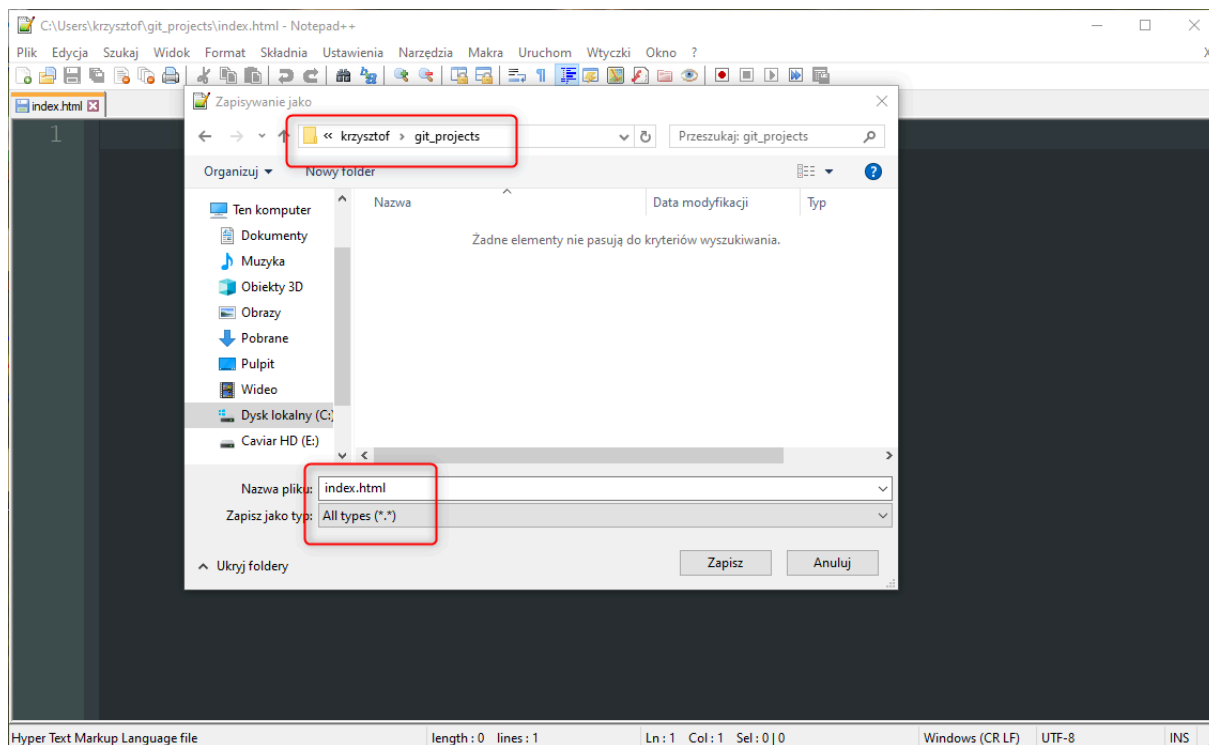


Obsługa repozytorium lokalnego w GIT

UWAGA: JEST TO JEDNA Z NAJWAŻNIEJSZYCH INSTRUKCJI DOTYCZĄCYCH GIT-a

Do naszego katalogu dodajemy plik. Np. index.html . Plik nie zawiera żadnego tekstu.



Teraz możemy sprawdzić jak zareaguje nasz GIT.

Tak przy okazji dlaczego GitHub:

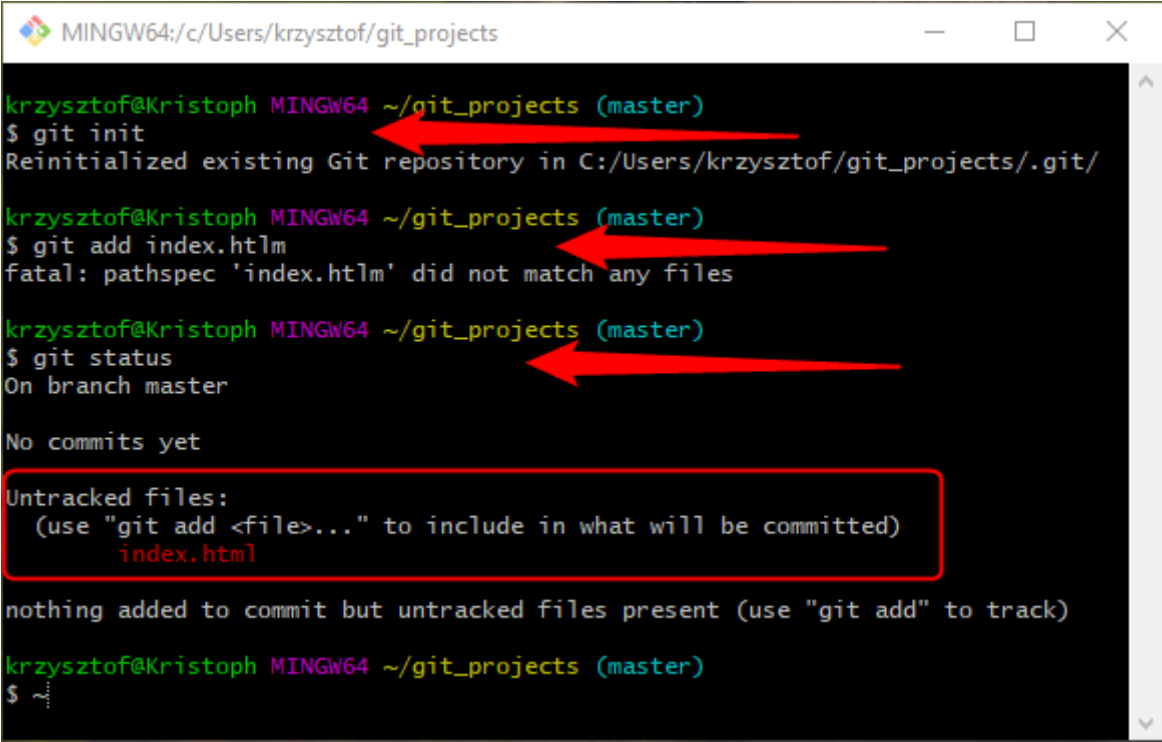
<https://docs.microsoft.com/pl-pl/contribute/get-started-setup-local>

Włączamy Git bash jako administrator i wchodzimy do naszego katalogu i inicjujemy Git-a poleceniem

```
$ git init
```

Aby **rozpocząć kontrolę wersji** istniejących plików (w przeciwieństwie do pustego katalogu), najprawdopodobniej powinieneś rozpocząć ich śledzenie i utworzyć początkową rewizję. Możesz tego dokonać kilkoma poleceniami add (dodaj) wybierając pojedyncze pliki, które chcesz śledzić, a następnie zatwierdzając zmiany poleceniem commit:

```
$ git add nazwa_naszego_pliku
```



```
MINGW64: c:/Users/krzysztof/git_projects

krzysztof@Kristoph MINGW64 ~/git_projects (master)
$ git init
Reinitialized existing Git repository in C:/Users/krzysztof/git_projects/.git/

krzysztof@Kristoph MINGW64 ~/git_projects (master)
$ git add index.html
fatal: pathspec 'index.html' did not match any files

krzysztof@Kristoph MINGW64 ~/git_projects (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html

nothing added to commit but untracked files present (use "git add" to track)

krzysztof@Kristoph MINGW64 ~/git_projects (master)
$ ~
```

W celu dodanie większej ilości plików do śledzenia warto użyć `*` zamiast nazwy plików. Wówczas zaczynamy śledzić zmiany w większej ilości plików. W poniższym przykładzie ponownie dodajemy plik do śledzenia za pomocą polecenia

```
git add *
```

Powoduje to tylko tyle że wydajemy tylko polecenie, które nic nie robi. Jeżeli znalazł by się tam plik kolejny o innej nazwie lub jeżeli zmienilibyśmy zawartość pliku wówczas wyświetlił by się na czerwono i również zaczęły by być śledzone zmiany w tym pliku.

```
MINGW64:/c/Users/krzysztof/git_projects
No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html

nothing added to commit but untracked files present (use "git add" to track)

krzysztof@Kristoph MINGW64 ~/git_projects (master)
$ git add *

krzysztof@Kristoph MINGW64 ~/git_projects (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   index.html

krzysztof@Kristoph MINGW64 ~/git_projects (master)
$
```

Na razie nasz plik jeszcze nie jest oficjalnym plikiem. Jest on śledzony pod kontem zmian i nie znajduje się jeszcze w repozytorium jako zapisany plik.

Dodamy kolejny plik np. style.css i ponownie sprawdzimy status naszego GIT-a. Oczywiście musimy najpierw ten plik utworzyć, np. w Notepad++ :

```
MINGW64:/c/Users/krzysztof/git_projects

krzysztof@Kristoph MINGW64 ~/git_projects (master)
$ git add *

krzysztof@Kristoph MINGW64 ~/git_projects (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   index.html
        new file:   style.css

krzysztof@Kristoph MINGW64 ~/git_projects (master)
$
```

Teraz już śledzimy zmiany w 2 plikach.

Wykonajmy zmianę w pliku html dopisując pierwszą linię kodu i ponownie sprawdzmy status GIT-a.

```
MINGW64:/c/Users/krzysztof/git_projects
(use "git rm --cached <file>..." to unstage)
new file:   index.html
new file:   style.css

krzysztof@Kristoph MINGW64 ~/git_projects (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   index.html
        new file:   style.css

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

krzysztof@Kristoph MINGW64 ~/git_projects (master)
$
```

Tym razem widać, że GIT wykrył zmianę w pliku oraz podpowiada, że możemy nowy plik dodać lub podmienić.

Zrobimy małe doświadczenie wykonując obie wersje sugestii.

Najpierw dodajemy i sprawdzamy status. Oczywiście mamy sytuację taką jak poprzednio czyli podmieniamy plik.

Dopiszmy 2 linie kodu html i sprawdźmy jak się zachowa GIT.

Git status – wykrywa zmiany, a wykonanie

git restore

```
MINGW64:/c/Users/krzysztof/git_projects

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

krzysztof@Kristoph MINGW64 ~/git_projects (master)
$ git restore *

krzysztof@Kristoph MINGW64 ~/git_projects (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   index.html
        new file:   style.css

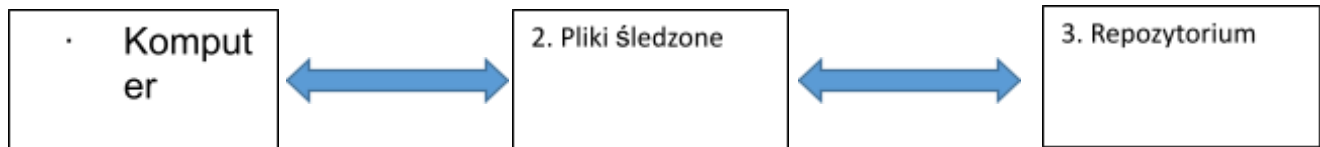
krzysztof@Kristoph MINGW64 ~/git_projects (master)
$
```

powoduje przywrócenie poprzedniej wersji pliku bez wykonania zmian. Hura GIT jest SUPER. Mamy więc możliwość zatwierdzania swoich zmian i usuwania ewentualnych błędów. Jest to krok do przodu.

Nasze plik mogą znaleźć się w 3 miejscach:

1. Komputer
2. Jako plik ślędzony
3. Jako repozytorium

Na razie wiemy co się dzieje w miejscu 1 i 2.



Zatwierdzenie wersji pliku

Kolejnym krokiem jest zatwierdzenie wersji pliku i nie ślędzenie go już. Czyli tworzymy wersję 1.0 ☺

Jednym słowem wrzucamy go do szufladki 3. W tym celu musimy zatwierdzić zmiany za pomocą polecenia

`–m commit`

Przy okazji musimy nadać nazwę repozytorium / wersji. Można posłużyć się nazewnictwem z linyksa co jest wskazane.

Zalecane jest nazewnictwo X.Y.Z co oznacza:

X –zatwierdzona wersja np. 1.0, 2.0, 3.0 itd.

Y – miesiąc w danym roku

Z – numer zmiany w danym miesiącu

Np. Ver. 1.0 z maja z czwartą zmianą będzie mieć oznaczenie 1.5.4

Teraz już wiemy jak zapisać naszą wersję.

Atrybut „-m” oznacza ‘master’ czyli główne repozytorium.

Na poniższym zrzuci widać że pliki zostały dodane do repozytorium Oczywiście w folderze gdzie znajdują się pliki nie zobaczysz zmian (sprawdź):

```
MINGW64:/c/Users/krzysztof/git_projects

krzysztof@Kristoph MINGW64 ~/git_projects (master)
$ git commit -m 'Ver.1.5.1'
[master (root-commit) d98a239] Ver.1.5.1
 2 files changed, 2 insertions(+)
 create mode 100644 index.html
 create mode 100644 style.css

krzysztof@Kristoph MINGW64 ~/git_projects (master)
$ git status
On branch master
nothing to commit, working tree clean

krzysztof@Kristoph MINGW64 ~/git_projects (master)
$ ..
```

Wydanie polecenia **ststus** nie wykaże tym razem plików do śledzenia (jak pokazano wyżej).

Możemy sprawdzić czy nasze pliki w wersji 1.5.1 już znajdują się w repozytorium. Wydajemy polecenie:

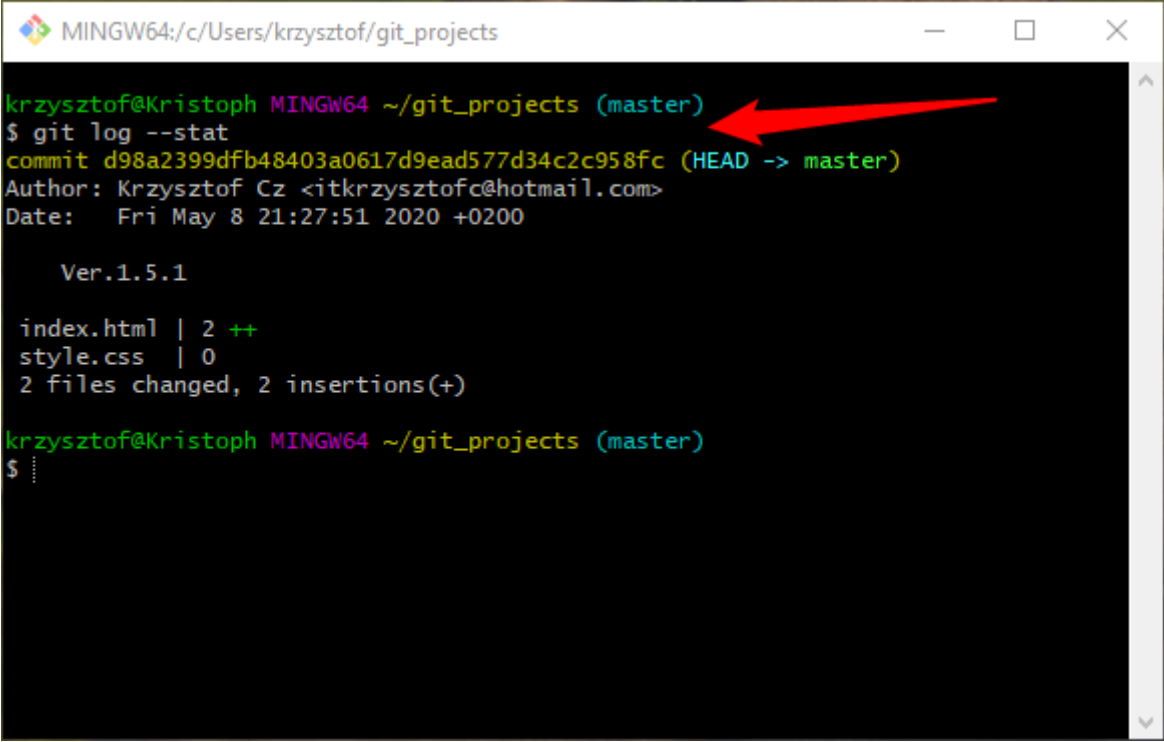
```
MINGW64:/c/Users/krzysztof/git_projects

krzysztof@Kristoph MINGW64 ~/git_projects (master)
$ git log
commit d98a2399dfb48403a0617d9ead577d34c2c958fc (HEAD -> master)
Author: Krzysztof Cz <itkrzysztof@hotmail.com>
Date:   Fri May 8 21:27:51 2020 +0200

    Ver.1.5.1

krzysztof@Kristoph MINGW64 ~/git_projects (master)
$ ..
```

Możemy również sprawdzić statystykę plików czyli ilość zmian zanim zostały zatwierdzone w repozytorium:

A screenshot of a terminal window titled "MINGW64:/c/Users/krzysztof/git_projects". The prompt is "krzysztof@Kristoph MINGW64 ~/git_projects (master)". The command "\$ git log --stat" has been executed. The output shows a commit with hash "d98a2399dfb48403a0617d9ead577d34c2c958fc" (HEAD -> master), authored by "Krzysztof Cz <itkrzysztofc@hotmail.com>" on "Fri May 8 21:27:51 2020 +0200". The commit message is "Ver.1.5.1". The file changes section shows "index.html | 2 ++" and "style.css | 0", with a summary of "2 files changed, 2 insertions(+)". The prompt returns to "\$..". A red arrow points from the right side of the terminal to the commit hash.

```
krzysztof@Kristoph MINGW64 ~/git_projects (master)
$ git log --stat
commit d98a2399dfb48403a0617d9ead577d34c2c958fc (HEAD -> master)
Author: Krzysztof Cz <itkrzysztofc@hotmail.com>
Date:   Fri May 8 21:27:51 2020 +0200

    Ver.1.5.1

index.html | 2 ++
style.css  | 0
2 files changed, 2 insertions(+)

krzysztof@Kristoph MINGW64 ~/git_projects (master)
$ ..
```

Na razie nasze pliki znajdują się w systemie GIT lokalnym , a nie na serwerze zdalnym.