

Lista 1

Git, zarządzanie środowiskami, Vetrex AI

Na liście znajdują się dwa zadania do wykonania w trakcie zajęć i jedno dodatkowe oznaczone ★:

1. Git Story Tutorial: Moje ulubione jedzenie (1pkt)
2. Zarządzanie środowiskami (1pkt)
3. ★ Vertex AI (1pkt)

Żeby otrzymać punkty za zadania, po rozwiązaniu zadania trzeba pokazać rozwiązanie i odpowiedzieć na „pytanie kontrolne”.

Zadanie 1. Git Story Tutorial: Moje ulubione jedzenie 🍲

W tym zadaniu nauczysz się podstaw Gita, tworząc repozytorium związane z Twoimi ulubionymi i mniej lubianymi potrawami.

Wyobraź sobie, że chcesz stworzyć książkę kucharską. W tym celu utwórz listę dań które lubisz i tych które mniej lubisz. Następnie, żeby "nie zaśmiecać" głównej gałęzi stworzysz gałąź, na której będziesz opracowywać przepis na jedno z dań. Na końcu wszystkie zmiany umieścisz na gałęzi głównej.

(Bardziej realistyczny scenariusz: Piszysz program. W trakcie tworzysz nowe komponenty nad którymi pracujesz na oddzielnych gałęziach. Podczas pracy nad nimi modyfikujesz kod i popełniasz błędy.)

Tworzymy repozytorium

Stwórz folder o nazwie `jedzenie` i zainicjuj repozytorium git.

```
mkdir jedzenie
cd jedzenie
git init
```

Dodaj folder i pliki

Stwórz folder `PRZEPISY_IMIE` (Twoje imie). Umieść w nim dwa pliki. Pierwszy o nazwie `imie.txt`, w którym wpiszesz swoje imie. W drugim pliku `moje_ulubione_jedzenie.txt` umieść nazwę dania, którego nie lubisz.

```
mkdir PRZEPISY_IMIE
cd PRZEPISY_IMIE
echo "IMIE" > imie.txt
echo "Brokuły" > moje_ulubione_jedzenie.txt
```

Dodaj, commit i push

Umieść zmiany w zdalnym repozytorium. Pamiętaj o dodaniu do commita odpowiedniego komentarza.

```
git add imie.txt moje_ulubione_jedzenie.txt
git commit -m "Dodano imię i ulubione jedzenie którego nie lubię"
git remote add origin <URL_Twojego_repozytorium>
git push -u origin main
```

Dodaj kolejne jedzenie i zobacz różnice

Dodaj do pliku `moje_ulubione_jedzenie.txt` inne jedzenie, które lubisz. Co pokazuje komenda `git diff`? Umieść zmiany w zdalnym repozytorium razem z odpowiednim komentarzem.

```
echo "Pizza" >> moje_ulubione_jedzenie.txt
git diff
git add moje_ulubione_jedzenie.txt
git commit -m "Dodałem moje ulubione jedzenie"
git push
```

Podgląd historii zmian

Zbadaj repozytorium za pomocą `git log`, `git show` i `git status`. Zobacz też wynik `git log` z flagami `--graph --decorate --oneline`.

```
git log
git log --graph --decorate --oneline
git show
git status
```

Revert – poprawiamy błąd

W pliku `moje_ulubione_jedzenie.txt` został popełniony błąd! Użyj `git revert`, żeby naprawić zmiany, a potem dodaj swoje ulubione jedzenie do `moje_ulubione_jedzenie.txt` i dodaj zmiany do repozytorium.

(Czy trzeba poprawić błąd za pomocą revert? Jest prostszy sposób?)

```
git revert HEAD
echo "Makaron" > moje_ulubione_jedzenie.txt
git add moje_ulubione_jedzenie.txt
git commit -m "Poprawiłem wpisane ulubione jedzenie"
git pull
git push
```

Tworzymy nową gałąź `przepis`

Utwórz nową gałąź o nazwie `przepis`. Przełącz się na nią i dodaj plik o nazwie `przepis.txt` w którym napisz przepis na jakieś danie. Następnie umieść go w zdalnym repozytorium na gałęzi `przepis`.

```
git checkout -b przepis
echo "Przepis na makaron: gotuj wodę, dodaj makaron, przypraw" >
przepis.txt
git add przepis.txt
```

```
git commit -m "Dodano przepis na ulubione danie"
git push -u origin przepis
```

Wracamy do main i dodajemy kolejny plik

Wróć na główną gałąź. Utwórz plik `moje_nieulubione_danie.txt` i napisz w nim danie, którego nie lubisz. Umieść zmiany w zdalnym repozytorium.

```
git checkout main
echo "Śledzie" > moje_nieulubione_danie.txt
git add moje_nieulubione_danie.txt
git commit -m "Dodano moje nieulubione danie"
git push
```

Sprawdzenie nazwy gałęzi i scalanie

Połącz gałąź `przepis` z gałęzią główną.

```
git branch
git merge przepis
git push
```

Zadanie 2. Zarządzanie środowiskami

W tym zadaniu stworzysz środowisko, w którym będziesz pracować na kolejnych zajęciach.

Utwórz nowe środowisko Python za pomocą jednego wybranego menedżera:

- `pip` + `venv`,
- `miniconda`,
- `conda-forge`.

Nadaj mu nazwę `numpy_lab`. W środowisku zainstaluj bibliotekę `numpy`

Sprawdź poprawność instalacji, uruchamiając interpreter w nowym środowisku i wpisując:

```
import numpy
print(numpy.__version__)
```

Wygeneruj plik konfiguracyjny środowiska, tak żebyś mógł go przekazać innej osobie.

Usuń obecne środowisko i przetestuj odtworzenie go, czyli wczytaj je z wygenerowanego pliku yml.

Zadanie 3. Vertex AI

Wejdź na stronę <https://cloud.google.com/vertex-ai/docs/tutorials#workbench>. Korzystając z Vertex AI Workbench wykonaj jeden z tutoriali:

- Train a classification model for tabular data
- Train a regression model for tabular data

Żeby zaliczyć zadanie, na następnych ćwiczeniach pokaż, że z sukcesem uruchomiłeś notatnik i przeszedłeś tutorial. Zostaniesz poproszony o wytłumaczenie jednej z komórek.