

Chap10. Linux Shell Programming

echo

```
#!/bin/bash  
echo "hello bash"
```

```
#!/bin/bash  
echo "hello, world"
```

```
printf "hello, world"
```

```
printf "%s %s" hello, world
```

Variable

```
#!/bin/bash
```

```
mysql_id='root'  
mysql_dir='/etc/mysql'  
echo $mysql_id  
echo $mysql_dir
```

```
#!/bin/bash
```

```
h="hello"  
w="world"  
echo "${h}, ${w}"
```

export

```
#!/bin/bash  
# export 활용
```

```
export MY_NAME="Shin InHo"
```

```
#!/bin/bash  
# export로 정의한 변수 활용
```

```
echo ${MY_VAR}
```

Array

```
#!/bin/bash
```

```
arr=("hello" "world" 1 2 3 4 5)
```

```
echo "배열 전체 : ${arr[@]}"
```

```
echo "배열 원소의 갯수 : ${#arr[@]}"
```

```
echo "배열 첫번째 : ${arr}, 혹은 ${arr[0]}"
```

```
echo "5번 index를 갖는 배열의 원소 : ${arr[5]}"arr[5]="five"
```

```
echo "5번 index를 갖는 배열의 원소 : ${arr[5]}"
```

```
# 5번 원소 해제
```

```
unset arr[5]
```

```
echo "5번 원소 삭제 후"
```

```
echo "5번 index를 갖는 배열의 원소 : ${arr[5]}"
```

```
echo "6번 index를 갖는 배열의 원소 : ${arr[6]}"
```

List

```
#!/bin/bash
```

```
daemons=('httpd' 'mysql' 'vsftpd')
```

```
echo ${daemons[1]}
```

```
echo ${daemons[@]}
```

```
echo ${daemons[*]}
```

```
echo ${#daemons[@]}
```

```
fileList=$(ls)
```

```
echo ${fileList[*]}
```

매개변수

```
#!/bin/bash
```

```
echo "Process Id:$$"  
echo "script name:${0}"  
echo "매개변수 갯수 :${#}"  
echo "전체 매개변수 값 :${*}"  
echo "전체 매개변수 값2 :${@}"  
echo "매개변수 1 :${1}"  
echo "매개변수 2 :${2}"
```

expr

```
#!/bin/bash
```

```
num=`expr $( 3 $( * 5 $( / 4 + 7 `
```

```
echo $num
```


문법

```
#!/bin/bash
function func(){
    echo "함수_func()"
}
```

```
#함수 호출
func
```

■ If문 함수(1/3)

```
#!/bin/bash
```

```
function func(){  
    a=10  
    b=5
```

```
    if [ ${a} -eq ${b} ]; then  
        echo "a와 b는 같다."  
    fi
```

```
    if [ ${a} -ne ${b} ]; then  
        echo "a와 b는 같지 않다."  
    fi
```

■ If문 함수(2/3)

```
if [ ${a} -gt ${b} ]; then
    echo "a가 b보다 크다."
fi
```

```
if [ ${a} -ge ${b} ]; then
    echo "a가 b보다 크거나 같다."
fi
```

```
if [ ${a} -lt ${b} ]; then
    echo "a가 b보다 작다."
fi
```

```
if [ ${a} -le ${b} ]; then
    echo "a가 b보다 작거나 같다."
fi
```

```
}
```

■ If문 함수(3/3)

#함수 호출
func

논리연산자

항 목	true 반환 조건
-eq	두 값의 값이 경우
-ne	두 값이 다른 경우
-lt	오른쪽 값보다 왼쪽 값이 작은 경우
-le	오른쪽 값보다 왼쪽 값이 작거나 같은 경우
-gt	오른쪽 값보다 왼쪽 값이 큰 경우
-ge	오른쪽 값보다 왼쪽 값이 크거나 같은 경우
-z	문자열의 길이가 0인 경우 (-z \$VALUE와 같이 씀)
-n	문자열의 길이가 0이 아닌 경우 (-n \$VALUE와 같이 씀)
==	두 개의 문자열이 동일한 경우
!=	두 개의 문자열이 서로 다른 경우
<	왼쪽의 문자열이 오른쪽의 문자열보다 정렬 시 선행되는 경우
>	오른쪽의 문자열이 왼쪽의 문자열보다 정렬 시 선행되는 경우

파일관련 조건문

항 목	내 용	예제
&&	두 논리 식에 AND를 수행	if [condition 1] && [condition 2]
	두 논리 식에 OR을 수행	if [condition 1] [condition 2]
!	논리 식의 결과값에 NOT을 수행	if [! condition]

파일관련 조건문

조건	설명
if [-d \${변수}]; then if [! -d \${변수}]; then	\${변수}의 디렉토리가 존재하면 참이 성립합니다. \${변수}의 디렉토리가 존재하지 않으면 참이 성립합니다.
if [-e \${변수}]; then if [! -e \${변수}]; then	\${변수}라는 파일이 존재하면 참입니다. \${변수}라는 파일이 존재하지 않으면 참입니다.
if [-L \${변수}]; then	파일이 symbolic link이면 참입니다.
if [-s \${변수}]; then	파일의 크기가 0보다 크면 참입니다.
if [-S \${변수}]; then	파일 타입이 소켓이면 참입니다.
if [-r \${변수}]; then	파일을 읽을 수 있으면 참입니다.
if [-w \${변수}]; then	파일을 쓸 수 있으면 참입니다.
if [-x \${변수}]; then	파일을 실행할 수 있으면 참입니다.
if [-f \${변수}]; then	파일이 정규 파일이면 참입니다.
if [-c \${변수}]; then	파일이 문자 장치이면 참입니다.
if [\${변수1} -nt \${변수2}]; then	변수1의 파일이 변수2의 파일보다 최신 파일이면 참입니다.
if [\${변수1} -ot \${변수2}]; then	변수1의 파일이 변수2의 파일보다 최신이 아니면 참입니다.
if [\${변수1} -ef \${변수2}]; then	변수1의 파일과 변수2의 파일이 동일하면 참입니다.

함수와 조건문

```
function func(){  
    a=aa  
    b=bb  
    if [ -f ${a} -a -d ${b} ]; then  
        echo "a는 파일이고 b는 디렉토리 "  
    fi  
}
```

```
#함수 호출  
func
```


Read와 조건문

```
#!/bin/sh
# file-name: sample.sh

read VALUE

if [ $VALUE -eq 1 ]; then
    echo "$VALUE is 1"
elif [ $VALUE -gt 1 ] && [ $VALUE -le 5 ]; then
    echo "1 < $VALUE ≤ 5"
else
    echo "$VALUE > 5 or $VALUE < 1"
fi
```

case

```
#!/bin/bash
```

```
case ${1} in
    "linux") echo "리눅스" ;;
    "unix") echo "유닉스" ;;
    "windows") echo "윈도우즈" ;;
    "MacOS") echo "맥OS" ;;
    *) echo "머야" ;;
esac
```

함수와 For (1/3)

```
#!/bin/bash
```

```
function func(){  
    echo "사용예1"  
    for i in 1 2 3 4 5  
    do  
        echo "${i}"  
    done  
  
    echo "사용예2"  
    list="1 2 3 4 5"  
    for i in ${list}  
    do  
        echo "${i}"  
    done
```

함수와 For (1/3)

```
echo "사용예3"  
for i in {1..5}  
do  
    echo "${i}"  
done
```

```
echo "사용예4: 크기를 2만큼 증가시키면서 출력"  
for i in {1..5..2}  
do  
    echo "${i}"  
done
```

함수와 For (1/3)

```
echo "사용예5: 배열을 이용"
arr=(1 2 3 4 5)
for i in "${arr[@]}"
do
    echo "${i}"
done
```

```
echo "사용예6: C와 유사한 형식의 for문"
for ((i=0; i<5; i++)); do
    echo "${i}"
done
```

```
}
```

```
#함수 호출
func
```

■ exit

```
#include <stdlib.h>
#include <stdio.h>
int main(int argc, char* argv[]){
    if(argc<2){
        fprintf(stderr,"인자는 2개 이상이어야 합니다.\n");
        exit(16);
    }
}
```

■ exit

```
#!/bin/bash
```

```
function func(){  
    ./a.out  
    echo "오류 코드 ($?)"  
}
```

```
#함수 호출  
func
```

■ exit

```
#!/bin/bash
```

```
exit 16
```

```
#!/bin/bash
```

```
function func(){  
    ./exam2.sh  
    echo "오류 코드 ($?)"  
}
```

```
#함수 호출  
func
```