

```
#!/usr/bin/perl

# Auswertung der Parameteruebergabe
read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
@pairs = split(/&/, $buffer);
foreach $pair (@pairs)
{
    ($name, $value) = split(/=/, $pair);
    $value =~ tr/+// ;
    $value =~ s/%([a-fA-F0-9] [a-fA-f0-9])/pack("C",hex($1))/eg;
    $value =~ s/<!--(.|\n)*-->//g;
    $FORM{$name} = $value;
}

# HTML-Ausgabe vorbereiten
print "CONTENT-type: text/html\n\n";

# Parameter ausgeben
print "Sie haben ".$FORM{'name'}." eingegeben<br><br>";
print "<a href=\"../uebung4.html\">Zurueck zur Eingabe</a>"
```

```
#!/usr/bin/perl
print "CONTENT-type: text/html\n\n";

#HTML-Seite öffnen
open (neueseite,"../htdocs/zweiteseite.html");

#Seite auslesen
@seitentext = <neueseite>;

#HTML-Seite schliessen
close (neueseite);

#eingelezene HTML-Seite ausgeben
print @seitentext;

# ENDE Script
```

```

#!/usr/bin/perl

# Uebung 5: Anzeigen des Directory-Inhaltes

# Auswertung der Parameteruebergabe
read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
@pairs = split(/&/, $buffer);
foreach $pair (@pairs)
{
    ($name, $value) = split(/=/, $pair);
    $value =~ tr/+// ;
    $value =~ s/%([a-fA-F0-9][a-fA-f0-9])/pack("C",hex($1))/eg;
    $value =~ s/<!--(.\n)*-->//g;
    $FORM{$name} = $value;
}

# HTML-Ausgabe vorbereiten
print "CONTENT-type: text/html\n\n";

# Parameter ausgeben
$verzname = $FORM{'verzeichnis'};
print "Inhalt des Verzeichnis ".$verzname." :<br><br>";

# Directory öffnen
opendir(DIR_HANDLE,$verzname);

# Wiederholen solange neue Zeilen aus dem Directory
# gelesen werden können. Einlesen einer Directory-Zeile
while($inhalt = readdir(DIR_HANDLE))
    { print $inhalt."<br>";}

# Directory schliessen
closedir(DIR_HANDLE);

print "<br><br><a href=\"../uebung5.html\">Zurueck zur Eingabe</a>";

```

```

#!/usr/bin/perl

# Uebung 6: Suchen eines Musters in einer Datei

# Auswertung der Parameteruebergabe
read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
@pairs = split(/&/, $buffer);
foreach $pair (@pairs)
{
    ($name, $value) = split(/=/, $pair);
    $value =~ tr/+// ;
    $value =~ s/%([a-fA-F0-9][a-fA-f0-9])/pack("C",hex($1))/eg;
    $value =~ s/<!--(.|\n)*-->//g;
    $FORM{$name} = $value;
}

# HTML-Ausgabe vorbereiten
print "CONTENT-type: text/html\n\n";

# Parameter ausgeben
$dateiname = $FORM{'datei'};
$mustertext = $FORM{'muster'};
print "Folgende Muster \"\".$mustertext.\"\" "
print "wird in der Datei \"\".$dateiname.\"\" gesucht:<br>";

# Muster suchen
$ergebnis = 0;

# Datei mit den Datei-Handle LESEN zum auslesen öffnen
open(LESEN, "<\".$dateiname);

# Wiederholen solange eine Zeile vom Datei-Handle LESEN
# gelesen werden kann. Zeile in Variable $zeile einlesen
while(defined($zeile = <LESEN>))
{
    # Prüfen ob Inhalt der Variable $mustertext in
    # der aktuelle Zeile enthalten ist
    if ($zeile =~ /$mustertext/)
    { $ergebnis = $ergebnis + 1; }
}

# Datei schliessen
close(LESEN);

print "Der Mustertext '$mustertext.' wurde '$ergebnis.' mal
gefunden.<br>";

# Link zurück anzeigen
print "<br><br><a href=\"../uebung6.html\">Zurueck zur Eingabe</a>";

```

```
#!/usr/bin/perl

# Uebung 7: Mitloggen eines Dateizugriffs
# Datei für HTML-Zugriffe: /var/log/httpd/access_log

# HTML-Ausgabe vorbereiten
print "CONTENT-type: text/html\n\n";

# Logdatei für anhängenden Schreibzugriff öffnen
# Dazu Datei-Handle LOGDATEI benutzen
open (LOGDATEI,">>ben_log");

# Shell-Befehl nutzbar machen (hier Befehl tail)
use Shell qw(tail);

# Der Variable LASTLINE die letzte Zeile der Datei
# /var/log/httpd/access_log zuweisen
$LASTLINE = tail("--lines=1","/var/log/httpd/access_log");

# Wert von Variable LASTLINE in Logdatei schreiben
# Dabei als Ziel den Datei-Handle verwenden
print LOGDATEI $LASTLINE;

# Logdatei schliessen
close(LOGDATEI);
```