

Bash skrypty

Polecenia BASHA można łączyć ze sobą tworząc skrypty, które dodatkowo mogą zawierać np. wyrażenia logiczne oraz odwoływać się do innych aplikacji.

Jako zwykły użytkownik wykonaj w konsoli:

```
touch skrypt
chmod 755 skrypt
```

Następnie otwórz plik "skrypt" w edytorze (nano, pico, jakiś ze środowiska graficznego). I wpisz następującą zawartość:

```
echo 'To jest tekst'
echo "A to drugi"
zmienna="cześć"
echo "Wartość zmiennej to: $zmienna"
echo 'Pojedynczy cudzysłów i $zmienna'
```

Następnie wykonaj plik wpisując w konsoli **./plik** . Zobaczysz coś takiego:

```
To jest tekst
A to drugi
Wartość zmiennej to: cześć
Pojedynczy cudzysłów i $zmienna
```

Powyższy skrypt BASHa prezentuje podstawowe jego składniki. Polecenie **echo** wyświetla tekst, jeżeli jest on w podwójnym cudzysłowie - będzie parsowany przez powłokę (zmienne zostaną zamienione na ich wartości) jeżeli pojedyncze - tekst nie będzie parsowany przez powłokę. Zmienną deklarujemy:

zmienna = "wartość"

A wywołujemy za pomocą **\$zmienna** (dodajemy znak dolara). Skrypt kończy wywołanie zmiennej w tekście z podwójnymi i pojedynczymi cudzysłowami. W przypadku tych drugich "\$zmienna" nie zostanie zamieniona na wartość zmiennej. Zmienną można też wyświetlić bezpośrednio: **echo \$zmienna**

Pobieranie danych od użytkownika

Chcemy skrypt, który poprosi użytkownika o dane, np:

```
echo 'Jak masz na imię?'
read zmienna
echo "Twoje imie to: $zmienna"
```

Wykonanie skryptu spowoduje wyświetlenie tekstu "Jak masz na imię?", a użytkownik będzie mógł wprowadzić dane, np. jeżeli wpisujemy "Piotr" to skrypt odpowie "Twoje imie to: Piotr". Skorzystaliśmy tutaj komendy **read** (read nazwa_zmiennej). Komenda ta czeka aż użytkownik wpisze coś i wciśnie enter. To co wpisał przypisane zostanie zmiennej o podanej nazwie.

Skrypt przykładowy 1 - Generator wizytówki - coś ala BASHowa księga gości

Chcemy prosty skrypt BASHA, który pobierze dane takie jak Imię i Nazwisko, adres Email i numer GG i zapisze do pliku. Oto kod

```
#!/bin/bash
echo 'Witaj w generatorze wizytówek v. 13.666'
echo '#####'
echo 'Program poprosi cię o wpisanie różnych danych.'
echo 'Po wpisaniu wciśnij Enter by przejść dalej.'
echo '#####'
echo 'Podaj swoje Imię i Nazwisko:'
read name
echo 'Podaj swój adres E-Mail:'
read mail
echo 'Podaj numer Gadu-Gadu:'
read gg
echo '#####'
echo 'Pobieranie danych zakończone'
touch wizytowka.txt
echo "$name" >> wizytowka.txt
echo "Email: $mail" >> wizytowka.txt
echo "Gadu-Gadu $gg" >> wizytowka.txt
echo 'Generowanie wizytówki zakończone'
echo 'Plik wizytowka.txt gotowy!'
echo '#####'
```

Wykonanie skryptu będzie wyglądać np. tak:

```
Witaj w generatorze wizytówek v. 13.666
#####
Program poprosi cię o wpisanie różnych danych.
Po wpisaniu wciśnij Enter by przejść dalej.
#####
Podaj swoje Imię i Nazwisko:
Piotr M
Podaj swój adres E-Mail:
tajny@email.pl
Podaj numer Gadu-Gadu:
123456
#####
Pobieranie danych zakończone
Generowanie wizytówki zakończone
Plik wizytowka.txt gotowy!
#####
```

I stworzy plik wizytowka.txt o zawartości:

```
Piotr M
Email: tajny@email.pl
Gadu-Gadu 123456
```

Przy okazji >> dopisuje dane do pliku więc wielokrotne wywołanie skryptu nie nadpisze danych...

Parametry wywołania

Wiele aplikacji przyjmuje parametry, np. **ls -a**. Bardzo prosto parametry te "przechwycić".

```
#!/bin/bash
echo "Nazwa skryptu to: $0"
echo '-----'
echo "Nazwa pierwszego argumentu to: $1"
echo '-----'
echo "Wszystkie argumenty: $*"
echo '-----'
```

```
echo "Podałeś $# poleceń"
```

Wywołanie ./a2.sh polecenie1 polecenie2 polecenie3 wyświetli:

```
Nazwa skryptu to: ./a2.sh
```

```
-----
```

```
Nazwa pierwszego argumentu to: polecenie1
```

```
-----
```

```
Wszystkie argumenty: polecenie1 polecenie2 polecenie3
```

```
-----
```

```
Podałeś 3 poleceń
```

\$0 - nazwa skryptu, **\$n** - wartość n-tego parametru, **\$*** - wszystkie parametry, **\$#** - liczba poleceń.