

Jak robić zadania?

v.1.4

Wskazówki

Najpierw przeczytaj

Wszystkie zadania są umieszczone w serwisie github.com.

Aby je wykonać, musisz mieć dostęp do prywatnego repozytorium **CodersLab**.

Jeżeli nie masz dostępu do repozytorium, napisz maila do swojego głównego wykładowcy.

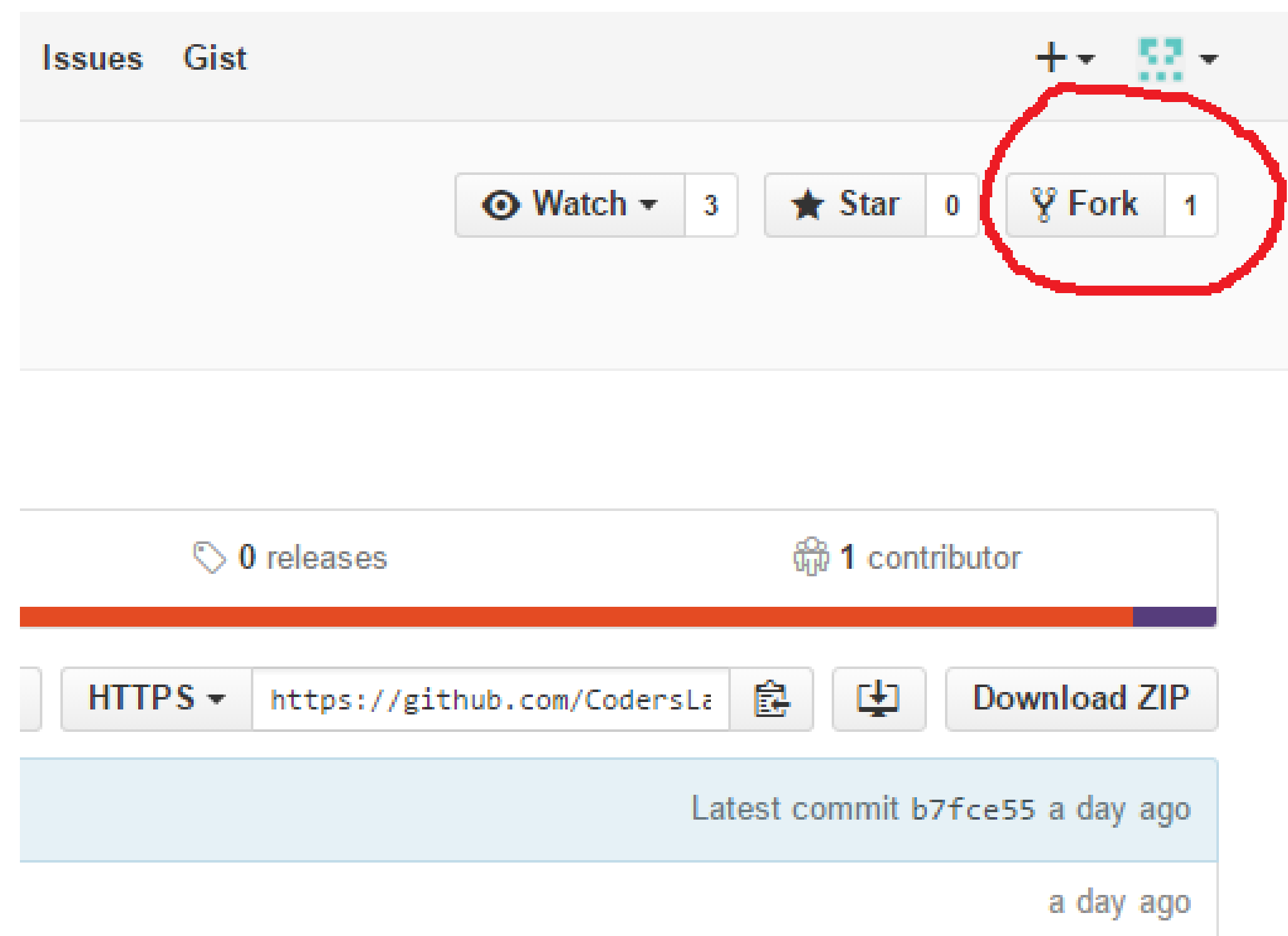
W mailu napisz swój login do githuba, miasto oraz numer grupy lub termin rozpoczęcia kursu.

**Dokładnie przeczytaj każde polecenie.
Wszystko jest opisane krok po kroku.**

Kroki postępowania

1. Zaloguj się do swojego konta na github.com.
2. Wejdź na swoją skrzynkę e-mail i potwierdź dodanie do organizacji **CodersLab** (sprawdź w zakładce SPAM).
Po kliknięciu w link zostaniesz przekierowany na stronę github.com z pytaniem o dołączenie do organizacji.
3. Wejdź na stronę organizacji **CodersLab** i kliknij na repozytorium, na którym chcesz pracować.

4. Kliknij w prawym górnym rogu przycisk **Fork**.



Kroki postępowania

5. Wybierz swój profil. Repozytorium zostanie ustawione na Twoim profilu i będziesz mógł nad nim pracować.

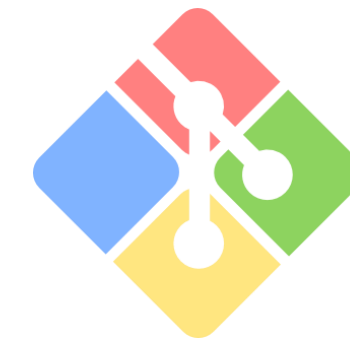


Kroki postępowania

6. Jeżeli jesteś użytkownikiem systemu **Linux** bądź **OS X**, otwórz terminal i przejdź do kroku 8.

Jeśli pracujesz w systemie **Windows** i jeszcze nie masz u siebie na dysku wersji desktopowej aplikacji GitHub, ściągnij ją ze strony <http://desktop.github.com>.
Dzięki temu będziesz mógł korzystać z konsoli i komend dla gita.

7. Po ściągnięciu i zainstalowaniu – znajdź aplikację **Git Bash**.



Kroki postępowania

8. Po otwarciu konsoli znajdź miejsce na dysku, do którego chcesz sklonować repozytorium.

Przykład

Jeżeli swój projekt chcesz umieścić w katalogu o nazwie **prework**, który stworzyłeś na pulpicie, to musisz do niego wejść z pozycji konsoli.

Po katalogach w konsoli poruszamy się za pomocą komendy **cd**.

Wejście do katalogu prework:

```
cd prework
```

Wyjście z katalogu (poziom wyżej):

```
cd ..
```

```
C:\Users\Agata>cd Desktop  
C:\Users\Agata\Desktop>cd prework  
C:\Users\Agata\Desktop\prework>_
```

Jeżeli masz problemy, w linku poniżej znajdziesz więcej informacji, jak poruszać się po katalogach:

<http://www.ubuntu-pomoc.org/moc-zakleta-w-terminalu-cz-1>

Kroki postępowania

9. Gdy wejdiesz już w konsoli do katalogu, do którego chcesz ściągnąć projekt, wpisz polecenie:

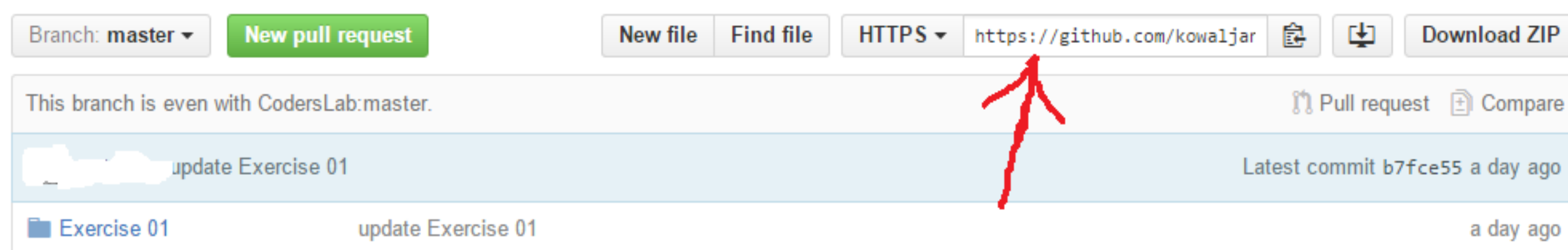
git clone NAZWA_REPOZYTORIUM

gdzie:

NAZWA_REPOZYTORIUM — to adres, który możesz skopiować stąd:

Konsola poprosi Cię o login i hasło. Hasło nie będzie widoczne. Nie martw się zatem, że coś piszesz, a nic się nie dzieje. Po prostu wpisz swoje dane i naciśnij **enter**.

Jeśli wszystko pójdzie dobrze, to Twoje repozytorium zostanie sklonowane na dysk.



Polecenie powinno wyglądać mniej więcej tak:

git clone [https://github.com/kowaljan/Kursanci-707-Advanced HTML CSS.git](https://github.com/kowaljan/Kursanci-707-Advanced_HTML_CSS.git)

Kroki postępowania

- 10.** Przejdź do katalogu z repozytorium, które zostało właśnie sklonowane:

cd Kursanci-707-Advanced_HTML_CSS

Sprawdź, jak u Ciebie nazywa się ten katalog, który sklonowałeś na dysk.

Kroki postępowania

11. Nie zamykaj konsoli. Otwórz projekt w swoim edytorze np. **Atom**, **Brackets** lub innym. Wprowadź zmiany do pierwszego zadania według opisu, który znajdziesz w pliku README.md. Możesz ten plik otworzyć także przez github.com.

css	update exercise 01 - add some styles	14 days ago
README.md	update Exercise 01 - change language	8 days ago
index.html	update Exercise 01	10 days ago

README.md

HTML - Struktura dokumentu

1. Wypełnij plik index.html :
 - dodaj tytuł do strony
 - dodaj plik style.css, znajdujący się w katalogu css
 - dodaj opis strony (tag meta)
 - dodaj słowa kluczowe (tag meta)
2. Dodaj paragraf do body i wypełnij go jakimś dowolnym tekstem np. lorem ipsum

Kroki postępowania

- 12.** Po wprowadzeniu zmian (do pierwszego zadania) musisz wykonać polecenia w konsoli **add** oraz **commit**.

Ta kropka na końcu nie jest przypadkowa! To polecenie doda wszystkie nowe pliki do gita.

git add .

lub

git add index.html

Następnie wpisz:

git commit -m "tutaj wpisz komentarz opisujący to, co zrobiłeś"

Kroki postępowania

- 13.** Jeżeli GitHub będzie Cię prosił o podanie jeszcze raz loginu i maila, wykonaj to w konsoli.

GitHub podpowie Ci podobne polecenie.

Wykonaj oba w konsoli:

```
git config --global user.name "twoj login do GitHuba"  
git config --global user.email "twoj email"
```

Kroki postępowania

- 14.** Aby wrzucić zmiany do repozytorium na GitHubie, musisz wykonać polecenie **push**:

git push

- 15.** Po poprawnym wypchnięciu zmian do GitHuba (czyli jeżeli polecenie **git push** zakończyło się sukcesem), zmiany pojawią się w Twoim repozytorium.

Koniecznienie sprawdź, czy wszystkie zmiany pojawiły się na stronie. Na przykład wejdź (przez stronę) do pliku, który zmieniałeś i sprawdź, czy są Twoje zmiany.

- 16.** Jeśli nie widzisz swoich zmian, to możliwe, że nie dodałeś plików do repozytorium gita. Wykonaj kroki od punktu **11.** jeszcze raz.

Możesz sprawdzać, jakie pliki są dodawane do gita za pomocą polecenia:

git status

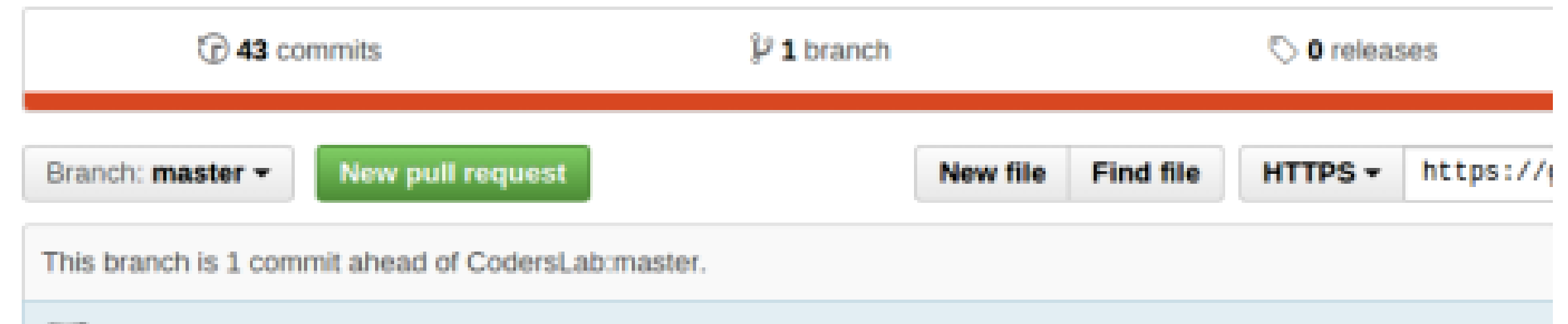
Kroki postępowania

- 17.** Kolejny krok to powtarzanie tych samych czynności dla każdego zadania, czyli od punktu **12**.

Wprowadzasz zmiany do drugiego zadania i wysyłasz je do zdalnego repozytorium na GitHubie. Następnie do trzeciego i tak dalej.

- 18.** Po wykonaniu wszystkich zadań wejdź na stronę github.com, otwórz Twoje repozytorium, na którym pracowałeś, i kliknij przycisk: **New pull request**.

Następnie sprawdź, czy wszystko wygląda ok, i wciśnij: **Create new pull request**.



GRATULACJE

Wykładowca w komentarzu napisze Ci ewentualne uwagi i wskazówki.

W razie problemów skontaktuj się ze swoim głównym wykładowcą.

W razie problemów zobacz filmik na Youtube