

Sheeple - Examples

Subset 0

examples/0/hello_world.sh	examples/0/hello_world.pl
<pre>#!/bin/dash echo hello world</pre>	<pre>#!/usr/bin/perl -w print "hello world\n";</pre>
examples/0/ls-l.sh	examples/0/ls-l.pl
<pre>#!/bin/dash ls -l /dev/null</pre>	<pre>#!/usr/bin/perl -w system "ls -l /dev/null";</pre>
examples/0/ls.sh	examples/0/ls.pl
<pre>#!/bin/dash ls /dev/null</pre>	<pre>#!/usr/bin/perl -w system "ls /dev/null";</pre>
examples/0/pwd.sh	examples/0/pwd.pl
<pre>#!/bin/dash pwd</pre>	<pre>#!/usr/bin/perl -w system "pwd";</pre>
examples/0/single.sh	examples/0/single.pl
<pre>#!/bin/dash pwd ls id date</pre>	<pre>#!/usr/bin/perl -w system "pwd"; system "ls"; system "id"; system "date";</pre>
examples/0/truth.sh	examples/0/truth.pl
<pre>#!/bin/dash echo When old age shall this generation waste, echo Thou shalt remain, in midst of other woe echo Than ours, a friend to man, to whom thou sayst, echo Beauty is truth, truth beauty, - that is all echo Ye know on earth, and all ye need to know.</pre>	<pre>#!/usr/bin/perl -w print "When old age shall this generation waste,\n"; print "Thou shalt remain, in midst of other woe\n"; print "Than ours, a friend to man, to whom thou sayst,\n"; print "Beauty is truth, truth beauty, - that is all\n"; print "Ye know on earth, and all ye need to know.\n";</pre>
examples/0/truth0.sh	examples/0/truth0.pl

```
#!/bin/dash
echo When old age shall this generation waste,
echo Thou shalt remain, in midst of other woe
echo Than ours, a friend to man, to whom thou sayst,
echo Beauty is truth, truth beauty, - that is all
echo Ye know on earth, and all ye need to know.
```

```
#!/usr/bin/perl -w
print "When old age shall this generation waste,\n";
print "Thou shalt remain, in midst of other woe\n";
print "Than ours, a friend to man, to whom thou
sayst,\n";
print "Beauty is truth, truth beauty, - that is
all\n";
print "Ye know on earth, and all ye need to know.\n";
```

[examples/0/variables.sh](#)
[examples/0/variables.pl](#)

```
#!/bin/dash
a=hello
b=world
echo $a $b
```

```
#!/usr/bin/perl -w
$a = 'hello';
$b = 'world';
print "$a $b\n";
```

[examples/0/variables0.sh](#)
[examples/0/variables0.pl](#)

```
#!/bin/dash
a=hello
b=world
echo $a $b
```

```
#!/usr/bin/perl -w
$a = 'hello';
$b = 'world';
print "$a $b\n";
```

Subset 1

[examples/1/cd.sh](#)
[examples/1/cd.pl](#)

```
#!/bin/dash
cd /tmp
pwd
```

```
#!/usr/bin/perl -w
chdir '/tmp';
system "pwd";
```

[examples/1/for.sh](#)
[examples/1/for.pl](#)

```
#!/bin/dash

for word in Houston 1202 alarm
do
    echo $word
done
```

```
#!/usr/bin/perl -w

foreach $word ('Houston', 1202, 'alarm') {
    print "$word\n";
}
```

[examples/1/for_exit.sh](#)
[examples/1/for_exit.pl](#)

```
#!/bin/dash

for word in Houston 1202 alarm
do
    echo $word
    exit 0
done
```

```
#!/usr/bin/perl -w

foreach $word ('Houston', 1202, 'alarm') {
    print "$word\n";
    exit 0;
}
```

[examples/1/for_gcc.sh](#)

```
#!/bin/dash
for c_file in *.c
do
    echo gcc -c $c_file
done
```

[examples/1/for_gcc.pl](#)

```
#!/usr/bin/perl -w
foreach $c_file (glob("*.c")) {
    print "gcc -c $c_file\n";
}
```

[examples/1/for_read.sh](#)

```
#!/bin/dash

for n in one two three
do
    read line
    echo Line $n $line
done
```

[examples/1/for_read.pl](#)

```
#!/usr/bin/perl -w

foreach $n ('one', 'two', 'three') {
    $line = <STDIN>;
    chomp $line;
    print "Line $n $line\n";
}
```

[examples/1/for_read0.sh](#)

```
#!/bin/dash

for n in one two three
do
    read line
    echo Line $n $line
done
```

[examples/1/for_read0.pl](#)

```
#!/usr/bin/perl -w

foreach $n ('one', 'two', 'three') {
    $line = <STDIN>;
    chomp $line;
    print "Line $n $line\n";
}
```

Subset 2

[examples/2/args.sh](#)

```
#!/bin/dash

echo My first argument is $1
echo My second argument is $2
echo My third argument is $3
echo My fourth argument is $4
echo My fifth argument is $5
```

[examples/2/args.pl](#)

```
#!/usr/bin/perl -w

print "My first argument is $ARGV[0]\n";
print "My second argument is $ARGV[1]\n";
print "My third argument is $ARGV[2]\n";
print "My fourth argument is $ARGV[3]\n";
print "My fifth argument is $ARGV[4]\n";
```

[examples/2/elif.sh](#)

```
#!/bin/dash
if test Andrew = great
then
    echo correct
elif test Andrew = fantastic
then
    echo yes
else
    echo error
fi
```

[examples/2/elif.pl](#)

```
#!/usr/bin/perl -w
if ('Andrew' eq 'great') {
    print "correct\n";
} elsif ('Andrew' eq 'fantastic') {
    print "yes\n";
} else {
    print "error\n";
}
```

examples/2/if.sh	examples/2/if.pl
<pre>#!/bin/dash if test Andrew = great then echo correct else echo error fi</pre>	<pre>#!/usr/bin/perl -w if ('Andrew' eq 'great') { print "correct\n"; } else { print "error\n"; }</pre>
examples/2/single_quotes.sh	examples/2/single_quotes.pl
<pre>#!/bin/dash echo 'hello world'</pre>	<pre>#!/usr/bin/perl -w print "hello world\n";</pre>
examples/2/truth.sh	examples/2/truth.pl
<pre>#!/bin/dash echo 'When old age shall this generation waste,' echo 'Thou shalt remain, in midst of other woe' echo 'Than ours, a friend to man, to whom thou sayst,' echo '"Beauty is truth, truth beauty", - that is all' echo 'Ye know on earth, and all ye need to know.'</pre>	<pre>#!/usr/bin/perl -w print "When old age shall this generation waste,\n"; print "Thou shalt remain, in midst of other woe\n"; print "Than ours, a friend to man, to whom thou sayst,\n"; print "\"Beauty is truth, truth beauty\", - that is all\n"; print "Ye know on earth, and all ye need to know.\n";</pre>
examples/2/truth2.sh	examples/2/truth2.pl
<pre>#!/bin/dash echo 'When old age shall this generation waste,' echo 'Thou shalt remain, in midst of other woe' echo 'Than ours, a friend to man, to whom thou sayst,' echo '"Beauty is truth, truth beauty", - that is all' echo 'Ye know on earth, and all ye need to know.'</pre>	<pre>#!/usr/bin/perl -w print "When old age shall this generation waste,\n"; print "Thou shalt remain, in midst of other woe\n"; print "Than ours, a friend to man, to whom thou sayst,\n"; print "\"Beauty is truth, truth beauty\", - that is all\n"; print "Ye know on earth, and all ye need to know.\n";</pre>

Subset 3

examples/3/double_quotes.sh	examples/3/double_quotes.pl
<pre>#!/bin/dash echo "hello world"</pre>	<pre>#!/usr/bin/perl -w print "hello world\n";</pre>
examples/3/filetest-d.sh	examples/3/filetest-d.pl

```
#!/bin/dash
if [ -d /dev/null ]
then
    echo /dev/null
fi
if [ -d /dev ]
then
    echo /dev
fi
```

```
#!/usr/bin/perl -w
if (-d '/dev/null') {
    print "/dev/null\n";
}
if (-d '/dev') {
    print "/dev\n";
}
```

[examples/3/filetest-r.sh](#)

```
#!/bin/dash
if test -r /dev/null
then
    echo a
fi
if test -r nonexistentfile
then
    echo b
fi
```

[examples/3/filetest-r.pl](#)

```
#!/usr/bin/perl -w
if (-r '/dev/null') {
    print "a\n";
}
if (-r 'nonexistentfile') {
    print "b\n";
}
```

[examples/3/filetest0.sh](#)

```
#!/bin/dash
if test -r /dev/null
then
    echo a
fi
if test -r nonexistentfile
then
    echo b
fi
```

[examples/3/filetest0.pl](#)

```
#!/usr/bin/perl -w
if (-r '/dev/null') {
    print "a\n";
}
if (-r 'nonexistentfile') {
    print "b\n";
}
```

[examples/3/filetest1.sh](#)

```
#!/bin/dash
if [ -d /dev/null ]
then
    echo /dev/null
fi
if [ -d /dev ]
then
    echo /dev
fi
```

[examples/3/filetest1.pl](#)

```
#!/usr/bin/perl -w
if (-d '/dev/null') {
    print "/dev/null\n";
}
if (-d '/dev') {
    print "/dev\n";
}
```

[examples/3/l.sh](#)[examples/3/l.pl](#)

```
#!/bin/dash
# l [file|directories...] - list files
# written by andrewt@cse.unsw.edu.au as a COMP2041
example

ls -las "$@"
exit 0
```

```
#!/usr/bin/perl -w
# l [file|directories...] - list files
# written by andrewt@cse.unsw.edu.au as a COMP2041
example

system "ls -las @ARGV";
exit 0;
```

[examples/3/sequence0.sh](#)
[examples/3/sequence0.pl](#)

```
#!/bin/dash
# print a contiguous integer sequence
start=$1
finish=$2

number=$start
while test $number -le $finish
do
    echo $number
    number=`expr $number + 1` # increment number
done
```

```
#!/usr/bin/perl -w
# print a contiguous integer sequence
$start = $ARGV[0];
$finish = $ARGV[1];

$number = $start;
while ($number <= $finish) {
    print "$number\n";
    $number = $number + 1; # increment number
}
```

[examples/3/sequence_expr.sh](#)
[examples/3/sequence_expr.pl](#)

```
#!/bin/dash
# print a contiguous integer sequence
start=$1
finish=$2

number=$start
while test $number -le $finish
do
    echo $number
    number=`expr $number + 1` # increment number
done
```

```
#!/usr/bin/perl -w
# print a contiguous integer sequence
$start = $ARGV[0];
$finish = $ARGV[1];

$number = $start;
while ($number <= $finish) {
    print "$number\n";
    $number = $number + 1; # increment number
}
```

Subset 4

[examples/4/filetest-d.sh](#)
[examples/4/filetest-d.pl](#)

```
#!/bin/dash
if test -d /dev/null
then
    echo /dev/null
fi
if test -d /dev
then
    echo /dev
fi
```

```
#!/usr/bin/perl -w
if (-d '/dev/null') {
    print "/dev/null\n";
}
if (-d '/dev') {
    print "/dev\n";
}
```

[examples/4/filetest2.sh](#)
[examples/4/filetest2.pl](#)

```
#!/bin/dash
if test -d /dev/null
then
    echo /dev/null
fi
if test -d /dev
then
    echo /dev
fi
```

```
#!/usr/bin/perl -w
if (-d '/dev/null') {
    print "/dev/null\n";
}
if (-d '/dev') {
    print "/dev\n";
}
```

[examples/4/primes.sh](#)

```
#!/bin/dash

is_prime() {
    local n i
    n=$1
    i=2
    while test $i -lt $n
    do
        test $((n % i)) -eq 0 && return 1
        i=$((i + 1))
    done
    return 0
}

i=0
while test $i -lt 1000
do
    is_prime $i && echo $i
    i=$((i + 1))
done
```

[examples/4/primes.pl](#)

```
#!/usr/bin/perl -w

sub is_prime {
    my ($n, $i);
    $n = $_[0];
    $i = 2;
    while ($i < $n) {
        $n % $i == 0 and return 1;
        $i = $i + 1;
    }
    return 0;
}

$i = 0;
while ($i < 1000) {
    is_prime $i or print "$i\n";
    $i = $i + 1;
}
```

[examples/4/sequence1.sh](#)

```
#!/bin/dash
# print a contiguous integer sequence
start=$1
finish=$2

number=$start
while test $number -le $finish
do
    echo $number
    number=$((number + 1)) # increment number
done
```

[examples/4/sequence1.pl](#)

```
#!/usr/bin/perl -w
# print a contiguous integer sequence
$start = $ARGV[0];
$finish = $ARGV[1];

$number = $start;
while ($number <= $finish) {
    print "$number\n";
    $number = $number + 1; # increment number
}
```

[examples/4/sequence_math.sh](#)
[examples/4/sequence_math.pl](#)

```
#!/bin/dash
# print a contiguous integer sequence
start=$1
finish=$2

number=$start
while test $number -le $finish
do
    echo $number
    number=$((number + 1)) # increment number
done
```

```
#!/usr/bin/perl -w
# print a contiguous integer sequence
$start = $ARGV[0];
$finish = $ARGV[1];

$number = $start;
while ($number <= $finish) {
    print "$number\n";
    $number = $number + 1; # increment number
}
```

[examples/4/series.sh](#)

```
#!/bin/dash
start=13
if test $# -gt 0
then
    start=$1
fi
i=0
number=$start
file=./tmp.numbers
rm -f $file
while true
do
    if test -r $file
    then
        if fgrep -x -q $number $file
        then
            echo Terminating because series is
repeating
            exit 0
        fi
    fi
    echo $number >>$file
    echo $i $number
    k=`expr $number % 2`
    if test $k -eq 1
    then
        number=`expr 7 '*' $number + 3`
    else
        number=`expr $number / 2`
    fi
    i=`expr $i + 1`
    if test $number -gt 100000000 -o $number -lt
-100000000
    then
        echo Terminating because series has become
too large
        exit 0
    fi
done
rm -f $file
```

[examples/4/series.pl](#)

```
#!/usr/bin/perl -w
$start = 13;
if (@ARGV > 0) {
    $start = $ARGV[0];
}
$i = 0;
$number = $start;
$file = './tmp.numbers';
system "rm -f $file";
while (1) {
    if (-r $file) {
        if (! system "fgrep -x -q $number $file") {
            print "Terminating because series is
repeating\n";
            exit 0;
        }
    }
    open F, '>>', $file or die;
    print F "$number\n";
    close F;
    print "$i $number\n";
    $k = $number % 2;
    if ($k == 1) {
        $number = 7 * $number + 3;
    } else {
        $number = $number / 2;
    }
    $i = $i + 1;
    if ($number > 100000000 || $number < -100000000)
{
        print "Terminating because series has become
too large\n";
        exit 0;
    }
}
system "rm -f $file";
```