

Zamiast wpisywać komendy do shell'a możemy stworzyć skrypt, który wykona je za nas:

Stwórz folder o nazwie skrypty w wygodnym miejscu

Stwórz plik o nazwie skrypt.sh

Wywołaj komendę **ls -l**, zobaczysz coś takiego:

“-rw-rw-r–” - pierwsze litery r i w oznaczają odpowiednio “read” i “write” i oznaczają uprawnienia jakie aktualny użytkownik ma do tego pliku, żeby wykonywać skrypt musimy dodać uprawnienie “execute”:

chmod +x skrypt.sh

Wykonaj to polecenie i sprawdź czy uprawnienia się zmieniły.

Użyj dowolnego edytora tekstu aby zacząć edycję naszego pliku (np. vim, nano)

Wpisz następujący kod:

```
#!/bin/bash  
echo “cześć”
```

Zapisz plik i wykonaj go wpisując:

```
./skrypt.sh
```

Aby wczytać zmienną należy napisać:

```
read name
```

i następnie wydrukować ją:

```
echo “$name”
```

Dodaj do programu ‘prompt’, czyli wyświetl tekst, który poprosi osobę wywołującą program o podanie imienia (komenda echo)

Możemy również zapisać do zmiennej wartość zwróconą przez inny program, np.:

```
user=$(whoami)
```

Wydrukuj ją na ekran, co widzisz?

Użyj powyższej komendy, ale zamiast polecenia **whoami** użyj **date**

Teraz użyj **date +%H_%M_%S**, co się zmieniło?

Wpisz do linii poleceń linijkę:

```
mkdir folder_$(date +%H_%M_%S) - jak to działa? Sprawdź używając polecenia ls
```

W bash-u możemy korzystać ze znanych z innych języków programowania konstrukcji takich jak **if**, **for**, **while**:

Konstrukcja if:

```
if [warunki]  
then
```

```

        kod
fi

Np:
if ((2 < 1))
    then
        echo "czy się wykonam?"
    else
        echo "2 nie jest mniejsze od 1!"
fi

```

Przetestuj powyższy program.

Napisz program, który wczytuje zmienną z klawiatury i sprawdza czy jest większa, mniejsza czy równa 0, napisz na ekran wynik tej rozkminy.

Jeśli chcemy wykonać nasz program z podanym argumentem (np. wywołanie **ls -l** jest wywołaniem programu **ls** z argumentem **-l**) możemy użyć następującej konstrukcji:

```

#!/bin/bash
zmienna1 = $1
echo "$1"

```

Nazwijmy ten skrypt **arg.sh** i wywołajmy:

```

./arg.sh coś

```

Wiedząc, że pierwsza linijka naszego skryptu definiuje program (interpreter), który wykona nasz skrypt, stwórz analogiczny do dotychczasowych plik, który będzie prostym skryptem w pythonie, np. Hello world.

Przydatne komendy:

- mkdir
- touch
- ls
- cd
- pwd
- echo

Zadania do wykonania:

1. Napisz program, który sprawdza czy wpisana z klawiatury nazwa użytkownika zgadza się z nazwą aktualnego użytkownika (przyda się `whoami`)
2. Napisz program "dziennik", który zapyta użytkownika o wpis i doda go razem z aktualną godziną i datą do pliku "diary.txt"
3. Napisz program, który jako argumenty przyjmuje współczynniki funkcji kwadratowej i drukuje na ekran jej miejsca zerowe