

정리하기

1-1) 함수, 전역변수, 지역변수를 사용한 예제

var="This is non-function variable" → 전역변수

```
func()
{
    local var="I'm in function"
    echo "Where are you?"
    echo $var
}
```

출력
⇒ This is non-function variable
Where are you?
I'm in function
I'm not in function

echo \$var → 전역변수 출력

func → 함수 인 명명식

var="I'm not in function" → 전역 변수 재설정

echo \$var → 재설정된 변수값으로 출력

1-2) 다중 조건문을 사용한 예제

echo How long does it take to go to school?

read ans

case \$ans

[Tt][Ww][Oo]) echo On my god!!

TWO 가 들어간 tWO가

[Oo][Nn][Ee]) echo That's normal!!

들어도 첫번째문자만 출력

*) echo pardon? !! → 둘다 해당되지 않으면 출력

2) 지정된 형식을 부여하여 문자열 혹은 숫자를 출력할 때, echo 대신 printf 사용

ex) printf "%s\n%.3s" Hello this is time. ⇒ Hello\thi\is\ime 3

printf "%d.%s\n%.2s" 24 seconds left ⇒ 24seconds\le

printf "%f.%s\n%.5" 3.141592 means π. ⇒ 3.14 means\pi

3-1) While, sleep 은 이용하는 예제

while true; do

sleep 3

3초마다 1번씩
⇒ abc.txt 이 있는 내용을 읽어들이는

cat abc.txt

done

● 3. 이해하기

```
#!/bin/sh
str="I am global variable"
foo()
{
    local str="I am local variable"
    echo "This is in function"
    echo $str
}
echo "start shell-program"
echo $str
foo
echo $str
str="I am global variable2"
echo $str
foo
echo $str
exit 0
```

<표 III-15> test02-01.sh

● 3. 이해하기

2) 다중 조건문

앞 강의에서 if조건문과 여러 가지 조건을 비교하는 AND, OR에 대하여 학습하였다. 이와 유사한 다중의 조건을 주어 비교하는 방법에 대하여 이해하고 실습을 진행한다.

① case 다중 조건문

- case 조건/ a)../b)../.../esac 조건이 a)일경우 a문장을 b)일경우 b)문장을 실행함
- 다중의 조건을 간단히 표현
- 각 조건의 종료를 ;; (세미콜론 두 개)로 표현
- a,b,c의 조건을 표현하는 다양한 방법이 있는데 예제를 통하여 학습함

② 예제 test02-02.sh

```
#!/bin/sh
echo "Do you love me?"
read ans
case $ans in
    [yY][eE][sS]) echo Thank you. ;;
    [nN]*) echo I am very sad. ;;
    *) echo really? ;;
esac
exit 0
```

<표 III-16> test02-02.sh

- 실행결과

● 3. 이해하기

```
#!/bin/sh
printf "%s\n" hello
printf "%s\n" hello world
printf "%s  " hello
printf "%s" world
printf "\n"
printf "%s %d %3s\n" hello 'expr 1 + 2 ' hello
exit 0
```

<표 III-18> test02-04.sh

• 실행결과

```
root@ubuntu:/home/sh/test/test02# sh test02-04.sh
hello
hello
world
hello world
hello 3 hel
root@ubuntu:/home/sh/test/test02#
```

<그림 III-45> test02-04.sh 실행결과

- printf "%s\n" hello는 hello라는 문자열을 printf문의 주어진 형식인 %s로 출력하므로 전체 문자열인 hello가 출력
- 여기서 \n 표시는 줄을 바꾸라는 의미로 개행문자라고 함
- printf "%s\n" "hello world" 는 hello world라는 문자열을 하나의 문자열로 보고 printf문의 주어진 형식인 %s로 출력하므로 전체 문자열인 hello world가 출력
- printf "%s " hello 는 hello라는 문자열을 하나의 문자열로 보고 printf문의 주어진 형식대로 출력함, 단 개행 문자가 없