

# 셸 스크립트

☰ Tags	
📅 Date	@07/12/2022
☑ Done	<input type="checkbox"/>
☰ 메모	
▼ 분기	
☰ 속성 1	
▼ 상태 1	

`#!/bin/sh` : 셸 스크립트 선언부

- '\$' 문자가 들어간 글자를 출력하려면 "로 묶어주거나 앞에 \' 붙인다 ⇒ \ 가 escape

```
var = "Hello"
echo $var           ▶ Hello
echo "$var"         ▶ Hello

echo '$var'         ▶ $var
echo \$var          ▶ $var

read var
// [World] 라고 입력. ▶ 변수에 대입

echo '$var' = $var   ▶ $var = World
exit 0
```

- **변수에 대입된 값은 모두 문자열로 취급**
- 변수 값을 숫자로해서 산술 연산을 하려면 `expr` 을 사용 + ``` 로 감싸준다.
- 수식에 괄호 또는 곱하기는 앞에 \를 붙인다.

```
var = "Hello"
echo $var           ▶ Hello
echo "$var"         ▶ Hello

echo '$var'         ▶ $var
echo \$var          ▶ $var

read var
// [World] 라고 입력. ▶ 변수에 대입

echo '$var' = $var   ▶ $var = World
exit 0
```

- “[ 조건 ]“ 사이의 각 단어에는 모두 공백이 있어야 한다.

```
if [ 조건 ]
then
    참일 경우 실행
fi
```

```
if [ 조건 ]
then
    참일 경우 실행
else
    거짓일 경우 실행
fi
```

중복 if문 (else if) => elif

문자열 비교	결과
"문자열1" = "문자열2"	두 문자열이 같으면 참
"문자열1" != "문자열2"	두 문자열이 같지 않으면 참
-n "문자열"	문자열이 NULL(빈 문자열)이 아니면 참
-z "문자열"	문자열이 NULL(빈 문자열)이면 참

산술 비교	결과
수식1 -eq 수식2	두 수식(또는 변수)이 같으면 참
수식1 -ne 수식2	두 수식(또는 변수)이 같지 않으면 참
수식1 -gt 수식2	수식1이 크다면 참
수식1 -ge 수식2	수식1이 크거나 같으면 참
수식1 -lt 수식2	수식1이 작으면 참
수식1 -le 수식2	수식1이 작거나 같으면 참
!수식	수식이 거짓이면 참

파일 조건	결과
-d 파일이름	파일이 디렉터리면 참
-e 파일이름	파일이 존재하면 참
-f 파일이름	파일이 일반 파일이면 참
-g 파일이름	파일에 set-group-id가 설정되면 참
-r 파일이름	파일이 읽기 가능하면 참
-s 파일이름	파일 크기가 0이 아니면 참
-u 파일이름	파일에 set-user-id가 설정되면 참
-w 파일이름	파일이 쓰기 가능 상태면 참
-x 파일이름	파일이 실행 가능 상태면 참

- case ~ esac

```
#!/bin/sh

echo "계속하시겠습니까? (yes/no)"

read answer

case $answer in
  yes|y|Y|Yes|YES)
    echo "계속"
  [nN]*)
    echo "그만"
  esac
```



exit 1 → 비정상 종료  
exit 0 → 정상 종료

- and : '-a' 또는 '&&'를 사용
- or : '-o' 또는 '||'를 사용

- 반복문 - for문

```
for 변수 in 값1 값2 값3 ...
do
    반복할 문장
done

-----

#!/bin/sh
hap=0
for i in 1 2 3 4 5 6 7 8 9 10
do
    hap=`expr $hap+$i`
done

echo "1부터 10까지의 합 : "$hap
exit 0
```



for i in 1 2 3 4 5 6 7 8 9 10  
= for(i=1;i≤10;i++)  
= for i in 'seq 1 10'

- 반복문 - while문

```
#!/bin/sh
hap=0
i=1
while [ $i -le 10 ]
do
    hap=`expr $hap + $i`
    i=`expr $i + 1`
done

echo "1부터 10까지의 합 : "$hap
exit 0
```



[ 1 ] = [ : ] 은 항상 참을 반환

- 반복문 - until문

- while문과 용도가 거의 같지만, until문은 조건식이 참일 때까지 반복

```
while [ $i -le 10 ] => until [ $i -gt 10 ]
```

- break : 반복문 종료 시
- continue : 조건식으로 돌아가게 함
- exit : 해당 프로그램을 완전히 종료
- return : 함수 안에서 사용. 함수를 호출한 곳으로 돌아간다.