



# Wprowadzenie do Git i Github

dla osób, które nie używają jeszcze Gita i Githuba



# Mini kurs...

To nie jest kompletny kurs Gita.



# Mini kurs...

To nie jest kompletny kurs Gita

To nie jest kurs Gita dla osób, które korzystają już z Gita (i z Github).



# Mini kurs...

To nie jest kompletny kurs Gita

To nie jest kurs Gita dla osób, które korzystają już z Gita (i z Github)

Uczysz się od zera i tylko fundamentów.



# Mini kurs...

To nie jest kompletny kurs Gita.

To nie jest kurs Gita dla osób, które korzystają już z Gita (i z Github).

Uczysz się od zera i tylko fundamentów.

Tłumaczę najprościej, jak potrafię. Pokazuję, jak, ale mówię też, dlaczego.



# Mini kurs...

To nie jest kompletny kurs Gita.

To nie jest kurs Gita dla osób, które korzystają już z Gita (i z Github).

Uczysz się od zera i tylko fundamentów.

Tłumaczę najprościej, jak potrafię. Pokazuję, jak, ale mówię też, dlaczego.

To będzie KRÓTKI kurs.



# Mini kurs...

To nie jest kompletny kurs Gita.

To nie jest kurs Gita dla osób, które korzystają już z Gita (i z Github).

Uczysz się od zera i tylko fundamentów.

Tłumaczę najprościej, jak potrafię. Pokazuję, jak, ale mówię też, dlaczego.

To będzie KRÓTKI kurs.

Pracuję na Windows (ale Ty możesz mieć Linux czy Mac).



# Mini kurs...

To nie jest kompletny kurs Gita.

To nie jest kurs Gita dla osób, które korzystają już z Gita (i z Github).

Uczysz się od zera i tylko fundamentów.

Tłumaczę najprościej jak potrafię. Pokazuję, jak, ale mówię też, dlaczego.

To będzie KRÓTKI kurs.

Pracuję na Windows (ale Ty możesz mieć Linux czy Mac).

Używam edytora Visual Studio Code.





# Mini kurs...

To nie jest kompletny kurs Gita.

To nie jest kurs Gita dla osób, które korzystają już z Gita (i z Github).

Uczysz się od zera i tylko fundamentów.

Tłumaczę najprościej jak potrafię. Pokazuję, jak, ale mówię też, dlaczego.

To będzie KRÓTKI kurs.

Pracuję na Windows (ale Ty możesz mieć Linux czy Mac).

Używam edytora Visual Studio Code.

Co najmniej podstawowa znajomość CSS i HTML mile widziana.



# Mini kurs...

**NIE ZNASZ GITA? POZNAJ GO RAZEM ZE MNĄ I ZACZNIJ  
UŻYWAĆ W SWOICH MAŁYCH PROJEKTACH, CO POZWOLI  
CI POZNAĆ GO NAPRAWDĘ DOBRZE.**

To nie jest kompletny kurs Gita.

To nie jest kurs Gita dla osób, które korzystają już z Gita (i z Github).

Uczysz się od zera i tylko fundamentów.

Tłumaczę najprościej jak potrafię. Pokazuję, jak, ale mówię też, dlaczego.

To będzie KRÓTKI kurs.

Pracuję na Windows (ale Ty możesz mieć Linux czy Mac).

Używam edytora Visual Studio Code.

Co najmniej podstawowa znajomość CSS i HTML mile widziana.



# Dwa powody, dlaczego warto poznać Gita



# Dwa powody, dlaczego warto poznać Gita

Bo przydaje się, gdy pracujesz sam i jest niezbędny, gdy pracujesz w zespole.



# Dwa powody, dlaczego warto poznać Gita

Bo przydaje się, gdy pracujesz sam i jest niezbędny, gdy pracujesz w zespole.

Bo trudno Ci będzie dostać pracę front-end developera (czy programisty) bez znajomości systemu kontroli wersji (Git jest najpopularniejszym systemem kontroli wersji).



# Czego się dowiesz z tego kursu



# Czego się dowiesz z tego kursu

- Czym jest Git i jak go używać (w stopniu bardzo podstawowym).



# Czego się dowiesz z tego kursu

- Czym jest Git i jak go używać (w stopniu bardzo podstawowym).
- Jak korzystać z Github (platforma do umieszczania repozytoriów z wieloma ciekawymi narzędziami) - m.in. pokażę Ci tworzenie repozytorium, umieszczanie repozytorium, klonowanie repozytorium.





# Czego się dowiesz z tego kursu

- Czym jest Git i jak go używać (w stopniu bardzo podstawowym).
- Jak korzystać z Github (platforma do umieszczania repozytoriów z wieloma ciekawymi narzędziami) - m.in. pokażę Ci tworzenie repozytorium, umieszczanie repozytorium, klonowanie repozytorium.
- Korzystanie z interpretera poleceń (PowerShell i Git Bash).



# Czego się dowiesz z tego kursu

- Czym jest Git i jak go używać (w stopniu bardzo podstawowym).
- Jak korzystać z Github (platforma do umieszczania repozytoriów z wieloma ciekawymi narzędziami) - m.in. pokażę Ci tworzenie repozytorium, umieszczanie repozytorium, klonowanie repozytorium.
- Korzystanie z interpretera poleceń (PowerShell i Git Bash).
- Integracja Git - Github - Visual Studio Code.



# Czego się dowiesz z tego kursu

- Czym jest Git i jak go używać (w stopniu bardzo podstawowym).
- Jak korzystać z Github (platforma do umieszczania repozytoriów z wieloma ciekawymi narzędziami) - m.in. pokażę Ci tworzenie repozytorium, umieszczanie repozytorium, klonowanie repozytorium.
- Korzystanie z interpretera poleceń (PowerShell i Git Bash).
- Integracja Git - Github - Visual Studio Code.
- Umieszczanie projektu na Github i Github Pages.



# Jak będziesz się uczył



# Jak będziesz się uczył

- Na przykładach, ale sporo teorii



# Jak będziesz się uczył

- Na przykładach, ale sporo teorii
- Zaczniemy od poleceń w konsoli

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\iMikser> mkdir git_project

Directory: C:\Users\iMikser

Mode                LastWriteTime         Length Name
----                -
d-----         2019-03-04    09:53             git_project

PS C:\Users\iMikser> cd .\git_project\
PS C:\Users\iMikser\git_project> git init
Initialized empty Git repository in C:/Users/iMikser/git_project/.git/
PS C:\Users\iMikser\git_project>
```



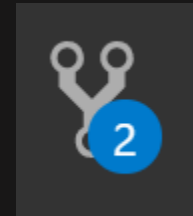
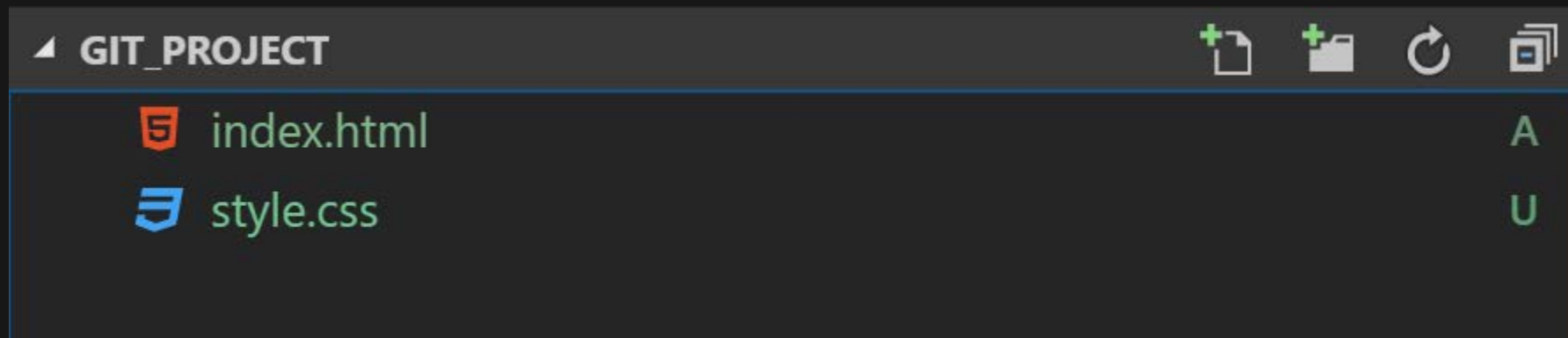
# Jak będziesz się uczył

- Na przykładach, ale sporo teorii
- Zaczniemy od poleceń w konsoli
- Później to samo będziemy robić w interfejsie graficznym VSC



```
index.html x style.css User Settings
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>

TERMINAL 1: powershell
PS C:\Users\iMikser\git_project> git add .
```





# Jak będziesz się uczył

- Na przykładach, ale sporo teorii
- Zaczniemy od poleceń w konsoli
- Później to samo będziemy robić w interfejsie graficznym VSC
- Po drodze pokażę platformę Github i integrację Git i Github

The screenshot shows the GitHub homepage with a dark theme. The main heading is "Built for developers". Below it, a text block describes GitHub as a development platform. On the right, there is a sign-up form with fields for Username, Email, and Password. A green "Sign up for GitHub" button is at the bottom of the form. Below the form, there is a section for "Get started with GitHub Enterprise" with a blue "Contact Sales" button and a link to start a free trial.

Why GitHub? ▾ Enterprise Explore ▾ Marketplace Pricing ▾ Search GitHub [7] Sign in Sign up

## Built for developers

GitHub is a development platform inspired by the way you work. From **open source** to **business**, you can host and review code, manage projects, and build software alongside 31 million developers.

Username  
JusiorFromMordor

Email  
you@example.com

Password  
.....

Make sure it's more than 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

[Sign up for GitHub](#)

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy statement](#). We'll occasionally send you account related emails.

### Get started with GitHub Enterprise

Take collaboration to the next level with security and administrative features built for businesses.

[Contact Sales](#)

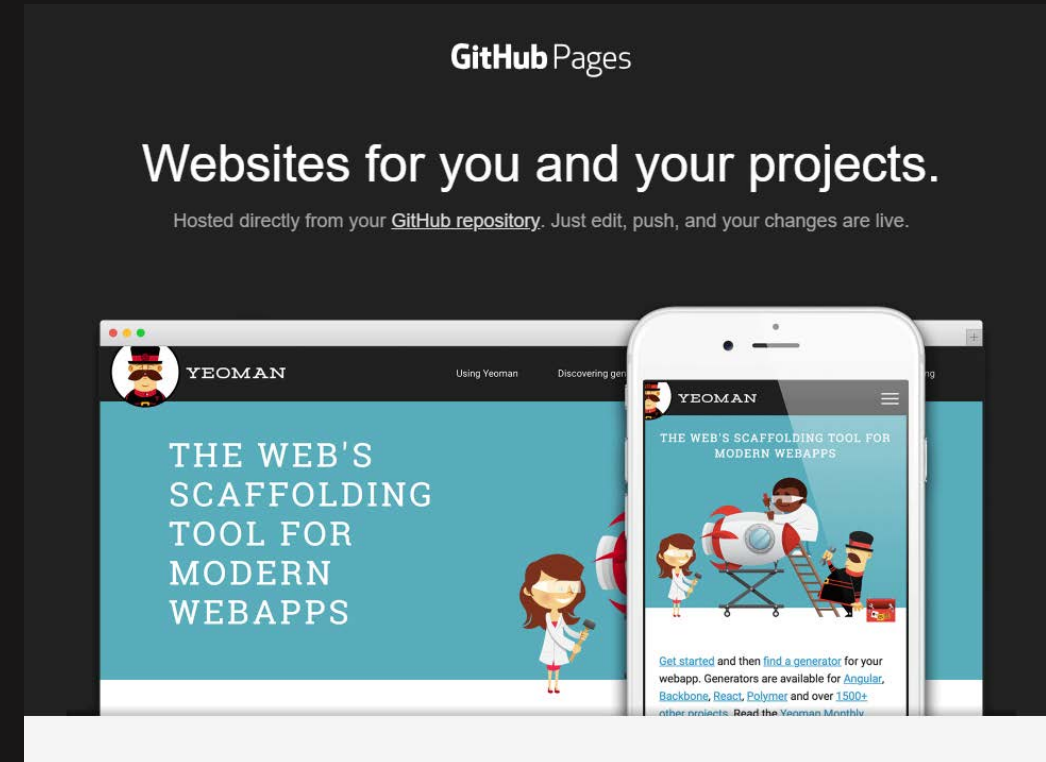
[Or start a free trial of Enterprise Server →](#)





# Jak będziesz się uczył

- Na przykładach, ale sporo teorii
- Zaczniemy od poleceń w konsoli
- Później to samo będziemy robić w interfejsie graficznym VSC
- Po drodze pokażę platformę Github i integrację Git i Github
- Opublikujemy nasze repozytorium na Githubie, ale zobaczysz też, jak opublikować i udostępnić światu projekt za pomocą rozwiązania Github Pages.





# Musi boleć

Bądź cierpliwy i nie irytuj się. Jeśli po drodze pojawi się coś, czego nie rozumiesz, idź dalej. Czasami świadomie wprowadzam jakieś pojęcie, które tłumaczę później. Wiele rzeczy w kursie też powtarzam, co jeśli rozumiesz już zagadnienie, może też trochę irytować.





# Rzeczywistość - kopia zapasowa

Chronimy dane! Sposoby na backup są różne. W najprostszej wersji gdy pracujemy sami, tworzymy wiele folderów z kolejnymi etapami projektów, np. `projekt_v232_5-03-2019`

Może przy małych, jednoosobowych projektach to się sprawdza, pomijając, że dysk nam może się uszkodzić, albo dziecko/znajomy coś skasuje.

Kopia zapasowa tylko na własnym komputerze to zawsze złe rozwiązanie.



# Rzeczywistość - coś zespuliśmy

Nadpisaliśmy, skasowaliśmy, potrzebujemy dostępu do wersji, która posiadała takie i takie rozwiązanie. Musimy łatwo zidentyfikować daną wersję projektu.

Śledzenie postępu, bardzo wiele wersji i możliwość odnalezienia się w chaosie. Porównywanie wersji i zmian. Szybkie i bezpieczne przywracanie starszych wersji lub ich fragmentów - wszystko to jest bardzo istotnym elementem pracy z projektem.



# Rzeczywistość - współpraca w zespole

Jeśli od dwóch do kilkuset (więcej) osób pracuje nad jednym projektem, to mamy pracę zespołową i liczne problemy w obszarze, jak robić to sprawnie i bezpiecznie.

Co zrobić, by jedna osoba nie czekała z pracą na otrzymanie aktualnej wersji, jak rozwiązywać konflikty i łączyć różne wersje. Jak dokumentować pracę, jak pracować niezależnie, ale jednak nad jednym projektem.



# Rozwiązanie to system kontroli wersji

Systemy kontroli wersji starają się rozwiązać te i inne problemy.  
Najpopularniejszym systemem kontroli wersji jest Git.



# git



# Czym jest GIT



# Czym jest GIT

Git to (rozproszony) system kontroli wersji.





# Czym jest GIT

Git to (rozproszony) system kontroli wersji.

Git to najpopularniejszy system kontroli wersji.



# Czym jest GIT

Git to (rozproszony) system kontroli wersji.

Git to najpopularniejszy system kontroli wersji.

Git to oprogramowanie darmowe oraz open source.



# Czym jest GIT

Git to (rozproszony) system kontroli wersji.

Git to najpopularniejszy system kontroli wersji.

Git to oprogramowanie darmowe oraz open source.

Git jest lubiany, wydajny, szybki, rozbudowany.



# Czym jest system kontroli wersji

*ang. version control system*

VCS służy do zarządzania projektem w czasie. Tworzy historię projektu, pozwala na współpracę wielu osób i łączenie ich pracy.



# Czym jest system kontroli wersji

Jest narzędziem, które umożliwia tworzenie repozytoriów\*

*\* repozytorium to pojęcie obejmujące pliki projektu oraz zapis historii zmian. Wyjaśnienie istotnych pojęć Gita (czy ogólnie systemów kontroli wersji) nastąpi w dalszej części kursu.*



# Czym jest system kontroli wersji

To rozwiązanie, które pomaga śledzić zmiany i łączyć kod wielu osób pracujących na projektem.



# Czym jest system kontroli wersji

Sprawną maszyną łączącą pracę wielu deweloperów. Maszyna, która potrafi do tego podróżować w czasie (do poprzednich wersji projektu) oraz podróżować między różnymi światami (**przełączanie między branch'ami**)





# Git jest rozproszonym systemem kontroli wersji

Każdy deweloper (czy użytkownik danego repozytorium Gita) ma własne kompletne repozytorium na swoim komputerze (repozytorium lokalne). Najczęściej istnieje też repozytorium, które określa się jako główne, centralne czy zdalne (bo znajduje się na serwerze, którym często jest platforma Github).





# Git jest rozproszonym systemem kontroli wersji

Każdy deweloper (czy użytkownik danego repozytorium Gita) ma własne kompletne repozytorium na swoim komputerze (repozytorium lokalne). Najczęściej istnieje też repozytorium, które określa się jako główne, centralne czy zdalne (bo znajduje się na serwerze, którym często jest platforma Github).

Zmiany w repozytorium lokalnym są łączone z głównym repozytorium, a lokalne repozytorium może być aktualizowane o zmiany w globalnym repozytorium. By pracować na projekcie, nie jest jednak wymagane stałe połączenie z głównym (zdalnym) repozytorium, ponieważ pracuje się na repozytorium lokalnym i tylko synchronizuje się je z głównym.



# Jak działa Git - commit robi snapshot'a

Git robi **snapshot'y** i zapisuje je w obiektach **repozytorium** projektu.

W pewnym momencie projektu, np. gdy zmodyfikujesz czy dodasz plik(i), możesz zrobić tzw. **commit**, który jest zapisaniem w repozytorium aktualnego stanu projektu, taką stopklatką czy save'm (po angielsku nazywa się taki stan snapshot'em, po polsku czasami tłumaczy się to na migawkę).

Taki snapshot zawiera kompletny stan projektu (wszystkich plików) w projekcie.

I to jest podstawowa, fundamentalna część Gita.



# Czym jest Github

Najpopularniejsza platforma do zdalnego tworzenia i przechowywania repozytorium projektu (opartego na Git). Pełni też rolę backupu oraz udostępnia dodatkowe narzędzia pomagające w pracy nad projektem w grupie.



# Programy wiersza poleceń a Git

Git umożliwia pracę zarówno w oparciu o interfejs wiersza poleceń jak i interfejs graficzny. Naukę Gita warto rozpocząć od nauki poleceń Gita a najlepszy sposób to po prostu wpisywanie komend do CLI (CLI - Command Line Interface).

Popularne programy wiersza poleceń (powłoki tekstowe): Bash, PowerShell, CMD.



# Jakie rozwiązanie dla wiersza poleceń?

Oczywiście korzystać z wiersza poleceń warto się nauczyć (w kontekście Gita, ale nie tylko). Przykładowe, także uniwersalne polecenia poznamy w tym kursie.



# Jakie rozwiązanie dla wiersza poleceń?

Oczywiście korzystać z wiersza poleceń warto się nauczyć (w kontekście Gita, ale nie tylko). Przykładowe, także uniwersalne polecenia poznamy w tym kursie.

Warto wiedzieć, że nie wszystkie komendy w różnych CLI są idenryczne, niektóre interpretery są znacznie uboższe (np. CMD w Windows).

Użytkownicy systemów Linux i Mac korzystają najczęściej z Basha, użytkownicy Windowsa pracując z Gitem, korzystając z PowerShella lub z Git Basha (specjalna emulacja Basha na Windows, którą dostajemy wraz z instalacją Gita).



# Edytor kodu

Git potrzebuje edytora kodu (choć może być i notatnik), nawet jeśli nie korzystamy z interfejsu graficznego (edytory oferują różne dodatki).

Ja polecam Visual Studio Code, ale to Twój wybór. Będę jednak pokazywał, jak korzystać z Gita właśnie w tym środowisku.



# Ciekawostki o Git

Git powstał w 2005 roku (jego pierwszą wersję stworzył twórca Linuxa Linus Torvald, szwedzkojęzyczny Fin).







# Ciekawostki o Git

Git powstał w 2005 roku (jego pierwszą wersję stworzył twórca Linuxa Linus Torvald, szwedzkojęzyczny Fin).

Git został napisany w języku C.





# Ciekawostki o Git

Git powstał w 2005 roku (jego pierwszą wersję stworzył twórca Linuxa Linus Torvald, szwedzkojęzyczny Fin).

Git został napisany w języku C.

Używa algorytmu SHA1, co czyni go też bardzo bezpiecznym (hash, identyfikator).





# Ciekawostki o Git

Git powstał w 2005 roku (jego pierwszą wersję stworzył twórca Linuxa Linus Torvald, szwedzkojęzyczny Fin).

Git został napisany w języku C.

Używa algorytmu SHA1, co czyni go też bardzo bezpiecznym (hash, identyfikator).

Przechowywany w katalogu `/.git` Twojego projektu.

Git robi kopię całego projektu (snapshot), a nie tylko zmian.







# Ciekawostki o Git

Git powstał w 2005 roku (jego pierwszą wersję stworzył twórca Linuxa Linus Torvald, szwedzkojęzyczny Fin).

Git został napisany w języku C.

Używa algorytmu SHA1, co czyni go też bardzo bezpiecznym (hash, identyfikator).

Przechowywany w katalogu `/.git` Twojego projektu.

Git robi kopię całego projektu (snapshot), a nie tylko zmian.

Nazwę Git wymyślił twórca systemu i nazwał go tak, bo taką miał ochotę :) Dziś Git rozszyfrowuje się Global Information Tracker (co można przetłumaczyć jako globalny system śledzenia informacji).





# Co dalej w tym kursie

Instalacja Gita i zabawa z CLI.

Konfiguracja Gita.

Pierwsze repozytorium - katalog roboczy, stage area (index), commit, snapshot.

Kilka podstawowych poleceń Gita i pokazanie, jak to działa.

I powrócimy do prezentacji, by poopowiadać sobie o tym, co pokazałem, czyli rozjaśniamy mroki Gita (wcześniej pokazałem, a teraz spróbuję wytłumaczyć). Pojęcia i teoria.

... oczywiście to nie będzie jeszcze koniec ;)



# Zapraszam!