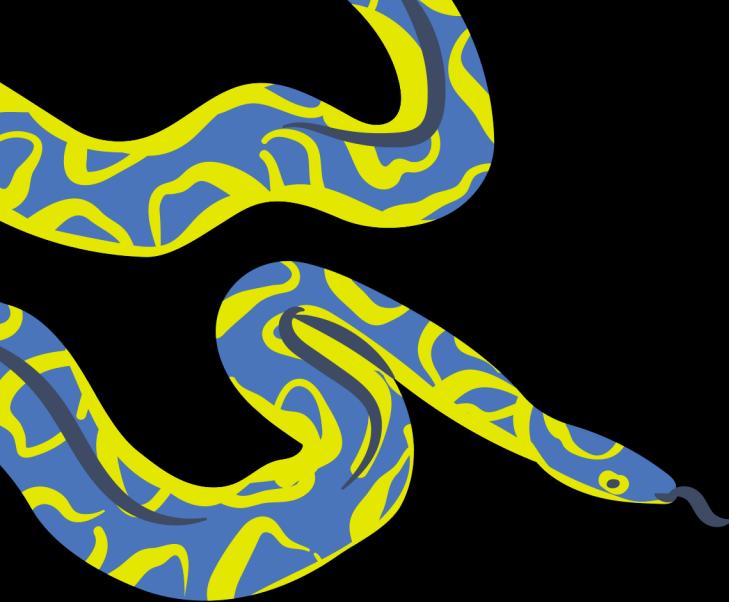




SciByteHub



Python

16.04



ssh-keygen

SciByteHub / Python-Kurs-ST-24

Type to search

Code Issues Pull requests Actions Projects Security Insights Settings

General Deploy keys / Add new

Access

Collaborators Moderation options

Code and automation

Branches Tags Rules Actions Webhooks Environments Codespaces Pages

Security

Code security and analysis Deploy keys Secrets and variables

Integrations GitHub Apps Email notifications

Title

Key

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'.

Allow write access

Can this key be used to push to this repository? Deploy keys always have pull access.

Add key



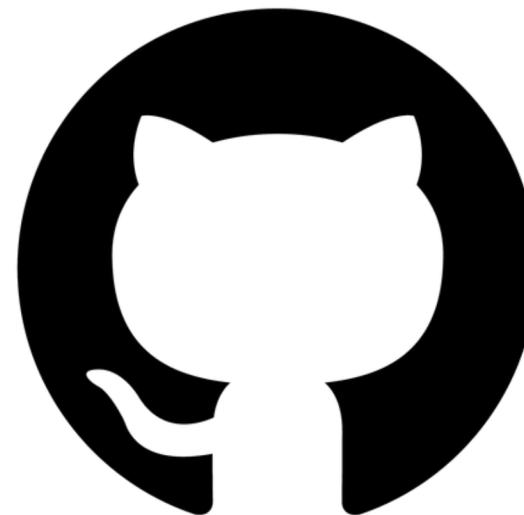
sudo apt install git

git config --global user.name takjaksienazywam

git clone nazwa_repo



SciByteHub



GitHub

Komendy

git init

inicjalizuje nowe repozytorium

git clone

kopiuje istniejące repozytorium

git pull

pobiera zmiany

git push

wysyła zmiany

git add *nazwa*

dodaje plik

git commit -a

zatwierdza wszystkie zmiany

git commit -m "*msg*"

zatwierdza zmiany z wiadomością bez edytora

git status

wyświetla stan repozytorium

git remote add *nazwa url*

dodaje nowe repozytorium

<https://github.com/SciByteHub/Python-Kurs-ST-24>



Algorytm

Idziecie na tour po pubach na starówce.

Ty chcesz wypić perłę chmielową, a Twój znajomy specjalą.

Jak znajdziesz pub, w którym oferują obie te opcje?

Napisz schemat algorytmu krokowego
(uwzględnij co jest inputem, a co outputem).

Python is the
easier language
to learn.
No brackets,
no main.



You get errors
for writing an
extra space





Wcięcia

if warunek1:
 = instrukcja 1

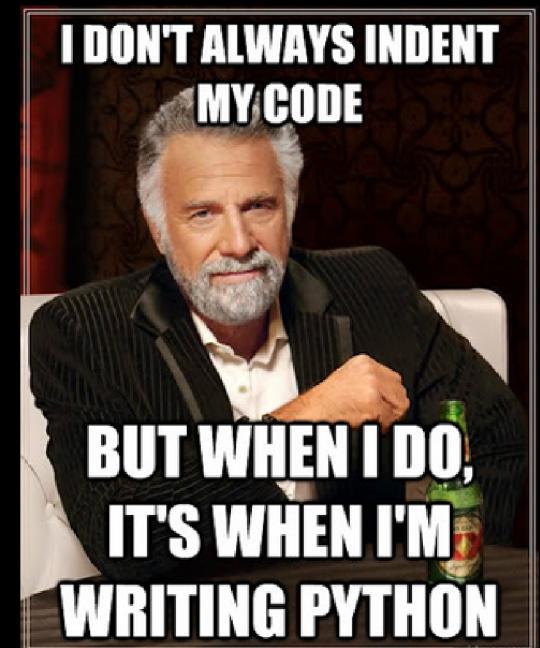
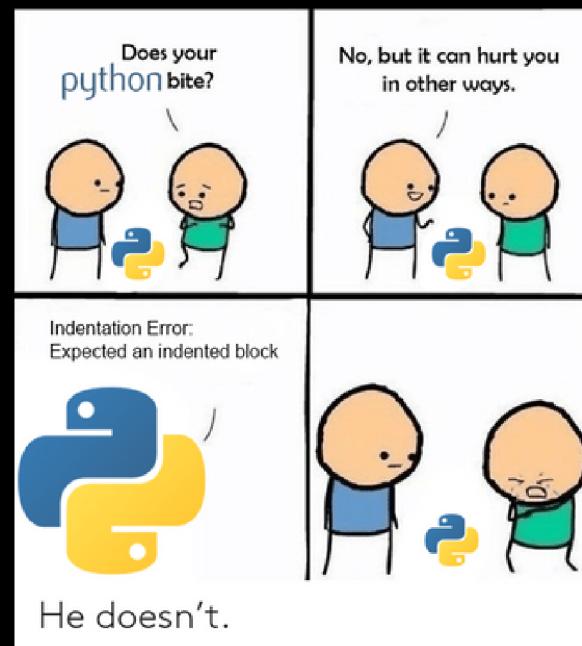
elif warunek2:
 = instrukcja 2

else:
 = instrukcja 3

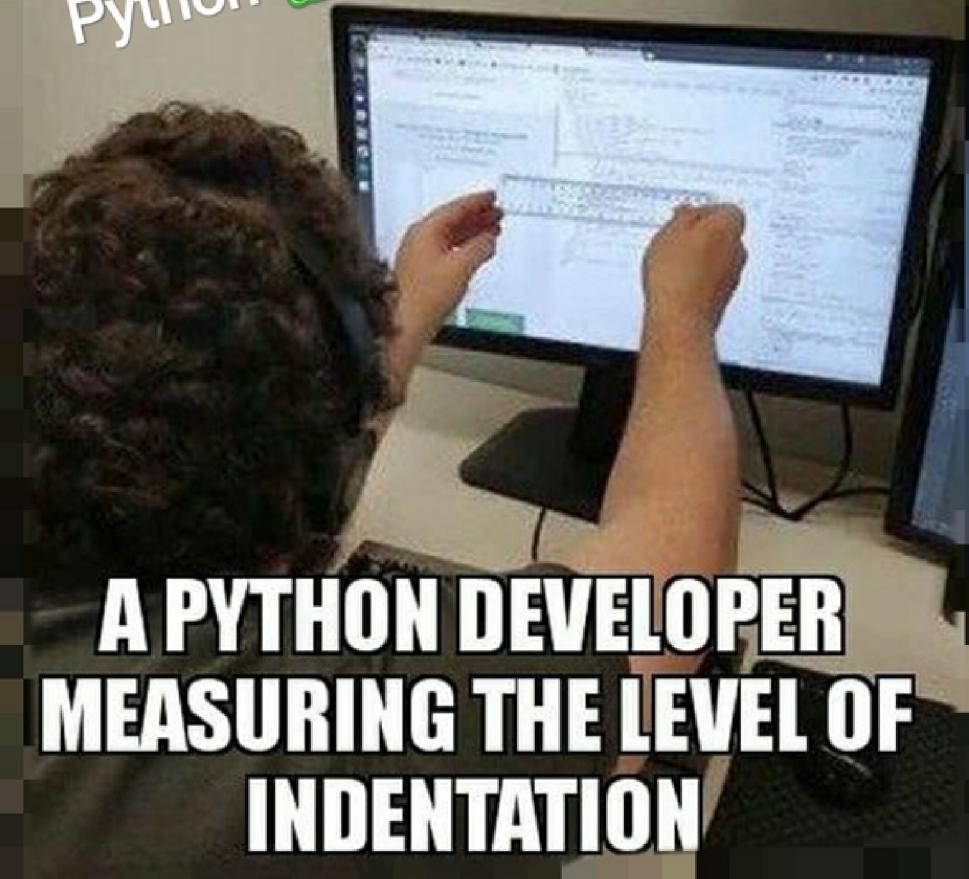
Grupowanie poleceń

Błędne wcięcia (indentation) dają błąd:

IndentationError: unexpected indent



Python 🐍



**A PYTHON DEVELOPER
MEASURING THE LEVEL OF
INDENTATION**



if warunek1:
instrukcja 1

Jeśli warunek1 jest TRUE wykonuje instrukcja1.

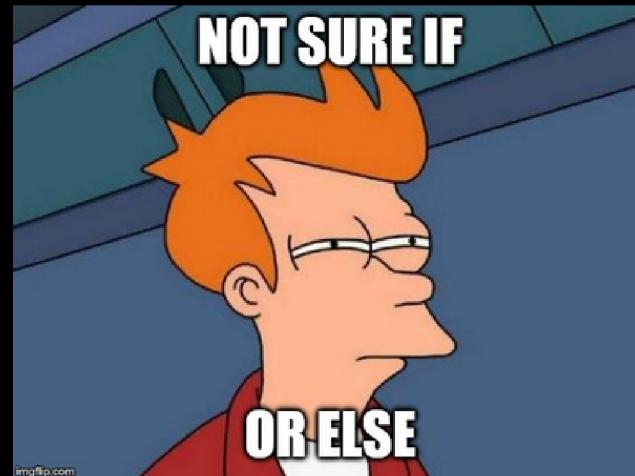
elif warunek2:
instrukcja 2

Jeśli warunek1 jest FALSE, sprawdza warunek2:
jeśli warunek2 jest TRUE, wykonuje instrukcja2.

else:
instrukcja 3

W przeciwnym wypadku (else), tzn. jeśli warunek2 również jest FALSE, wykonuje instrukcja3.

| F





```
wiek = 20
```

```
if wiek >= 18:
```

```
    print("Możesz kupić piwo!")
```

```
else:
```

```
    print("Nie możesz kupić piwa :(")
```



Zadanie 1: IF

```
preferencja=['ciemne','gorzkie','mocne']

lager = ['rześkie', 'jasne', 'lekkie']
pils = ['jasne', 'goryczkowe', 'średnio-mocne']
porter = ['gorzkie', 'mocne', 'ciemne']
stout = ['ciemne', 'kawowe', 'bogate']
bezalkoholowe = ['owocowe', 'lekkie', 'bezalkoholowe']
ale = ['lekkie', 'owocowe', 'kwiatowe']

if(set(preferencja).issubset(set(pils))):
    print('Wybieram pils')
elif (set(preferencja).issubset(set(porter))):
    print('Wybieram porter')
elif (set(preferencja).issubset(set(stout))):
    print('Wybieram stout')
elif (set(preferencja).issubset(set(bezalkoholowe))):
    print('Wybieram bezalkoholowe')
elif (set(preferencja).issubset(set(ale))):
    print('Wybieram ale')
else:
    print('Nie ma tu piwa dla mnie, ide do innego pubu')
```

bröther may i have some lööps





**VATICAN confirms
secret WHILE(TRUE) LOOP
keeps the universe running**



While

while warunek:
instrukcja 1

Dopóki warunek jest TRUE,
wykonuje instrukcję wewnątrz pętli.

if warunek2:
instrukcja 2
break

Jeśli warunek przestanie być prawdziwy (FALSE),
zakończy się pętla.

instrukcja 3

Aby przerwać pętlę w określonym wypadku,
można użyć polecenia break.



wiek = 20

while wiek < 18:

 print("Nie możesz kupić piwa :(Wróć jak dorosniesz")

 wiek=int(input("Podaj swój wiek:"))

 print("Możesz kupić piwo!")

Zadanie 2: WHILE

```
puby = {  
    'Pub1': ['zywiec', 'perla', 'konska dawka'],  
    'Pub2': ['gaska beerbika', 'specjal'],  
    'Pub3': ['balios', 'orkiszowe z czosnkiem', 'uchod'],  
    'Pub4': ['zywiec', 'fabrykanckie swinie']  
}  
  
moje_piwo = 'uchod'  
  
nazwy_pub = list(puby.keys())  
  
znalezione = False  
index = 0  
  
while index < len(nazwy_pub) and not znalezione:  
    aktpub = nazwy_pub[index]  
    if moje_piwo in puby[aktpub]:  
        znalezione = True  
        print(f"Pije '{moje_piwo}' w {aktpub}")  
    index += 1  
  
if not znalezione:  
    print(f"Nie ma '{moje_piwo}' w zadnym pubie.")
```



I wrote a while(true) loop



With break
statement right?



I wrote a while(true) loop



With break
statement right?



FOR loop

```
for x in range(0, 5, 1):  
    wykonaj warunek1
```

```
found_obj = None  
for obj in objects:  
    if obj.key == search_key:  
        found_obj = obj  
        break  
else:  
    warunek2  
    print('No object found.')
```



```
for wiek in range (18):  
    print(f"Masz {wiek} lat.")  
    print("Nie możesz kupić piwa.")
```



Here's A Simple Loop With Python



Funkcje

```
def nazwa_funkcji(arg1, arg2):  
    instrukcje  
    ...  
return obiekt
```

```
output = nazwa_funkcji(a,b)  
print(output)
```



```
def sprawdz_wiek(wiek):  
    if wiek >= 18:  
        print("Możesz kupić piwo!")  
    else:  
        print("Nie możesz kupić piwa :(")
```

```
sprawdz_wiek(20)  
sprawdz_wiek(15)
```



Zadanie 4: FUNKCJE

```
napoje = {
    'Cola': ['gazowane', 'kofeina', '0'],
    'sok': ['niegazowane', 'owocowy', '0'],
    'Piwo1': ['gazowane', 'ciemne', '9'],
    'Piwo2': ['gazowane', 'jasne', '5']
}

jedzenie = {
    'Przysnacki': ['chrupkie', 'ostre', 'cebula'],
    'Lays': ['chrupkie', 'słone'],
    'Popcorn': ['masło', 'słone'],
    'Kebab1': ['cebula', 'falafel', 'ostry sos'],
    'Kebab2': ['salata', 'frytki', 'mięso', 'ostry sos']
}

def znajdz_menu(preferencje):
    pref_picia, pref_jedzenie = preferencje
    wybrane_picie, wybrane_jedzenie = None, None

    # Znajdz napoj
    for napoj, cechy in napoje.items():
        if all(pref in cechy for pref in pref_picia.split(' i ')):
            wybrane_picie = napoj
            break

    # Znajdz jedzenie
    for przekaska, cechy in jedzenie.items():
        if all(pref in cechy for pref in pref_jedzenie.split(' i ')):
            wybrane_jedzenie = przekaska
            break

    return wybrane_picie, wybrane_jedzenie

preferencje = ['gazowane i ciemne', 'mięso i ostry sos']

wynap, wybjad = znajdz_menu(preferencje)
print(f"Wybrany napój: {wynap}")
print(f"Wybrane jedzenie: {wybjad}")
```

Zasięg zmiennej

```
def funkcja():
    lokalna = 1
    return lokalna
```

```
print("zmienna lokalna =", lokalna)
```



Wszystkie zmienne, którym przypisane są wartości wewnątrz funkcji, są domyślnie lokalne (o ile nie zostaną specjalnie zadeklarowane jako globalne).