

Chap V

Wydzielanie funkcji.

Czasami w skrypcie zdarza się, że kod wielokrotnie powtarza się. Aby uprościć jego użycie oraz poprawić czytelność kodu można użyć funkcji, są to małe „podskrypty” działające w ramach głównego skryptu.

Składnia funkcji:

```
function nazwa(){  
    Zestaw instrukcji;  
}
```

Funkcji używa się wywołując jej nazwę.

```
#!/bin/bash  
function licz()  
{  
    y=5  
    x=10  
    echo $(( $x + $y ))  
}
```

licz

Wywołanie funkcji w skrypcie, zostanie zwrócony wynik dodawania.

Jednak wpisywanie zmiennych na sztywno nie jest dobrym rozwiązaniem, ponieważ ogranicza elastyczność funkcji. Tu z pomocą przychodzi własność – funkcja może przyjmować zmienne przy wywołaniu identycznie tak samo jak cały skrypt.

```
#!/bin/bash  
function licz()  
{  
    y=$1  
    x=$2  
    echo $(( $x + $y ))  
}
```

licz 5 10

Funkcja może też korzystać z wartości globalnych nie zadeklarowanych w niej.

```
function licz()  
{  
    echo $(( $x + $y ))  
}
```

x=10

y=20

licz

Może też korzystać z lokalnych zadeklarowanych w niej, mimo, że mogą istnieć inne globalne o tej samej nazwie.

```
#!/bin/bash
x=10
y=20

function licz()
{
    local y=$1
    local x=$2
    echo $(( $x + $y ))
}
```

```
licz 5 10
```

```
echo $(( $x + $y ))
```

Jak widać zmienne globalne nie zostały zmienione.

Funkcja może też wywoływać samą siebie, jest to rekurencja. Poniżej kod silni.

```
#!/bin/bash
function silnia()
{
    local liczba=$1
    if [ $liczba -eq 0 ]; then
        wynik=1
    else
        let "nastepna = liczba - 1"
        silnia $nastepna
        let "wynik = $liczba * $wynik"
    fi

    return $wynik
}
```

```
silnia $1
echo $wynik
```