



C SHELL

* csh with prompt %. (* sh with prmt \$ *)

* In csh, .login .cshrc .logout. (* in sh, .profile *)

* Local variable

% set name = peter (* space "=" ?? *)

% unset name

(* in shell \$ name=paeter *)

% set name=(peter tom)

% echo name[1]

peter

% echo name

peter tom

% echo name[1-2]

peter tom

% echo \$#name

2

% set

...

Global variables

% setenv name peter

% unsetenv name

% printenv

...

(* similar to use export varialbe command in sh *)

(* will only affect its child, grandchild .. *)



* Variable modification.

```
% set file=/usr/snred/cwork/wood.c
% echo $file
/usr/snred/cwork/wood.c
% echo $file:h
/usr/snred/cwork/ (* contain header *)
% echo $file:r
/usr/snred/cwork/wood (* remove extension .*** *)
% echo $file:t
wood.c (* contain tail *)
$ echo $file:e
c (* the extension part *)
```

* Execution

```
$ csh ex (* method 1 *)
```

```
$ chmod ex u+x (* method 2 *)
$ ex
```

Put remark # at the very beginning of the csh program; otherwise, when use method 2 to execute a csh program, instead of executed by csh, sh will execute it and results in an error !!!

==> Put # in csh programs at the first line always !!
Do not put # in sh programs at the first line !!

* Parameters

```
$#argv      (* $# in sh *)   or $#
$argv[1]    (* $1 in sh )   or $1
$argv[$#argv] (* the last parameter *)
$0          (* command *)
$?name      (* if variable name is defined, 1 else 0 *)
$status     (* $? in sh *)
```



* Control Structure

* foreach

```
foreach name (wordlist)
    commands
end
```

```
for name in wordlist
do
    commands
done
```

```
foreach word ($*)
foreach word ($argv)
foreach word (argv[2-])
foreach file (*)
foreach file (*.c)
foreach name (`command`)
```

* while

```
while (expression)
    commands
end
```

- EX.

```
echo What is your name \?
set name=$< (* read in sh *)
while ($name != "peter")
    echo wrong \! enter again.
    set name $<
end
```

* repeat

```
repeat count command
```



*Expression.

```
% set age = 24 (* a string *)
% @ max = 10 (* a number, an expression *)
% @ max = ($age + 5) (* OK, type conversion *)
% @ max++
```

* Relational Operations.

```
% @ flag = $max > 50
0 (* false; 1 true *)
```

* if

```
if (expression) command
```

```
if (expression) then
    commands
endif
```

```
if (expression) then
    commands
else if (expression) then
    commands
..
else
    commands
endif
```

* EX.

```
if ($1 == stop)
if ($1 != stop)
if ($file =~ *.o) (* pattern checking *)
if ($file !~ *.o)
if (-x $file)
if ({grep -s pp test.*}) (* if ({ command}) .. *)
```



< Bourne Shell >

```
echo What is your age\  
read age  
if [ $age -le 6 ]  
then  
    echo What a nice child!  
elif [ $age -gt 6 -a $age -le 9 ]  
then  
    echo That\'s a good age!  
elif [ $age -gt 9 -a $age -le 20 ]  
then  
    echo Ah! In the prime of life!  
elif [ $age -gt 20 -a $age -le 30 ]  
then  
    echo Watch out, the younger ones are gaining on you  
else  
    echo Well, have fun, and don\'t look back.  
fi
```

< C Shell >

```
#  
echo What is your age\  
Set age=$<  
If ( $age <= 6 ) then  
    echo What a nice child\  
else if ( $age > 6 && $age <= 9 ) then  
    echo That\'s a good age\  
else if ( $age > 9 && $age <= 20 ) then  
    echo Ah! In the prime of life\  
else if ( $age > 20 && $age <= 30 ) then  
    echo Watch out, the younger ones are gaining on you  
else  
    echo Well, have fun, and don\'t look back.  
endif
```



* switch

```
switch (string)
case s1:
    commands
    breaksw
case s2:
    commands
    breaksw
...
default:
    commands
    breaksw
endsw
```

* break; continue; shift

* source (* similar to . in sh *)
Will not create a new process.
% source file

* ignoreeof (* ignore EOF ^D *)

* history

```
% set history = 10
% !! (* repeat the last command *)
% !3 (* the third command *)
% history
% !p
%! ?testfile?
```



```
72 % echo apple grape orange pear
apple grape orange pear
```

```
73% echo !72:2
```

```
echo grape
```

```
grape
```

```
74 % echo !72:^
```

```
echo apple
```

```
apple
```

```
75% echo !72:$
```

```
echo pear
```

```
pear
```

```
76 % echo !72: 2-4
```

```
echo grape orange pear
```

```
grape orange pear
```

```
77 % !72:0-$
```

```
echo .....
```

```
.....
```

```
% !echo:s/echo/cat
```

```
(* or ^echo^cat if change the previous command *)
```

```
* alias
```

```
% alias new old (* a->b->c*)
```

```
% alias w who
```

```
% alias
```

```
..
```

```
% unalias w
```

```
* set prompt = '! %'
```



* On/off shell variables.

\$ echo -x (* echo each command executed *)

\$ ignoreeof

\$ noclobber

(* to avoid > existing file; >> non-existing file *)

\$noglob (* no file extension *)

* Built-in commands in C shell

- Will not create a new process to execute; execute it directly.

@

alias

cd

echo

exec (execl)

exit

foreach

goto

history

if

login

logout

repeat

set

setenv

shift

source

switch

time unalias

unset

wait

while