

Perl

Подпрограми



Дефиниране

Чрез служебната дума **sub**:

```
sub subname {           # subname - произволен валиден идентификатор.  
    statement_1;  
    statement_2;  
    statement_3;  
}
```

Пример:

```
sub get_info {  
    print 'Enter your user name: ' ;  
    chomp ($name=<>) ;  
    print "\nEnter your password: " ;  
    chomp ($pass=<>) ;  
    return ($name, $pass) ;  
}
```

Извикване

Чрез името на процедурата:

```
($user_name, $user_pass) = get_info();  
print "Hello $user_name, your pass is: $user_pass !";
```

Когато се извиква без параметри може и `&subname`, вместо празните скоби:

```
($user_name, $user_pass) = &get_info;
```

Връщани стойности

Процедурите в Perl винаги връщат стойност - или явно, чрез **return** :

```
sub get_info {  
    print 'Enter your user name: '  
    chomp ( $name=<>);  
    print "\nEnter your password: "  
    chomp ( $pass=<>);  
    return ( $name, $pass);  
}
```

или като стойност на последния *изчислен* израз:

```
$x=3; $y=4;  
print sum();  
sub sum {  
    $x+$y;  
}
```

Аргументи

Предават се чрез масив, в автоматичната променлива `@_`, която има *private* статут за всяка отделна функция.

Пример:

```
print sum(2,3);
# вариант 1:
sub sum {
    print "@_ \n";          # ако искаме да видим всички аргументи
    $args_num = @_ ;      # броят на получените аргументи - в случая: 2
    return $_[0]+ $_[1];
}
# вариант 2:
sub sum {
    ($x,$y)=@_;           # @_=(2,3), $x=2, $y=3
    return $x+$y;
}
# вариант 3:
sub sum {
    $x=shift;             # $x=2, @_=(3)
    $y=shift;             # $y=3, @_=( )
    return $x+$y;
}
```

Променлив брой аргументи

! Гъвкавост при използване на функции с променлив брой аргументи:

```
print sum(1,2);
print sum(2,3,4);
# Следната дефиниция на
sub sum {
    $sum=0;
    foreach $ele (@_) {
        sum+=$ele;
    };
    return $sum;
}
```

Глобални променливи

Глобални (за една функция) променливи:

```
$sum=256;
print sum(2,3,4);
print "\n$sum";
sub sum {
    $sum=0; # $sum е глобална променлива
    foreach $ele(@_){
        $sum+=$ele;
    };
    return $sum;
}
```

Локални променливи

Локални променливи:

Създават се чрез оператора **my**. Областта на видимост е в рамките на **най-вътрешния** обхващащ блок !

```
my $i=0;
for (my $i=1; $i<=3; $i++){
    print "i=$i\n";    # i=1, i=2, i=3
};
print "\ni=$i\n";    # i=0
```

Полу-локални променливи ?

Чрез `local` се създава динамично локално копие на глобална променлива, което е видимо в блока в който е декларирано, както и във всички вътрешни блокове.

```
$name='Original';
edno();
print "$name\n";
sub edno {
    local $name='Edno';
    print "$name\n";
    dve();
    print "$name\n";
}
sub dve {
    $name='Dve';
}
```