3-0

#!bin/bash

echo "hello World"

exit 0



3-1

#!/bin/bash

if [ -n "$1" ]

then

    c=$1

    while [ $c -ne 0 ]

    do

        echo "hello world"

        c=$(($c - 1))

    done

else

    while [ 1 ]

    do

        echo "Hello World"

    done

fi

exit 0

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

3-2

#!/bin/bash

# 숫자와 연산자를 입력받습니다.

read -p "Enter first number: " num1

read -p "Enter second number: " num2

read -p "Enter operator (+ or -): " operator

# 계산을 수행하고 결과를 출력합니다.

case $operator in

  +)

     result=$((num1 + num2))

     ;;

  -)

     result=$((num1 - num2))

     ;;

  \*)

     echo "Invalid operator!"

     exit 1

     ;;

esac

echo "Result: $result"

#!/bin/bash

# 숫자와 연산자를 입력받습니다.

read -p "Enter first number: " num1

read -p "Enter second number: " num2

read -p "Enter operator (+ or -): " operator

# 계산을 수행하고 결과를 출력합니다.

case $operator in

  +)

     result=$((num1 + num2))

     ;;

  -)

     result=$((num1 - num2))

     ;;

  \*)

     echo "Invalid operator!"

     exit 1

     ;;

esac

echo "Result: $result"

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

3-3

#!/bin/bash

# 사용자로부터 키와 체중을 입력받습니다.

read -p "Enter your height in meters (e.g., 1.75): " height

read -p "Enter your weight in kilograms (e.g., 75): " weight

# Bash에서 소수 계산을 직접 처리하기 어렵기 때문에,

# 키의 제곱을 계산하기 위해 100을 곱한 정수로 변환합니다.

height\_cm=$(echo "$height \* 100" | awk '{printf "%d\n", $1}')

weight\_kg=$(echo "$weight" | awk '{printf "%d\n", $1}')

# BMI 계산

# BMI = (체중(kg) / 키(m)의 제곱) \* 10000

# 키는 cm로 변환되었기 때문에, 10000을 곱하는 것으로 제곱을 보정합니다.

bmi=$(( (weight\_kg \* 10000) / (height\_cm \* height\_cm) ))

# BMI 값에 따라 결과 메시지를 결정합니다.

if [ $bmi -lt 185 ]; then

    status="Underweight"

elif [ $bmi -ge 185 ] && [ $bmi -le 230 ]; then

    status="Normal weight"

else

    status="Overweight"

fi

# 결과 출력

echo "Your BMI is: $bmi"

echo "You are: $status"

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

3-4

#!/bin/bash

# 질문을 출력하고 사용자의 입력을 받습니다.

read -p "Is Linux fun? (y/n): " answer

# 입력받은 답변을 소문자로 변환합니다.

answer=$(echo $answer | tr '[:upper:]' '[:lower:]')

# case 문을 사용하여 입력 처리

case $answer in

  y)

    echo "You said yes!"

    ;;

  n)

    echo "You said no."

    ;;

  \*)

    echo "Invalid input! Please answer 'y' or 'n'."

    ;;

esac

텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

3-5

#!/bin/bash

# 내부 함수를 정의합니다.

list\_directory\_contents() {

    # 함수에 전달된 인자를 ls 명령어의 옵션으로 사용합니다.

    ls $1

}

# 스크립트에 전달된 첫 번째 인자를 내부 함수에 전달합니다.

# 예를 들어, -l 또는 -a 같은 옵션을 사용할 수 있습니다.

list\_directory\_contents $1



3-6

#!/bin/bash

# 폴더 이름을 입력받습니다.

read -p "Enter the directory name: " dir\_name

# 폴더가 없으면 생성합니다.

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

    mkdir "$dir\_name"

fi

# 폴더 안에 5개의 파일을 생성합니다.

for i in {1..5}; do

    touch "${dir\_name}/file${i}.txt"

done

# 파일들을 압축합니다.

tar -czf "${dir\_name}.tar.gz" -C "$dir\_name" .

# 새로운 폴더를 생성합니다.

new\_dir\_name="${dir\_name}\_uncompressed"

mkdir "$new\_dir\_name"

# 압축 해제합니다.

tar -xzf "${dir\_name}.tar.gz" -C "$new\_dir\_name"

echo "The directory and files have been created, compressed, and uncompressed successfully."

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

3-7

#!/bin/bash

# 폴더 이름을 입력받습니다.

read -p "Enter the directory name: " base\_dir

# 입력받은 이름으로 폴더가 없으면 생성합니다.

if [ ! -d "$base\_dir" ]; then

    mkdir "$base\_dir"

fi

# 폴더 내에 5개 이상의 파일을 생성합니다.

for i in {1..5}; do

    touch "${base\_dir}/file${i}.txt"

done

# 생성된 각 파일에 대하여 하위 폴더를 생성하고 심볼릭 링크를 만듭니다.

for file in "${base\_dir}"/\*; do

    if [ -f "$file" ]; then

        # 파일 이름으로 폴더 이름을 생성합니다.

        dir\_name="${file%.\*}\_dir"

        mkdir "$dir\_name"

        # 원본 파일의 절대 경로를 얻습니다.

        full\_file\_path=$(realpath "$file")

        # 하위 폴더에 심볼릭 링크를 생성합니다.

        ln -s "$full\_file\_path" "${dir\_name}/"

    fi

done

echo "Folders have been created and symbolic links have been established."

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

3-8

#!/bin/bash

# 파일 이름을 정의합니다.

DB\_FILE="db.txt"

# 사용자에게 입력 모드를 선택하게 합니다.

read -p "Do you want to add to the existing database or rewrite it? (add/rewrite): " mode

# 사용자가 파일을 새로 작성하길 원한다면, 기존 db.txt 파일을 삭제합니다.

if [[ "$mode" == "rewrite" ]]; then

    > "$DB\_FILE"

    echo "The database has been reset."

fi

# 사용자의 입력을 받아 파일에 추가합니다.

while true; do

    # 이름, 생일 또는 전화번호를 입력받습니다.

    read -p "Enter team member's name: " name

    read -p "Enter team member's birthday or phone number: " info

    # 입력받은 정보를 파일에 추가합니다.

    echo "$name, $info" >> "$DB\_FILE"

    # 사용자가 더 추가할지 묻습니다.

    read -p "Do you want to add another team member? (yes/no): " add\_more

    # 사용자가 더 이상 추가하지 않기를 원하면 루프를 종료합니다.

    if [[ "$add\_more" != "yes" ]]; then

        break

    fi

done

echo "Team member information has been updated in $DB\_FILE."

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

3-9

#!/bin/bash

# db.txt 파일 경로를 지정합니다.

DB\_FILE="db.txt"

# 사용자로부터 검색할 팀원의 이름을 입력받습니다.

read -p "Enter the name of the team member to search: " search\_name

# 파일에서 이름을 검색하고 해당하는 라인을 출력합니다.

# grep 명령어의 -i 옵션은 대소문자를 구분하지 않고 검색하며,

# -w 옵션은 단어 단위로 일치하는 것만 찾습니다.

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

# 결과에 따라 메시지를 출력합니다.

if [ -z "$result" ]; then

    echo "No team member found with the name: $search\_name"

else

    echo "Information for $search\_name:"

    echo "$result"

fi

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 소프트웨어, 폰트이(가) 표시된 사진

자동 생성된 설명