

برنامه نویسی به زبان

پرل

مرجع کامل

تالیف

مهندس هادی کیامرثی

تمام مثال های موجود در این کتاب با کامپیوتر تست شده اند تا از هر گونه خطا  
مبرا باشند با این حال ممکن است باز هم خطاهایی در آن وجود داشته باشد از  
کلیه خوانندگان این کتاب ، اساتید و دانشجویان محترم خواهشمندم برای مطلع  
کردن مولف از این خطا ها لطفا با ایمیل آدرس زیر تماس بگیرند

hadikiamarsi@gmail.com

لازم به ذکر است کلیه حقوق مادی و معنوی این اثر برای مولف محفوظ می  
باشد و هرگونه کپی برداری و استفاده از محتویات این کتاب به هر نوعی تحت  
پیگرد قانونی قرار می گیرد

# فصل چهاردهم

فادری  
پیک  
مدتی

در این فصل مطالب زیر را خواهید آموخت

مرجع ها در پرل

ایجاد مرجع ها

ایجاد مرجع برای مرجع ها

مرجع های چرخشی

مرجع ها در توابع

کیا مدتی

# مرجع ها در پرل

یک مرجع در زبان برنامه نویسی پرل ( perl ) یک نوع داده ای اسکالر می باشد که در واقع یک آدرس است برای یک متغیر . فرقی نمی کند مرجع برای یک اسکالر یا آرایه و یا هاش ساخته شده باشد در هر حالت خود مرجع یک نوع داده ای اسکالر می باشد . مرجع در زبان های برنامه نویسی دیگر مانند C و یا پاسکال ( pascal ) به نام اشاره گر خوانده می شود .

## ایجاد مرجع ها

ایجاد کردن مرجع ها در زبان برنامه نویسی پرل ( perl ) بسیار ساده است و تنها به علامت \ ( backslash ) نیاز دارد نحوه ایجاد مرجع ها در زیر نشان داده شده است .

```
$scalarref = \$foo;  
$arrayref = \@ARGV;  
$hashref = \%ENV;  
$coderef = &handler;  
$globref = \*foo;
```

بیاد داشته باشید برای یک اشاره گر فایل یا دایرکتوری نمی شود مرجع ایجاد کرد . در ضمن شما می توانید یک آرایه را مستقیما به یک مرجع منتسب نمایید مانند مثال زیر

```
$arrayref = [1, 2, ['a', 'b', 'c']];
```

و در حالتی مشابه یک هاش را به یک مرجع منتسب نمایید این حالت را در مثال زیر می بینید

```
$hashref = {  
  'Adam' => 'Eve',  
  'Clyde' => 'Bonnie',  
};
```

همچنین می توانید یک زیرتابع یا تابع را مستقیما به یک مرجع منتسب نمایید مانند مثال زیر

```
$coderef = sub { print "Boink!\n" };
```

## ایجاد مرجع برای مرجع ها

هر متغیر مرجع به یک آدرس در حافظه اشاره دارد حال اگر به این مرجع یک مرجع دیگر منتسب نماییم زبان برنامه نویسی پرل ( perl ) مقدار متغیری که مرجع اول به آن اشاره می کرد را بر می گرداند . برای ایجاد مرجع مرجع از علامت های \$ ، @ و % استفاده می گرد برای آشنایی بیشتر با این مفهوم به مثال زیر توجه نمایید

```
#!/usr/bin/perl

$var = 10;

# Now $r has reference to $var scalar.
$r = \$var;

# Print value available at the location stored in $r.
print "Value of $var is : ", $$r, "\n";

@var = (1, 2, 3);
# Now $r has reference to @var array.
$r = \@var;
# Print values available at the location stored in $r.
print "Value of @var is : ", @$r, "\n";

%var = ('key1' => 10, 'key2' => 20);
# Now $r has reference to %var hash.
$r = \%var;
# Print values available at the location stored in $r.
print "Value of %var is : ", %$r, "\n";
```

اجرای کد بالا نتیجه زیر را ظاهر خواهد نمود

```
Value of 10 is : 10
Value of 1 2 3 is : 123
Value of %var is : key220key110
```

اگر شما نمی دانید متغیر مرجعی که دارید از چه نوع است برای تعیین نوع متغیر مرجع می توانید از تابع `ref` استفاده نمایید . لازم به ذکر است که خروجی این تابع یکی از گزینه های آورده شده در زیر می باشد

```
SCALAR
ARRAY
HASH
CODE
GLOB
REF
```

در زیر مثالی برای شما آورده شده است

```
#!/usr/bin/perl

$var = 10;
$r = \$var;
print "Reference type in r : ", ref($r), "\n";

@var = (1, 2, 3);
$r = \@var;
print "Reference type in r : ", ref($r), "\n";

%var = ('key1' => 10, 'key2' => 20);
$r = \%var;
print "Reference type in r : ", ref($r), "\n";
```

اجرای کد بالا نتیجه زیر را ظاهر خواهد نمود

```
Reference type in r : SCALAR
Reference type in r : ARRAY
Reference type in r : HASH
```

## مرجع های چرخشی

مرجع های چرخشی زمانی ایجاد می شوند که دو تا متغیر مرجع به هم اشاره نمایند . لازم به ذکر است که در استفاده از مرجع های چرخشی محتاط باشید چرا که دیده شده سبب ایجاد اشکالات امنیتی در برنامه شده اند . برای آشنایی بیشتر با نحوه کاربرد مرجع های چرخشی به مثال زیر توجه نمایید

```
#!/usr/bin/perl

my $foo = 100;
$foo = \$foo;

print "Value of foo is : ", $$foo, "\n";
```

اجرای کد بالا نتیجه زیر را ظاهر خواهد نمود

```
Value of foo is : REF(0x9aae38)
```

## مرجع ها در توابع

در زبان برنامه نویسی پرل ( perl ) شما براحتی می توانید برای توابع نیز مرجع ایجاد نمایید . در واقع ایجاد مرجع برای تابع راهی دیگر برای صدا زدن یا همان اجرا کردن تابع می باشد . برای آشنایی با نحوه کاربرد ایجاد مرجع برای تابع به مثال زیر توجه نمایید

```
#!/usr/bin/perl

# Function definition
sub PrintHash {
    my (%hash) = @_;

    foreach $item (%hash) {
        print "Item : $item\n";
    }
}

%hash = ('name' => 'Tom', 'age' => 19);

# Create a reference to above function.
$cref = \&PrintHash;

# Function call using reference.
&$cref(%hash);
```

اجرای کد بالا نتیجه زیر را ظاهر خواهد نمود

```
Item : name
Item : Tom
Item : age
Item : 19
```