

Bootcamp Data Science

zjazd 1 - GIT

Oliwia Wojtkowska



Konsola linuxa

SHELL

Program, który otrzymuje komendy i przekazuje do system operacyjnego

Tab podpowiada

Ctrl+C – w bezpieczny sposób

Ctrl+Z wymusza zakończenie działania

Strzałka do góry podpowiada polecenia

LINUX - polecenia

- **Ctrl-Alt-T** - otwarcie terminala
- **Pwd** - print working directory
- **Cd** - change directory np. `cd /home/USERNAME/Desktop`
- **cd ~** - Przejście do katalogu domowego tzw home directory. – obecne miejsce
- **cd ../..** - wychodzenie wyżej
- **mkdir** – make directory – tworzenie katalogu
- **rmdir** – remove directory – usuwanie katalogu
- **ls** – list – wyświetlenie zawartości katalogu
- **ls > output.txt** – zapisywanie wyniku polecenia do pliku
- **touch** – tworzenie nowego pliku np. `touch file.txt`
- **rm** – usuwanie pliku np. `rm file.txt`
- **cp** – copy [-opcje] źródło cel
- **Mv** – move [-opcje] źródło cel
- **head** - Wyświetl początkową część pliku
- **tail** - Wyświetl końcową część pliku
- **cat** - Połącz pliki i wyświetl zawartość na standardowe wyjście

VIM – obsługa plików

Vim – wejście do edytora tekstowego np. vim plik.txt

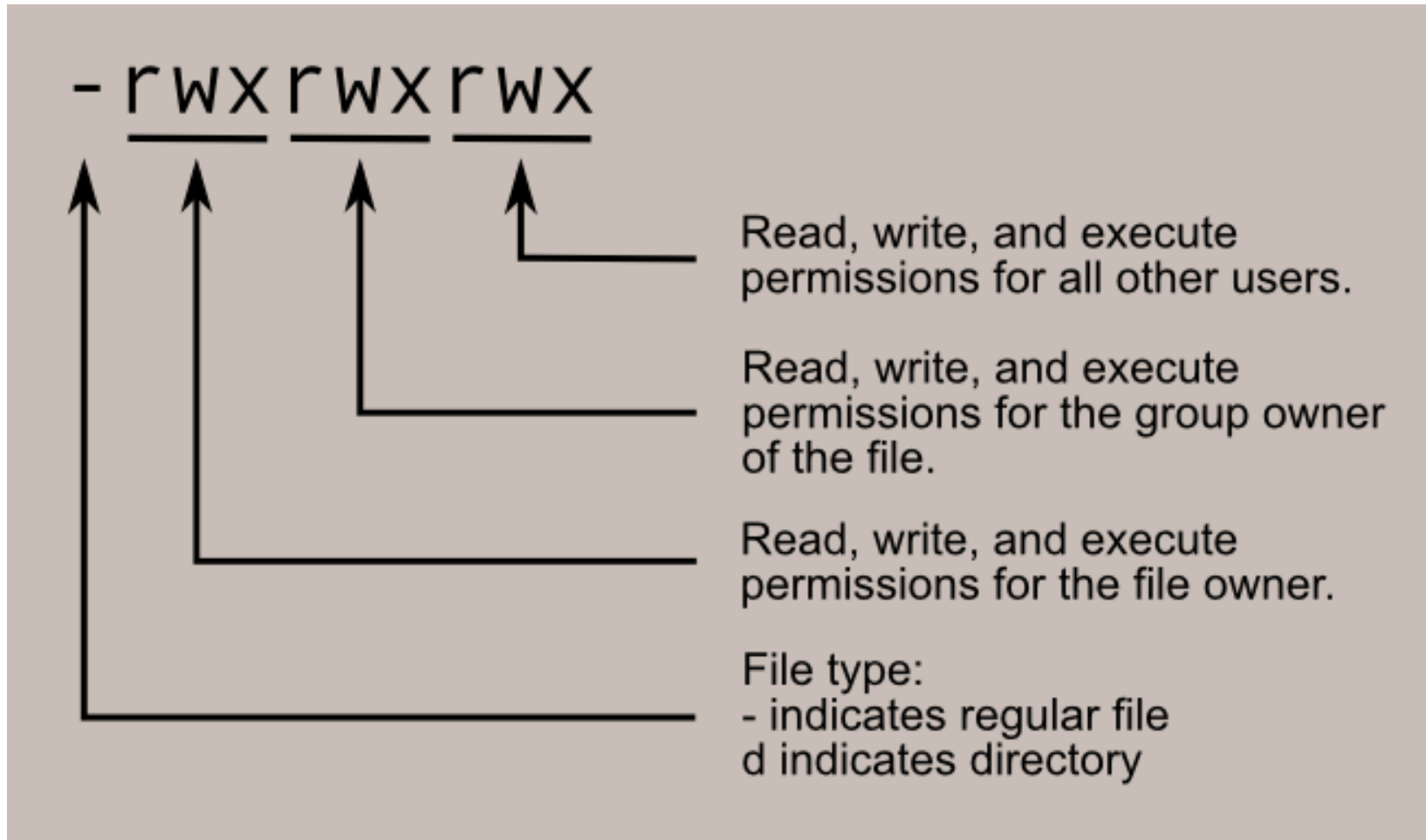
Esc – w edytorze tekstowym wejście w tryb komend

a lub SHIFT+A - w edytorze tekstowym wejście w tryb pisania

:wq – wyjście z zapisaniem z edytora tekstowego:

q! – wyjście bez zapisania z edytora tekstowego

Uprawnienia



Zadanie: shell

W dowolnej lokalizacji stwórz katalog o nazwie *bootcamp*, w środku którego stworzysz katalog o nazwie *zjazd_3*

Za pomocą “cd” przejdź do katalogu *zjazd_3* i stwórz w nim plik ‘*python_code.py*’

Wpisz do niego ‘*print(‘Hello’)*’, zapisz i wyjdź. Za pomocą “ls” sprawdź, czy plik został stworzony. Użyj “cat”, aby zobaczyć jego zawartość.

Wykonaj kod wpisując w konsoli ‘*python python_code.py*’

Na ekranie powinien pojawić się napis ‘Hello’

Usuń plik *python_code.py*

Usuń katalog *bootcamp*



GitLab





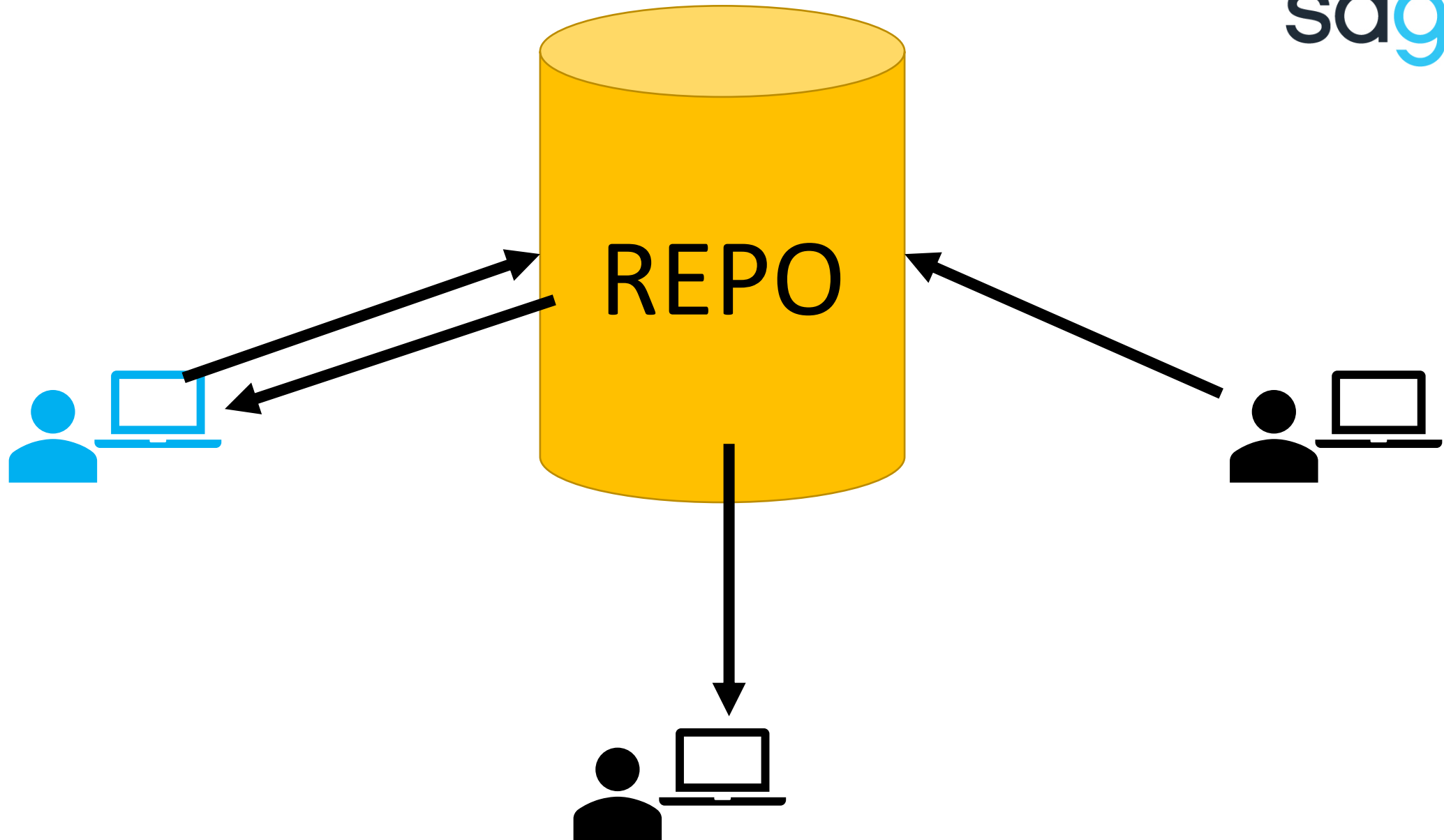
To jak mamy konto na github.com:

Wysłać oliwii adres @ konta lub nazwę usera

To w konsoli:

```
git config --global user.name "Jan Nowak"
```

```
git config --global user.email jannowak@example.com
```





git

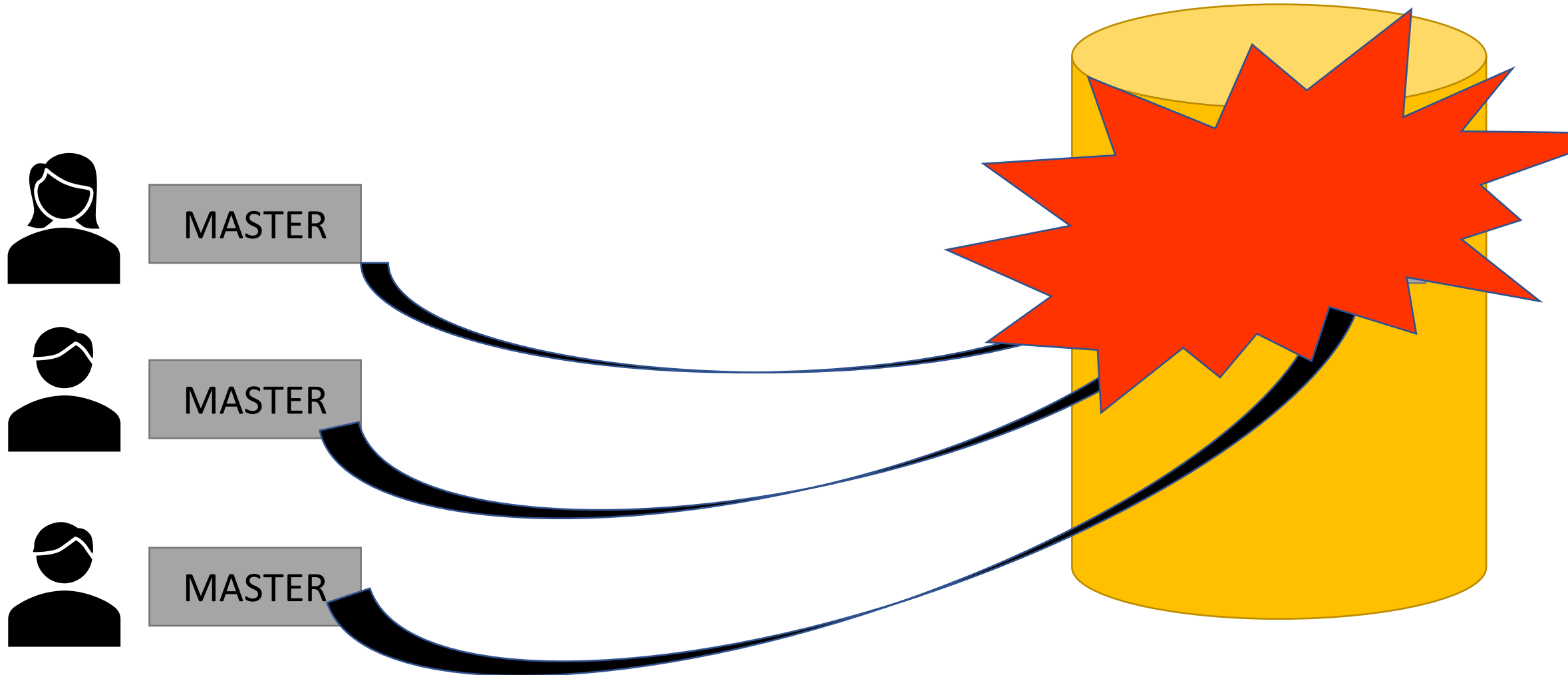
sages





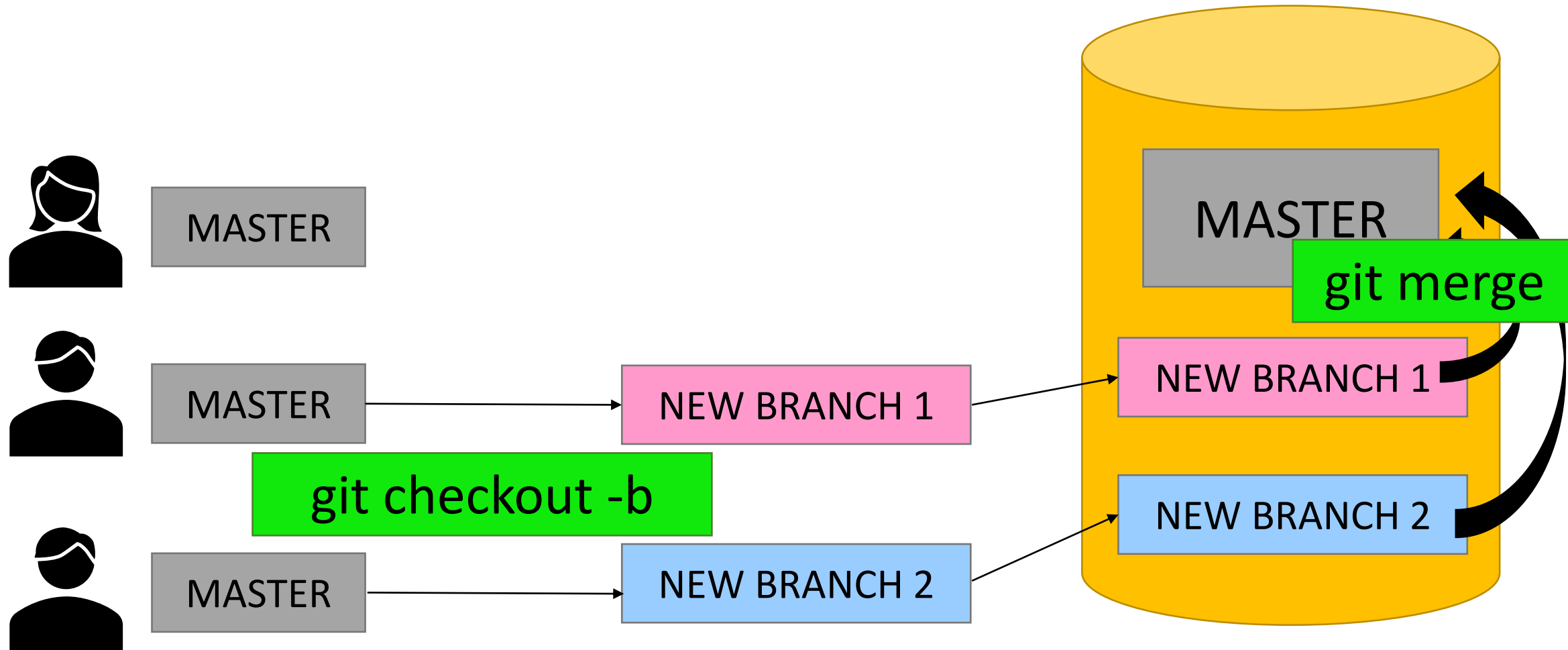
git

sages

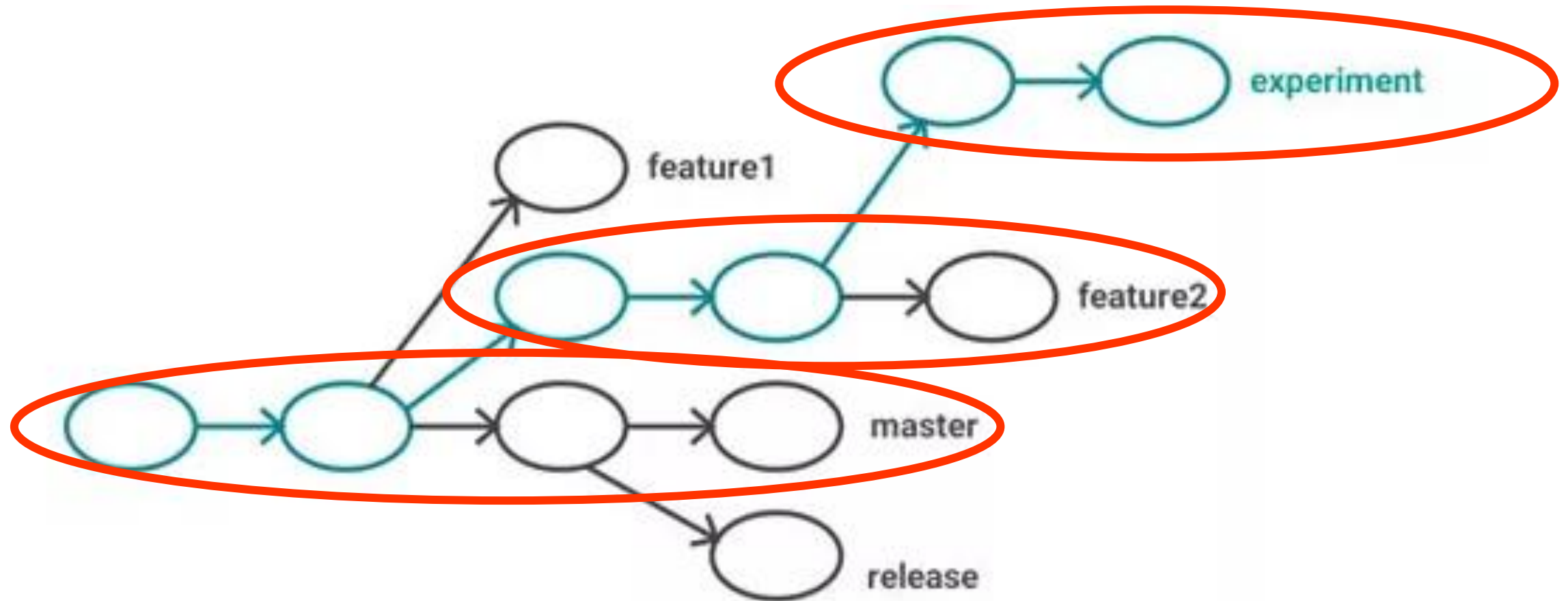




sages



Czym jest Branch?





git

sages

CZY REPO JUŻ ISTNIEJE W OGÓLE?

NIE

git init – tworzy
nowe repozytorium

TAK

**CZY REPO JUŻ ISTNIEJE NA
NASZYM KOMPUTERZE?**

NIE

git clone – klonuje już istniejące
repozytorium tworząc na naszym
komputerze kopie lokalną

TAK

git pull – ściąga zmiany
ze zdalnego repozytorium na
nasze lokalne



git

sages





git add – dodaje pliki do obserwowania

git commit – migawka z obecnego stanu pliku

git push – przesyła zmiany do repo



git status – pokazuje status naszych zmian

git checkout – zmiana branch'a

Pobieramy repozytorium na swój computer:

```
git clone https://github.com/bootcamp-zjazd2/zjazd\_1.git
```

Tworzymy własny branch od mastera:

```
git checkout -b nazwa_branch'a
```

(tworzymy plik .py z rozwiązaniem zadania 33)

Ścieżka wypchnięcia pliku do repozytorium:

```
git add moj_plik.py
```

```
git commit -m 'moj opis co sie stalo' – może być kilka
```

```
git push origin nazwa_branch'a
```

opcjonalnie : możemy sprawdzić status zmienionych plików:

```
git status
```

(robimy merge swojego branch'a do mastera w przeglądarce)

Możemy teraz pobrać pliki innych osób:

```
git checkout master
```

```
git pull origin master
```