**목표 : shell script 프로그래밍 실습**

**1. hello world를 출력.**

#!/bin/bash

echo "Hello, World!"

exit 0

**텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명**

**→ echo(내장 명령어)를 이용해서 “Hello, World!”를 출력한다.**

**2. 숫자를 입력 받아 hello world를 여러 번 출력.**

#!/bin/bash

echo "반복할 횟수를 입력하세요:"

read num

for((i = 0; i<num;i++))

do

echo "Hello, World!"

done

exit 0

**텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명**

**→ read를 이용해서 반복할 횟수를 입력받는다.**

**for((i = 0 ; i<num ; i++)) : for 루프를 이용해서 입력 받은 숫자 만큼 반복하여 출력한다.**

**3. 두 숫자와 연산자(+ 또는 -)를 입력 받아 계산하기.**

#!/bin/bash

echo "num1 (+ or -) num2 :"

read num1 opr num2

case $opr in

+)

echo $((num1 + num2))

;;

-)

echo $((num1 - num2))

;;

\*)

echo "옳지 않은 연산자입니다."

;;

esac

exit 0

**텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명**

**→ read를 이용해서 연산식(+ or -)을 입력받는다.**

**case를 이용해서 +와 -를 구분하고, + - 가 아닌 연산자를 입력했을 경우에는 “옳지 않은 연산자입니다.”라는 문구를 출력한다.**

**4. 체질량 지수 (BMI) 계산하여 비만 여부를 판단.**

**▪ 정상 체중: 18.5 <= 몸무게 / (신장 \* 신장) < 23**

#!/bin/bash

echo "체중을 입력해주세요.(kg):"

read kg

echo "키를 입력해주세요.(cm):"

read cm

BMI=$(echo "scale=2; $kg / (($cm/100)\*($cm/100))" | bc)

if (( $(echo "$BMI < 18.5" | bc -l) )); then

echo "저체중 입니다."

elif (( $(echo "$BMI < 23" | bc -l) )); then

echo "정상체중입니다."

else

echo "과체중입니다"

fi

exit 0

**텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명**

**→ 입력받은 정보를 바탕으로 BMI를 구한다.**

**scale을 2로 조정하여 소숫점 2째 자리 까지 취급한다.**

**If문을 이용하여 범위를 나눠 알맞은 대답을 출력한다.**

**5. case문을 활용한 예제 실습**

**▪ 입력에 따라 분기하여 서로 다른 출력. 다양한 입력에 대한 방어 코드.**

#!/bin/bash

echo "리눅스가 재미있나요?(yes/no):"

read ans

case $ans in

yes | Yes | YES | y | Y)

echo " >> choice : Yes "

echo " >> result : Linux Master ";;

[nN]\*)

echo " >> choice : No"

echo " >> result : Good Luck~ ";;

\*)

echo " >> yes or no로 입력해주세요.";;

esac

exit 0

**텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명**

**→ read를 이용하여 질문에 대한 대답을 입력받는다.**

**Case를 이용하여 ans에 따른 명령을 선택한다.**

**yes | Yes | YES | y | Y 중 하나를 입력했을 경우 아래의 명령을 실행하게 된다.**

**[nN]\*) n또는 N으로 시작하는 문자열을 입력했을 경우 아래의 명령을 실행하게 된다.**

**만약 이 중 단 한 개도 포함되지 않는 대답을 입력받았을 경우 :**

**" >> yes or no로 입력해주세요."을 출력한다.**

**6. 내부 함수를 만들어 리눅스 명령어(e.g. ls)를 실행.**

**▪ 입력 인자를 내부 함수 내의 리눅스 명령어 옵션으로 전달.**

#!/bin/bash

run\_ls(){

echo "함수 안으로 들어왔음."

ls "$@"

return

}

echo "원하는 ls 명령어 :"

read ls\_opt

echo "프로그램을 시작합니다."

run\_ls $ls\_opt

echo "프로그램을 종료합니다."

exit 0

**텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명**

**→ run\_ls() : run\_ls라는 이름의 함수를 정의한다.**

**호출되었을 경우 "함수 안으로 들어왔음."이라는 멘트를 출력하고,**

**함수가 호출될 때 전달된 모든 인자를 ls 명령어에 넘긴다.**

**Read를 이용해 원하는 ls명령어를 입력 받는다.**

**run\_ls $ls\_opt : 사용자가 입력한 옵션 인자로 함수를 호출한다.**

**만약 아무것도 입력하지 않았을 경우 현재 디렉토리의 파일 목록을 출력한다.**

**7. 입력된 이름으로 폴더가 있는 확인. 없다면 생성.**

**▪ 5개의 파일을 만들고 압축. 새로운 폴더를 생성하여 압축 해제.**

#!/bin/bash

# 사용자로부터 폴더 이름 입력 받기

echo "폴더 이름을 입력하세요:"

read folder\_name

# 폴더가 존재하는지 확인하고, 없으면 생성하기

if [ ! -d "$folder\_name" ]; then

mkdir "$folder\_name"

else

echo "$folder\_name 폴더는 이미 존재합니다."

fi

# 5개의 파일 생성하기

cd "$folder\_name"

for i in {0..4}

do

touch "file$i.txt"

done

# 동일한 이름의 하위 폴더 'files' 생성하기

mkdir "$folder\_name"\_files

# 생성한 파일들을 tar로 압축하기

tar -cvf "$folder\_name"\_files/"$folder\_name"\_files.tar file0.txt file1.txt file2.txt file3.txt file4.txt >/dev/null 2>&1

# 압축된 파일을 다시 풀어보기

tar -xvf "$folder\_name"\_files/"$folder\_name"\_files.tar -C "$folder\_name"\_files >/dev/null 2>&1

텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

**→ if [ ! -d "$folder\_name" ] : 입력한 폴더 명의 폴더가 존재하는지 확인한다.**

**만약 없을 경우 다음의 폴더 명으로 폴더를 만든다. (있다면 이를 알리는 문구 출력)**

**for i in {0..4}**

**do**

**touch "file$i.txt"**

**done**

**: 반복문을 이용하여 파일을 만든다.**

**mkdir "$folder\_name"\_files : 동일한 이름의 하위 폴더를 만든다.**

**이후 생성한 파일들을 tar로 압축한 뒤, 생성한 폴더에 푼다.**

**8. 입력된 이름으로 폴더에서 5개 이상의 파일을 만듦.**

**▪ 파일 이름대로 하위 폴더들을 생성하고 각 폴더에 해당 파일을 링크.**

#!/bin/bash

# 사용자로부터 폴더 이름 입력 받기

echo "폴더 이름을 입력하세요:"

read folder\_name

# 폴더가 존재하는지 확인하고, 없으면 생성하기

if [ ! -d "$folder\_name" ]; then

mkdir "$folder\_name"

else

echo "$folder\_name 폴더는 이미 존재합니다."

fi

# 폴더 안에 5개의 파일 생성하기

cd "$folder\_name"

for i in {0..4}

do

touch "file$i.txt"

# 각 파일과 같은 이름의 폴더 생성하기

mkdir "file$i"

# 해당 폴더 안에 파일에 대한 심볼릭 링크 생성하기

ln -s "../file$i.txt" "file$i/file$i.txt" >/dev/null 2>&1

done

**텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명**

**→ 폴더이름을 입력 받고, 다음의 폴더가 있는지 확인한다.**

**만약 있다면 이를 알리는 문구를 출력하고, 없다면 폴더를 만든다.**

**cd "$folder\_name" : 입력한 폴더로 이동한다.**

**반복문을 이용해서 폴더 안에 5개의 파일과 폴더를 만들고, 폴더 안에는 파일에 대한 심볼릭 링크를 생성한다.**

**9. 팀원의 이름과 생일 또는 전화번호를 DB.txt에 기록**

**▪ 이름과 정보를 DB.txt에 계속 추가할 수 있음. 또는 처음부터 다시 작성.**

#!/bin/bash

# DB.txt 파일 경로 설정

DB\_FILE="DB.txt"

# 파일이 없으면 생성

if [ ! -f "$DB\_FILE" ]; then

touch "$DB\_FILE"

fi

# 사용자로부터 팀원 정보 입력 받기

echo "팀원의 이름과 전화번호를 입력하세요:"

read name phone

# 입력된 정보 DB.txt에 추가

echo "$name - $phone" >> "$DB\_FILE"

**텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명**

**→ DB\_FILE="DB.txt" : 파일 경로를 설정한다.**

**조건문을 이용해서 DB.txt가 없다면 파일을 생성한다.**

**사용자에게 read를 이용해서 팀원의 정보를 입력받는다.**

**echo "$name - $phone" >> "$DB\_FILE" : 입력 받은 정보를 다음의 형식으로 DB.txt에 저장한다.**

**10. 이름으로 검색. DB.txt에서 검색된 팀원의 정보를 확인.**

#!/bin/bash

# DB.txt 파일 경로 설정

DB\_FILE="DB.txt"

# 사용자로부터 검색할 이름 입력 받기

echo "검색할 팀원의 이름을 입력하세요:"

read search\_name

# DB.txt에서 이름을 검색하고 결과 출력

result=$(grep -i "$search\_name" "$DB\_FILE")

# 결과가 존재하는지 확인

if [ -n "$result" ]; then

echo "검색된 팀원 정보:"

echo "$result"

else

echo "해당 이름의 팀원이 없습니다."

fi

**텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명**

**→ 파일 경로를 지정하고, 사용자로 부터 검색할 사람의 이름을 입력받는다.**

**grep -i "$search\_name" "$DB\_FILE" : DB.txt파일에서 대소문자 구분 없이(-i) 입력한 이름을 검색한다. 이 결과는 result에 저장한다.**

**조건문을 이용해서 result에 결과가 존재하는지(문자열이 비었는지 : -n) 확인하고 있을 경우 DB.txt에 저장되어있던 팀원의 정보를 출력한다.**

**(없을 경우 이에 대한 문구를 출력한다.)**