

# Shell Script 기초 문법

## 1. 변수

### 1.1 환경변수

### 1.2 인자변수

### 1.3 리스트 변수

## 2. 기본 실행 및 권한 부여

### 2.1 방법

### 연산자

### 날짜 연산

### 조건문

### AND / OR

### 반복문 - for

### 반복문 - while

## 1. 변수

- 셸변수는 대소문자를 구분
- 셸변수는 데이터를 문자열로 저장.
  - 실제 수치를 대입해도 문자열로 저장
  - 계산이 필요하면 문자열을 자동으로 수치로 변환하여 변환하여 계산 후 다시 문자열로 저장
- 셸변수의 값을 사용할 때는 변수명 앞에 \$를 붙임
  - 예) echo \$Test
- 셸변수에 값을 대입할 때는 변수명 앞에 \$를 붙이지 않음. 변수명=데이터 사이에 띄어쓰기는 허용되지 않음
  - 예) name='aa'
  - 예) name = 30 (X)
- 셸변수는 타입을 명시해줄 필요가 없음

## 1.1 환경변수

- 셸을 가동하면 기본적으로 세팅되어있는 변수들이 존재. 일반변수처럼 값을 얻어오거나 세팅할 수 있음

- \$\$ : 쉘의 프로세스 번호 (pid)
- \$# : 인자의 개수
- \$0 : 쉘 스크립트 이름
- \$? : 최근 실행한 명령어의 종료 값
  - 0 성공, 1~125 에러, 126 파일이 실행가능하지 않음, 128~255 시그널 발생

## 1.2 인자변수

- 쉘 스크립트에 인자를 넘겨줄 때 그 인자들에 대한 정보를 가지고 있는 변수
  - \$1~\$nnn : 넘겨진 인자들
  - \$\* : 스크립트에 전달된 인자들을 모아놓은 문자열, 하나의 변수에 저장되며 IFS 환경변수의 첫번째 문자로 구분
  - @\$ : \$\* 와 같으나 IFS 변수의 영향을 받지 않음

## 1.3 리스트 변수

- 선언
  - 변수명=(데이터1 데이터2 데이터3)
- 사용
  - \${변수명[인덱스번호]}
- 코드예

```
#!/bin/bash

daemons=("haha" "hoho" "hihi") # 변수선언
echo ${daemons[1]} # 두번째 인덱스 출력
echo ${daemons[@]} # daemons 배열의 모든 데이터 출력
echo ${daemons[*]} # daemons 배열의 모든 데이터 출력
echo ${#daemons[@]} # daemons 배열의 배열 크기 출력

filelist=$(ls) # 해당 쉘 스크립트 실행 디렉토리의 파일 리스트를 배열로 변수 선언
echo ${filelist[*]} # filelist 배열의 모든 데이터 출력
```

- 참고
  - \${daemons[1]}이 아니라 \$daemons[1] 이라고 입력할 경우, 쉘스크립트는 daemons 까지를 읽고 데이터로 변환시키고 [1] 은 일반적인 스트링으로 해석 → haha[1] 이라고 출력됨

## 2. 기본 실행 및 권한 부여

- 파일의 가장 위 첫 라인은 `#!/bin/bash` 로 시작
- 셸스크립트 파일은 코드를 작성한 후에 실행 권한을 부여해주어야 함
- 일반적으로 '파일이름.sh' 와 같은 형태로 파일을 작성
- 주석은 `#내용`으로 처리

### 2.1 방법

1. `vi` 파일이름.sh 로 파일을 생성
2. 코드 작성 및 `:wq`로 저장
3. `chmod 777 ./파일이름.sh` 로 권한부여

## 연산자

- `expr` : 숫자 계산
  - `expr` 을 사용하는 경우 역 작은 따옴표(```)를 사용해야 한다
  - 연산자 `*` 와 괄호 `()` 앞에는 역슬래시를 붙여서 같이 사용해야 한다.
  - 연산자와 숫자, 변수, 기호 사이에는 공백을 넣어야 한다.
- 코드예

```
#!/bin/bash  
  
cal=`expr \( 3 \* 5 \) / 3 + 7`
```

## 날짜 연산

- `date -d`
  - 날짜 연산으로 구해진 날짜를 다시 연산하고자 한다면, `-i` 옵션을 붙여주어야 함
- 코드예

```
#!/bin/bash  
  
now=2021-11-01  
# 특정 날짜를 기준으로 계산
```

```
now=`date -d "$now + 32days"`
now=`date -d "20211101 + 32days"`
now=`date -d "2021-11-01 + 32days"`
# 월 계산
now=`date -d "$now + 3months"`

now=`date -d "+32days"` # 현재 날짜를 기준으로 계산
echo $now
```

```
start=2021-11-23
start=`date -I -d "$start + 7days"`
start=`date -I -d "$start + 7days"`
```

## 조건문

- 기본 if 문법
  - if 조건 괄호 [ 뒤와 ] 앞에는 반드시 공백이 존재해야 함

```
if [ 조건 ]
then
    명령문 # 참일 때 실행됨
else
    명령문 # 거짓일 때 실행됨
fi
```

- 코드예

```
#!/bin/bash
echo $0

if [ 1 -eq 2 ]
then
    echo "same values"
else
    echo "diff values"
fi
```

- 조건
  - 문자열 비교
    - 문자1 == 문자2 // 문자 1과 문자 2가 일치
    - 문자1 != 문자2 // 문자 1과 문자 2가 일치하지 않는다.

- z 문자 // 문자가 null 이면 참(값이 없으면 true)
- n 문자 // 문자가 null 이 아니면 참
- 수치 비교
  - 값1 -eq 값2 //값이 같음(equal)
  - 값1 -ne 값2 //값이 같지 않음(not equal)
  - 값1 -lt 값2 //값1이 값2보다 작음(less than)
  - 값1 -le 값2 //값1이 값2보다 작거나 같음(less or equal)
  - 값1 -gt 값2 //값1이 값2보다 큼(greater than)
  - 값1 -ge 값2 //값1이 값2보다 크거나 같음(greater or equal)
- 파일 검사
  - e 파일명 //파일이 존재하면 참
  - d 파일명 //파일이 디렉토리면 참
  - h 파일명 //파일이 심볼릭 링크 파일이면 참
  - f 파일명 //파일이 일반파일이면 참
  - r 파일명 //파일이 읽기 가능하면 참
  - s 파일명 //파일크기가 0이 아니면 참
  - u 파일명 //파일이 set-user-id가 설정되면 참
  - w 파일명 //파일이 쓰기 가능이면 참
  - x 파일명 //파일이 실행 가능이면 참

## AND / OR

- 기본 and 문법

```
[ 조건문 ] && [ 조건문 ] && [ 조건문 ]
```

- 기본 or 문법

```
[ 조건문 ] || [ 조건문 ] || [ 조건문 ]
```

- and, or 혼용

```
[ 조건문 ]&&[ 조건문 ]||[ 조건문 ]
```

- 코드예

```
#!/bin/bash

if {[ 3 -eq 3 ]&&[ 2 -eq 2 ]}||[ 5 -eq 3 ]
then
    echo equal
fi
```

## 반복문 - for

- 기본 for 문법

```
for 변수 in 변수값1 변수값2 ...
do
    명령문
done
```

- 코드예

```
#!/bin/bash

fruits=("apple" "banana")
for fruit in ${fruits[*]}
do
    echo $fruit
done
```

```
#!/bin/bash
#현재 디렉토리에 있는 파일과 디렉토리를 모두 출력

for data in $(ls)
do
    echo $data
done
```

# 반복문 - while

- 기본 while 문법

```
while [ 조건문 ]  
do  
    명령문  
done
```

- 코드예

```
#!/bin/bash  
  
num=3  
  
while [ 6 != $num ]  
do  
    echo $num  
    num=`expr $num + 1`  
done
```

```
#!/bin/bash  
  
start=2021-11-23  
  
while [ "$start" != 2021-11-30 ]  
do  
    echo $start  
    start=`date -I -d "$start + 1days"`  
done
```