrozumieć podstawy Git-a dzięki interaktywnej grze.
  - Skonfigurować Visual Studio Code i GitHub Copilot.
  - Napisać prosty program wspólnie z AI.
- 

## □ CZĘŚĆ I — WPROWADZENIE DO GITHUBA

### 1. Co to jest Git i GitHub

- **Git** – system kontroli wersji (śledzi zmiany w kodzie).
- **GitHub** – platforma online do hostowania repozytoriów Git i współpracy w zespołach.

#### 🔗 Wyjaśnij pojęcia:

- Commit, Push, Pull, Branch, Merge, Clone, Fork.
  - Repozytorium = „projekt + historia zmian”.
- 

### 2. Zakładanie konta na GitHub

➡ □ <https://github.com>

Pokaz krok po kroku:

1. Kliknij **Sign up**.
  2. Wpisz e-mail, hasło, nazwę użytkownika.
  3. Potwierdź adres e-mail.
  4. (Opcjonalnie) ustaw 2FA.
- 

### 3. Tworzenie nowego repozytorium

1. Kliknij przycisk **New repository**.
2. Nazwij np. `moj-pierwszy-projekt`.
3. Zaznacz opcję „Add a README file”.

4. Wybierz licencję (np. MIT).
5. Kliknij **Create repository**.

🗨 Pokaż:

- Widok plików.
  - Zakładkę **Commits**.
  - Jak dodać nowy plik online.
  - Jak dodać współpracownika (**Settings** → **Collaborators**).
- 

## 4. Klonowanie repozytorium lokalnie

Na komputerze:

```
git clone https://github.com/uzytkownik/moj-pierwszy-projekt.git
cd moj-pierwszy-projekt
code .
```

Wytłumacz:

- `git status`
  - `git add .`
  - `git commit -m "Pierwszy commit"`
  - `git push`
- 

## 5. Nauka przez grę 🎮

🔗 <https://learngitbranching.js.org/>

To bardzo atrakcyjna część — pozwól uczestnikom **interaktywnie** przećwiczyć:

- tworzenie branchy (`git branch feature`)
- przełączanie (`git checkout`)
- łączenie (`git merge`)
- resetowanie (`git reset`)

🗨 Prowadź wspólnie kilka poziomów — np. "Introduction Sequence" i "Ramping Up".

---

# CZĘŚĆ II — GITHUB COPILOT W PRAKTYCE

## 6. Wprowadzenie do GitHub Copilot

Krótką teorią:

- Asystent AI od GitHub, wspierany przez OpenAI.
  - Działa w VS Code, JetBrains, Neovim.
  - Pomaga pisać kod, komentować, testować.
- 

## 7. Instalacja i konfiguracja w VS Code

1. Zainstaluj **Visual Studio Code**:  
<https://code.visualstudio.com/>
  2. Zaloguj się do **GitHub** w VS Code (prawy dolny róg → Sign in).
  3. Zainstaluj rozszerzenie:
    - ☐ *GitHub Copilot*
    - ☐ *GitHub Copilot Chat* (opcjonalnie, jeśli masz dostęp).
  4. Włącz Copilota w ustawieniach.
- 

## 8. Demo: Pisanie prostego programu z Copilotem

### Przykład: „Kalkulator w Pythonie”

1. Utwórz nowy plik `kalkulator.py`.
2. Napisz komentarz:  
3. `# Napisz prosty kalkulator, który dodaje, odejmuje, mnoży i dzieli`  
`dwie liczby`
4. Poczekaj, aż Copilot zaproponuje kod.
  - 💡 Omów, co zrobił i dlaczego.
5. Uruchom:
6. `python kalkulator.py`

### Inne ciekawe przykłady:

- Generowanie listy Fibonacciego.
  - Konwersja temperatur.
  - Prosty serwer Flask.
  - Klasa `Car` z metodą `drive()`.
- 
-

## 9. Pokaz Copilot Chat (jeśli dostępny)

W pasku bocznym → zakładka **Copilot**

Przykładowe pytania:

- „Napisz test jednostkowy dla tej funkcji.”
  - „Wy tłumacz mi, co robi ten kod.”
  - „Zoptymalizuj poniższy fragment.”
- 

## 10. Ćwiczenie dla uczestników 🧡

Niech uczestnicy:

1. Utworzą swoje repozytorium.
  2. Skopiuja prosty projekt (np. kalkulator).
  3. Commit + Push.
  4. Dodadzą nową funkcję z pomocą Copilota (np. potęgowanie).
- 

## 11. Podsumowanie

- GitHub → współpraca i wersjonowanie.
- Copilot → asystent kodowania.
- Wspólnie przyspieszają rozwój projektów.

💬 Zachęć uczestników do:

- Pracy z gałęziami w przyszłych projektach.
  - Korzystania z Copilot Chat do nauki języków.
- 

## 📖 Materiały dla uczestników:

- <https://learngitbranching.js.org>
- GitHub Docs – Getting started
- GitHub Copilot Docs
- VS Code Download