

perl

Posted March 25th, 2024

perl is an old school language.

1. mod_perl

2. variables

```
my $name = "value";
my %hash = ( "blah"=>"foo",
"base"=>"path");
print $hash{"blah"};
```

```
my @array = ("a","b","c");
print $array[1];
```

3. comparators:

Numerical operators: <, >, <=, >=, ==, !=, <=>, +, *

String operators: lt, gt, le, ge, eq, ne, cmp, ., x

4.

```
unless ($foo > 3)
{
    print "foo";
}
else
{
    print "blah";
}
```

5. if, elsif, else

\$foo = 10 if (\$blah == 4);

6. ?:

```
$blah = ($foo==1) ? "one":"two";
```

```
7. for (my $i=0;$i<10;$i++)
{
}
foreach my $blah (@reverse)
{
    print $blah."\n";
}
```

8. keys %hash - to get keys of hash

9. next, last can be used in loops..
equivalent to continue, and break

10. pop,
push,
shift,
unshift
splice
join
grep

```
11. my $rc = system "perl", "foo.pl";
$rc >>= 8;
print "Status Code:".$rc."\n";
```

```
my @lines = `perl foo.pl`;
```

```
exit 37;
```

```

12. files
my $filename= "text.txt";
my $result = open my $handle, "<", $filename;

if (!$result)
{
die "Failed to open ".$filename." because, $!;
}

open (my $handle, "<", $filename) || die "failed ".$!."\n";

```

```

13. while (!eof $handle)
{
my $line = readline $fh;
}

```

```

14. while (my $line = readline $handle)
{
}

```

```

while (my $line = <$handle>)
{
}

```

```

while (<$handle>)
{
print $_;
}

```

```

15. -e
-d
-f

```

16. regex

```

if ( $hello =~ m/(\w+)\s+(\w)/)
echo $1." ".$2;

```

```

replace $foo =~ s/blah/foo/g;
// g is global

```

17. modules.

- 1) must return 1 in .pl file
- 2) require "Demo::module_name";
(comes from subdir Demo/module_name.pl)
- 3) PERL5LIBS=blah