Shell Script

1. Basic

- 쉘스크립트를 작성하고 실행하는 방법

```
$ vi first.sh
```

```
#!/bin/sh
echo Hello World
```

```
$ source first.sh
Hello World
```

2. Variable

변수 선언 및 사용방법

`$ vi var.sh`

```
#!/bin/sh
MY_MESSAGE="Hello World" # MY_MESSAGE 변수를 선언
echo $MY_MESSAGE # MY_MESSAGE 변수를 사용
```

`$ source var.sh`

```
Hello World
```

변수 입력 받기

```
$ vi var2.sh
```

```
#!/bin/sh
echo What is your name?
read MY_NAME # MY_NAME 변수를 입력 받음
echo "Hello $MY_NAME - hope you're well."
```

```
$ source var2.sh
```

```
What is your name?
shell
Hello shell - hope you're well.
```

변수 선언 시기

```
$ vi var3.sh
```

```
#!/bin/sh
echo "MYVAR is: $MYVAR" # 변수가 선언되지 않아 MYVAR 변수가 출력
MYVAR="hi there"
echo "MYVAR is: $MYVAR"
```

```
$ source var3.sh
```

```
MYVAR is:
hi there
MYVAR is: hi there
```

export를 이용해서 변수 선언

```
$ export MYVAR="Hello" # MYVAR 변수 선언
$ echo $MYVAR
Hello
```

```
$ source var3.sh
```

```
MYVAR is: Hello
hi there
MYVAR is: hi there
```

변경된 변수 확인

```
$ echo $MYVAR
hi there
```

선언된 변수 확인

```
$ export | grep declare
```

파일 만들기

```
$ vi user.sh
```

```
#!/bin/sh
echo "What is your name?"
read USER_NAME
echo "Hello $USER_NAME"
# USER_NAME_file 변수로 인식
echo "I will create you a file called $USER_NAME_file"
touch $USER_NAME_file
```

```
$ source user.sh
```

```
What is your name?
hello
Hello hello
I will create you a file called
touch: missing file operand
Try 'touch --help' for more information.
```

```
$ vi user2.sh
```

```
#!/bin/sh
echo "What is your name?"
read USER_NAME
echo "Hello $USER_NAME"
# 입력받은 변수 USER_NAME과 _file이 결합되어 실행
echo "I will create you a file called ${USER_NAME}_file"
touch "${USER_NAME}_file"
```

```
$ source user2.sh
```

```
What is your name?
hello
Hello hello
I will create you a file called hello_file
$ ls hello_file # 입력받은 문자열로 파일이 생성됨
hello_file
```

`#:아규먼트갯수, $0:실행 쉘, $1~$9:입력받은 아규먼트`

```
$ vi var4.sh
```

```
#!/bin/sh
echo "I was called with $# parameters"
echo "My name is $0"
echo "My first parameter is $1"
echo "My second parameter is $2"
echo "All parameters are $@"
```

```
$ source var4.sh a b c
```

```
I was called with 3 parameters
My name is -bash
My first parameter is a
My second parameter is b
All parameters are a b c
```

여러개의 변수 받기

```
$ vi var5.sh
```

```
#!/bin/sh
IFS=,
echo "Please input some data separated by comma ..."
read x y z
echo "x is $x y is $y z is $z"
```

```
$ source var5.sh
```

```
Please input some data separated by comma ...
a,b,c
x is a y is b z is c
```

shell command를 변수에 저장

```
$ vi var6.sh
```

```
#!/bin/sh
var=`whoami`
echo $var
```

```
$ source var6.sh
```

```
ubuntu
```

3. Loop

3.1 For

```
$ vi for.sh
```

```
#!/bin/sh
for i in 1 2 3
do
  echo "Looping ... number $i"
done
```

```
$ source for.sh
```

```
Looping ... number 1
Looping ... number 2
Looping ... number 3
```

3.2 while

```
$ vi while.sh
```

```
#!/bin/sh
INPUT_STRING=hello
while [ "$INPUT_STRING" != "bye" ]
do
  echo "Please type something in (bye to quit)"
  read INPUT_STRING
  echo "You typed: $INPUT_STRING"
done
```

```
$ source while.sh
```

```
Please type something in (bye to quit)
hello
You typed: hello
Please type something in (bye to quit)
bye
You typed: bye
```

4. condition

4.1 if, then, else

```
$ vi condition1.sh
```

```
#!/bin/sh
# -le:<=, -lt:<, -gt:>, -ge:>=, -n:!(not), -ne:!=
if [ $1 -lt 10 ]
then
  echo "$1 < 10"
else
  echo "$1 >= 10"
fi
```

```
$ source condition1.sh 5
```

```
5 < 10
```

```
$ source condition1.sh 15
```

```
15 >= 10
```

4.2 [condition] && (True) || (False)

```
$ vi condition3.sh
```

```
#!/bin/sh
[ $1 -lt 10 ] && echo "$1 < 10" || echo "$1 >= 10"
```

```
source condition2.sh 5
```

```
5 < 10
```

```
source condition3.sh 15
```

```
15 >= 10
```

4.3 case

```
$ vi case.sh
```

```
#!/bin/sh
echo "Please talk to me ..."
while :
do
  read INPUT_STRING
  case $INPUT_STRING in
        hello)
                echo "Hello yourself!"
                ;;
        bye)
                echo "See you again!"
                break
                ;;
        *)
                echo "Sorry, I don't understand"
                ;;
  esac
done
echo
echo "That's all folks!"
```

```
$ source case.sh
```

```
Please talk to me ...
hello
Hello yourself!
world
Sorry, I don't understand
bye
See you again!

That's all folks!
```

5. Function

```
# vi function.sh
```

```
#!/bin/sh

### function

myfunc()
{
  echo "I was called as : $@"
  x=2
}

### main script

echo "Script was called with $@"
x=1
echo "x is $x"
myfunc 1 2 3
echo "x is $x"
```

```
$ source function.sh
```

```
Script was called with
x is 1
I was called as : 1 2 3
x is 2
```