

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

قادی کین مدثری

فصل اول

در این فصل مطالب زیر را خواهید آموخت

برنامه نویسی در حالت محاوره ای

برنامه نویسی در حالت متنی

پسوند فایل های پرل

توضیحات در پرل

فضاهای خالی در پرل

علامت نقل قول تکی و جفتی در زبان پرل

رشته های چند خطی در پرل

چاپ کاراکترهای خاص

شناسه ها در زبان پرل

زبان برنامه نویسی پرل ساختار دستوری خود را از زبان های برنامه نویسی `awk`, `sed`, `C`, `Bourne Shell`, `Smalltalk`, `Lisp` قرض گرفته است . اگرچه اختلافاتی با این زبان ها دارد . در این فصل آشنایی مختصری با زبان برنامه نویسی پرل پیدا خواهید نمود .

یک برنامه پرل شامل خطوط دستوری مرتب و پشت سرهمی می باشد که از بالا به پایین اجرا می شوند . هر خط یک برنامه پرل به علامت سمی کالن (`;`) ختم می شود .

پرل بر خلاف زبان هایی مانند پایتون و فرترن یک زبان با فرم آزاد می باشد به این معنا که براحتی می توانید هر چه قدر دوست داشتید از فضای خالی و تو رفتگی یا همان حاشیه گذاری در دستورات خود استفاده نمایید .

اولین برنامه پرل

برنامه نویسی در حالت محاوره ای

شما می توانید مفسر پرل را با استفاده از پارامتر e در خط فرمان مستقیماً بکار بگیرید . به عبارتی شما می توانید در خط فرمان مستقیماً دستورات زبان پرل را بنویسید و نتیجه آن را ببینید . لطفاً به مثال زیر توجه نمایید

```
$perl -e 'print "Hello World\n"'
```

اجرای کد بالا نتیجه زیر را ظاهر خواهد نمود

```
Hello, world
```

برنامه نویسی در حالت متنی

شما می توانید دستورات زبان پرل را در یک فایل نوشته و آن را اجرا نمایید . برای مثال دستورات زیر را با استفاده از یک ویرایشگر فایل (در سیستم عامل های یونیکس بیس مانند لینوکس از ویرایشگرهای vi یا vim یا nano می توانید استفاده نمایید) در فایل hello.pl نوشته و ذخیره نمایید .

```
#!/usr/bin/perl
```

```
# This will print "Hello, World"  
print "Hello, world\n";
```

این قسمت از کد **/usr/bin/perl** مسیر فایل اجرایی پرل را نشان می دهد . حال برای اجرای کد موجود در فایل hello.pl ابتدا باید سطح دسترسی به فایل را به 0755 تغییر دهید سپس دستورات را اجرا نمایید برای اجرا دستورات زیر را در خط فرمان لینوکس اجرا نمایید .

```
$chmod 0755 hello.pl  
$./hello.pl
```

اجرای کد بالا نتیجه زیر را ظاهر خواهد نمود

```
Hello, world
```

برای اجرای فایل hello.pl در سیستم عامل ویندوز از دستور زیر استفاده می نمایم

```
$/hello.pl
```

اجرای کد بالا نتیجه زیر را ظاهر خواهد نمود

```
Hello, world
```

در پرل برای استفاده از توابع هم می توانید از پرانتز استفاده نمایید و هم می توانید بدون پرانتز آن ها را بکار ببرید. برای روشن شدن موضوع به مثال زیر توجه نمایید هر دو خط زیر نتیجه یکسانی را ایجاد می نمایند

```
print("Hello, world\n");  
print "Hello, world\n";
```

پسوند فایل های پرل

یک فایل که حاوی متن کدهای پرل می باشد می تواند توسط هر نرم افزار ویرایش متنی که در سیستم عامل شما موجود است ایجاد گردد فقط نکته حائز اهمیت در این قضیه این می باشد که پسوند این فایل باید حتما pl یا PL باشد. لازم به ذکر است که در نام فایل حاوی کدهای پرل نباید فاصله (space) بکار رفته باشد.

توضیحات در پرل

در هر زبان برنامه نویسی توضیحات موجود در کدها مهمترین دوستان برنامه نویسان می باشند چرا که باعث خوانا تر شدن کد برنامه برای برنامه نویس در مراجعات بعدی به آن کد می شوند.

در زبان برنامه نویسی پرل برای ایجاد توضیحات یک خطی باید علامت هاش (#) را در ابتدای آن خط قرار دهید. لطفا به مثال زیر توجه نمایید.

```
# This is a comment in perl
```

برای ایجاد توضیحات چند خطی در زبان برنامه نویسی پرل باید در ابتدای توضیح خود از علامت = استفاده نمایید سپس توضیح خود را تایپ نمایید و در نهایت در انتهای توضیح خود عبارت =cut را قرار دهید برای روشن شدن بیشتر این درس به مثال زیر توجه نمایید

```
#!/usr/bin/perl

# This is a single line comment
print "Hello, world\n";

=begin comment
This is all part of multiline comment.
You can use as many lines as you like
These comments will be ignored by the
compiler until the next =cut is encountered.
=cut
```

اجرای کد بالا نتیجه زیر را ظاهر خواهد نمود

```
Hello, world
```

فضاهای خالی در پرل

زبان پرل به تعداد فضاهای خالی حساس نمی باشد برای روشن شدن مطلب به مثال زیر توجه نمایید

```
#!/usr/bin/perl

print  "Hello, world\n";
```

اجرای کد بالا نتیجه زیر را ظاهر خواهد نمود

```
Hello, world
```

اما بیاد داشته باشید پرل فضاهای خالی بکار رفته در علامت نقل قول را در نظر می گیرد . برای روشن شدن مطلب به مثال زیر توجه نمایید .

```
#!/usr/bin/perl

# This would print with a line break in the middle
print "Hello
      world\n";
```

اجرای کد بالا نتیجه زیر را ظاهر خواهد نمود

```
Hello
      world
```

پرل انواع فضاهای خالی مانند spaces, tabs, newlines را در صورتی که خارج از علامت نقل قول باشد در نظر نمی گیرد ولی در صورتی که در داخل علامت نقل قول باشد آنها را در نظر می گیرد .

علامت نقل قول تکی و جفتی در زبان پرل

در زبان پرل شما می توانید برای تعریف یک رشته هم از علامت نقل قول تکی (single quotes) و هم نقل قول جفتی (double quotes) استفاده نمایید

```
#!/usr/bin/perl

print "Hello, world\n";
print 'Hello, world\n';
```

اجرای کد بالا نتیجه زیر را ظاهر خواهد نمود


```
Hello, world  
Hello, world\n
```

در اینجا یک تفاوت خیلی مهم در استفاده از علامت نقل قول تکی و نقل قول جفتی وجود دارد اگر نام متغیری در بین علامت نقل قول جفتی قرار گیرد مقدار متغیر مد نظر قرار می گیرد در صورتی که اگر متغیری در بین علامت نقل قول تکی قرار گیرد نام خود متغیر مد نظر قرار می گیرد . برای روشن شدن بیشتر این درس لطفا به مثال زیر توجه نمایید

```
#!/usr/bin/perl  
  
$a = 10;  
print "Value of a = $a\n";  
print 'Value of a = $a\n';
```

اجرای کد بالا نتیجه زیر را ظاهر خواهد نمود

```
Value of a = 10  
Value of a = $a\n
```

رشته های چند خطی در پرل

برای ایجاد رشته های چند خطی در زبان برنامه نویسی پرل باید در ابتدای رشته خود از عبارت "EOF" << یا 'EOF' << استفاده نمایید و سپس رشته خود را تایپ نمایید و در نهایت در انتهای رشته خود از عبارت EOF استفاده نمایید . لازم به ذکر است قوانینی که برای رشته با علامت نقل قول جفتی صادق می باشد برای عبارت "EOF" << نیز صادق است و همچنین قوانینی که برای رشته با علامت نقل قول تکی صادق می باشد برای عبارت 'EOF' << نیز صادق است برای روشن شدن بیشتر این درس لطفا به مثال زیر توجه نمایید

```
#!/usr/bin/perl
```

```
$a = 10;
$var = <<"EOF";
This is the syntax for here document and it will continue
until it encounters a EOF in the first line.
This is case of double quote so variable value will be
interpolated. For example value of a = $a
EOF
print "$var\n";

$var = <<'EOF';
This is case of single quote so variable value will be
interpolated. For example value of a = $a
EOF
print "$var\n";
```

اجرای کد بالا نتیجه زیر را ظاهر خواهد نمود

```
This is the syntax for here document and it will continue
until it encounters a EOF in the first line.
This is case of double quote so variable value will be
interpolated. For example value of a = 10

This is case of single quote so variable value will be
interpolated. For example value of a = $a
```

چاپ کاراکترهای خاص

در زبان برنامه نویسی پرل بعضی علامت ها معنی خاصی دارند که اگر تصمیم بگیرید آن ها را چاپ نمایید چاپ نمی شوند به عنوان مثال علامت \$ که در ابتدای هر نامی قرار بگیرد به این معناست که آن نام یک متغیر می باشد حال برای چاپ این علامت باید از کاراکتر backslash (\) در ابتدای این علامت استفاده نمایید برای روشن شدن بیشتر این موضوع لطفاً به مثال زیر توجه نمایید .

```
#!/usr/bin/perl

$result = "This is \"number\"";
print "$result\n";
print "\\$result\n";
```

اجرای کد بالا نتیجه زیر را ظاهر خواهد نمود

```
This is "number"  
$result
```

شناسه ها در زبان پرل

یک شناسه در زبان پرل در واقع یک نام می باشد که برای شناسایی متغیرها ، تابع ها ، کلاس ها ، ماژول ها و ... بکار می رود . یک شناسه برای متغیر یا به عبارتی نام یک متغیر در زبان پرل نمی تواند با \$ ، @ و % شروع شود ولی برای نام گذاری متغیرها می توان از حروف ، اعداد و خط فاصله استفاده کرد

زبان برنامه نویسی پرل یک زبان حساس به بزرگ و کوچکی حروف می باشد بر این اساس دو شناسه **\$Manpower** و **\$manpower** با هم تفاوت دارند