

Pętla for

Pętla for

Wykonuje polecenia zawarte wewnątrz pętli, na każdym składniku listy (iteracja).

Składnia:

```
for zmienna in lista
do
    polecenie
done
```

Przykład:

```
for x in jeden dwa trzy
do
    echo "To jest $x"
done
```

Zmiennej **x** przypisana jest lista, która składa się z trzech elementów: **jeden, dwa, trzy**. Wartością zmiennej **x** staje się po kolei każdy element listy, na wszystkich wykonywane jest polecenie: **echo "To jest \$x"**. Pętla for jest bardzo przydatna w sytuacjach, gdy chcemy wykonać jakąś operację na wszystkich plikach w danym katalogu. Na przykład chcemy uzyskać listę wszystkich plików o danym rozszerzeniu znajdujących się w jakimś katalogu, robimy to tak:

```
#!/bin/bash
for x in *html
do
    echo "To jest plik $x"
done
```

lub jeśli chcemy zmienić nazwy plików pisane *DUŻYMI* literami na nazwy pisane *małymi* literami:

```
#!/bin/bash
for nazwa in *
do
    mv $nazwa `echo $nazwa | tr ' [A-Z]' '[a-z]'`
done
```

Za zmianę *DUŻYCH* liter na *małe* (i na odwrót) odpowiedzialne jest polecenie **tr**.

Pętla select

Pętla select

Wygeneruje z listy słów po **in** proste ponumerowane menu, każdej pozycji odpowiada kolejna liczba od 1 wzwyż. Poniżej menu znajduje się znak zachęty PS3 gdzie wpisujemy cyfrę odpowiadającą wybranej przez nas pozycji w menu. Jeśli nic nie wpisujemy i wciśniemy ENTER, menu będzie wyświetlone ponownie. To co wpisaliśmy zachowywane jest w zmiennej **REPLY**. Gdy odczytane zostaje **EOF** (ang. *End Of File*) czyli znak końca pliku(**CTRL+D**) to select kończy pracę. Pętla działa dotąd dopóki nie wykonane zostanie polecenie **break** lub **return**.

Składnia:

```
select zmienna in lista
do
    polecenie
done
```

Od razu nasuwa się możliwość zastosowania wewnątrz niej instrukcji case:

```
#!/bin/bash
echo "Co wybierasz?"
select y in X Y Z Quit
do
    case $y in
        "X") echo "Wybrałeś X" ;;
        "Y") echo "Wybrałeś Y" ;;
        "Z") echo "Wybrałeś Z" ;;
        "Quit") exit ;;
        *) echo "Nic nie wybrałeś"
    esac
break
done
```

Najpierw zobaczymy proste ponumerowane menu, składające się z czterech elementów: **X**, **Y**, **Z** i **Quit**, teraz wystarczy tylko wpisać numer inetersującej nas opcji, a resztę zrobi instrukcja case. Polecenie **break**, które znajduje się w przedostatniej linii skryptu, kończy pracę pętli.

A teraz bardziej praktyczny przykład, poniższy skrypt (**Uwaga!**) przeznaczony dla dystrybucji **Slackware** wygeneruje menu składające się z listy Window Mangerów, po wybraniu konkretnej pozycji uruchomiony zostanie dany WM. Oczywiście należy skrypt zmodyfikować pod kątem własnego systemu. Jeśli komuś odpowiada takie rozwiązanie, wystarczy utworzyć alias: **alias startx="~/ten_skrypt"** i po ponownym zalogowaniu mamy po wpisaniu polecenia **startx** menu wyboru Window Managerów.

```
#!/bin/bash
echo ""
echo "[ JAKI WINDOW MANAGER URUCHOMIĆ? WYBIERZ CYFRĘ Z LISTY: ]"
echo ""
select l in BLACKBOX ENLIGHTENMENT GNOME ICEWM KDE MWM OPENWIN TWM WMAKER WYJŚCIE
do
    case "$l" in
        "BLACKBOX") cat /etc/X11/xinit/xinitrc.blackbox > ~/.xinitrc; startx $@ ;;
        "ENLIGHTENMENT") cat /etc/X11/xinit/xinitrc.e > ~/.xinitrc; startx $@ ;;
    esac
done
```

```

"GNOME") cat /etc/X11/xinit/xinitrc.gnome > ~/.xinitrc; startx $@ ;;
"ICEWM") cat /etc/X11/xinit/xinitrc.icewm > ~/.xinitrc; startx $@ ;;
"KDE") cat /etc/X11/xinit/xinitrc.kde > ~/.xinitrc; startx $@ ;;
"MWM") cat /etc/X11/xinit/xinitrc.mwm > ~/.xinitrc; startx $@ ;;
"OPENWIN") cat /etc/X11/xinit/xinitrc.openwin > ~/.xinitrc; startx $@ ;;
"TWM") cat /etc/X11/xinit/xinitrc.twm > ~/.xinitrc; startx $@ ;;
"WMAKER") cat /etc/X11/xinit/xinitrc.wmaker > ~/.xinitrc; startx $@ ;;
"WYJŚCIE") exit ;;
*) startx $@
esac
break
done

```

Elementy składowe listy w pętli **select**, noszą takie same nazwy jak wzorce w instrukcji **case** co umożliwia wykonanie skoku do danego wzorca i wykonania poleceń jemu przypisanych. Jak to wygląda w praktyce? Na przykład chcemy uruchomić **KDE**, wybieramy więc z menu opcje o wyżej wymienionej nazwie, następnie polecenie **cat** nadpisuje nasz domowy plik **.xinitrc**, kopiując do niego zawartość pliku **xinitrc** zoptymalizowanego dla **KDE**, znajdującego się w katalogu: **/etc/X11/xinit/xinitrc.kde**, po czym wykonywane jest polecenie **startx**. Zmienna **\$@** to zmienna specjalna umożliwiająca przekazywanie do skryptu parametrów (**startx** to też skrypt powłoki), dzięki czemu możemy spokojnie stosować wszelkie parametry np. dwukrotne odpalenie **X-ów**: **startx -- :0** na pierwszej konsoli i **startx -- :1** na drugiej. Gdy nie wpisujemy żadnych parametrów **\$@** jest pusta. A co się stanie w przypadku gdy podczas wyboru Window Managera podamy większą cyfrę niż tą jaką ma ostatni element menu lub jakiś inny znak? Uruchomiony zostanie ten WM, który ostatnio odpalaliśmy.

Ten sam skrypt przeznaczony dla dystrybucji **Red Hat W** tym przypadku polecenie: **echo "exec window_manager"** nadpisuje plik **.Xclients** znajdujący się w naszym katalogu **home**. Z aliasem postępujemy analogicznie jak w powyższym przykładzie.

```

#!/bin/bash
echo ""
echo "[ JAKI WINDOW MANAGER URUCHOMIĆ? WYBIERZ CYFRĘ Z LISTY: ]"
echo ""
select l in BLACKBOX ENLIGHTENMENT GNOME ICEWM KDE MWM OPENWIN TWM WMAKER WYJŚCIE
do
    case "$l" in
        "BLACKBOX") echo "exec blackbox" > ~/.Xclients; startx $@ ;;
        "ENLIGHTENMENT") echo "exec enligtenment" > ~/.Xclients; startx $@ ;;
        "GNOME") echo "exec gnome-session" > ~/.Xclients; startx $@ ;;
        "ICEWM") echo "exec icewm" > ~/.Xclients; startx $@ ;;
        "KDE") echo "exec startkde" > ~/.Xclients; startx $@ ;;
        "MWM") echo "exec mwm" > ~/.Xclients; startx $@ ;;
        "OPENWIN") echo "exec openwin" > ~/.Xclients; startx $@ ;;
        "TWM") echo "exec twm" > ~/.Xclients; startx $@ ;;
        "WMAKER") echo "exec wmaker" > ~/.Xclients; startx $@ ;;
        "WYJŚCIE") exit ;;
        *) startx $@
    esac
done
break
done

```