

Shell script

(주) 시엠아이코리아
정재윤

Python 목차

1. Shell script 시작
2. 변수 정의
3. 숫자 계산
4. 사용자 입력
5. 숫자 및 문자 비교
6. 조건문
7. 관계연산
8. for, while
9. 함수



Shell Script 시작

`#!/bin/bash`

- 스크립트 파일의 가장 위의 첫 라인은 "`#!/bin/bash`"로 시작
- Bash로 작성되었다는 것을 의미하며 bash를 기준으로 실행된다는 것을 의미
- Bourne Shell인 경우 "`#!/bin/sh`"

셸 스크립트의 실행

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

```
echo bash script test!!
```

작성한 스크립트 파일에 실행 권한이 있는지 확인 필수
chmod 명령어로 실행권한 부여

```
chmod +x test.sh
```



변수 정의

변수명=값

Shell에 있어 변수의 사용은 매우 간단히 정의가 됨

주의할 것은 '='사이의 공백이 없어야 된다.

대소문자 구별이 가능하고 숫자형 변수와 문자형 변수의 구별이 없음

```
a=10
```

```
b="hellow linux"
```

변수의 내용을 사용할 때는 변수 명 앞에 '\$'를 붙인다



숫자계산

expr

숫자 계산을 위한 명령어

더하기, 빼기, 곱하기, 나누기, 나머지 값을 연산

expr를 변수로 지정하는 경우 역작은 따옴표(`)를 사용
연산자 *와 괄호 앞에는 역슬래시를 같이 사용
연산자와 숫자, 변수, 기호 사이에는 space가 존재해야 함

```
A=2
```

```
B=4
```

```
echo A=$A
```

```
echo B=$B
```

```
result1=`expr $A + $B`
```

```
echo A+B=$result1
```



사용자 입력 받기

read

read를 사용하여 사용자 입력을 변수에 받음

```
read input
```

```
echo 'input string : '
```

```
read input
```

```
echo you input : $input
```



숫자 및 문자 비교

숫자 비교

| | |
|-------------------|----------------|
| [\$A -gt \$B] : | A가 B보다 크다 |
| [\$A -lt \$B] : | A가 B보다 작다. |
| [\$A -ge \$B] : | A가 B보다 크거나 같다. |
| [\$A -le \$B] : | A가 B보다 작거나 같다. |
| [\$A -eq \$B] : | A와 B가 같다. |
| [\$A -ne \$B] : | A와 B가 다르다. |

문자 비교

| | |
|------------------------------|-----------------------|
| ["string1" = "string2"] : | 두 문자열이 같은 경우(== 도 가능) |
| ["string1" != "string2"] : | 두 문자열이 다른 경우 |
| [-z "string"] : | 문자열의 길이가 0인 경우 |
| [-n "string"] : | 문자열의 길이가 0이 아닌 경우 |



조건문

if, elif, else

```
if [ 조건문 ]  
then  
    문장  
elif [ 조건문 ]  
then  
    문장  
else  
    문장  
fi
```



조건문

if, elif, else 예제

```
num=10
compare=7

if [ $num -gt $compare ]
then
    echo "$num은 $compare보다 큰 수입니다."
    exit
elif [ $num -lt $compare ]
then
    echo "$num은 $compare보다 작은 수입니다."
    exit
else
    echo "$num은 $compare와 같은 수 입니다."
fi
```



관계 연산자

-a : and
-o : or

```
echo '숫자를 입력하세요 : '  
read num  
  
if [ $( $num -lt 7 ) -o $( $num -gt 9 ) ]  
then  
    echo "7, 8 or 9의 숫자를 입력하세요"  
else  
    echo "$num 입력하였습니다."  
fi
```



for 문

```
for 변수 in 변수값1 변수값2 변수값3 ...  
do  
    문장  
done
```

```
for i in 1 2 3 4 5 6  
do  
    echo $i  
done
```

or

```
for i in {1 .. 6}  
do  
    echo $i  
done
```



while 문

```
while [ 조건문 ]  
do  
    문장  
done
```

```
echo "숫자 입력 : "  
read num  
  
while [ $num -gt 0 ]  
do  
    echo $num  
    num=`expr $num - 1`  
done
```



함수

함수는 불러주기 전에 선언되어야 함

```
two_sum_fun()
{
    echo "첫번째 인자 : $1"
    echo "두번째 인자 : $2"
    sum=`expr $1 + $2`

    return $sum
}
```

```
two_sum_fun 1 2
result=$?
```

```
echo "합은 $result 입니다."
```

