#!/bin/bash

# Get the path of the directory to create

path="/home/user1/ips/hello"

# Create the directory

mkdir "$path"

# Print a message to indicate that the directory has been created

echo "Directory created successfully"

To run the script, you can save it as a file with a .sh extension and then run it from the command line. For example, if you save the script as create\_directory.sh, you can run it by typing the following command into a terminal:

bash create\_directory.sh

This will create the directory hello in the path /home/user1/ips.

Here is an explanation of the code in the script:

The #!/bin/bash line tells the shell that the script is written in bash.

The path="/home/user1/ips/hello" line assigns the path /home/user1/ips/hello to the variable path.

The mkdir "$path" command creates the directory called hello at the path specified by the path variable.

The echo "Directory created successfully" line prints a message to indicate that the directory has been created successfully.

To create multiple directories at different paths, you can modify the shell script to handle an array of paths and directory names. Here's an example of how to do it using a Bash script:

#!/bin/bash

# Array of paths where the directories should be created

directory\_paths=(

"/home/user1/ips"

"/home/user2/documents"

"/var/www/html"

)

# Array of directory names to create

directory\_names=("hello" "test" "newdir")

# Loop through the array and create each directory at the corresponding path

for ((i = 0; i < ${#directory\_paths[@]}; i++)); do

path="${directory\_paths[$i]}"

dir="${directory\_names[$i]}"

# Create the directory

mkdir -p "$path/$dir"

# Check if the directory was created successfully

if [ $? -eq 0 ]; then

echo "Directory '$dir' created successfully in '$path'."

else

echo "Failed to create directory '$dir' in '$path'."

fi

done

```

In this script, we use two arrays: `directory\_paths` to hold the paths where the directories should be created and `directory\_names` to hold the names of the directories to be created. The script then iterates through the arrays and creates each directory at the corresponding path using the `mkdir` command with the `-p` option.

bash create\_multiple\_directories.sh

Save the above code into a file, for example, `create\_multiple\_directories.sh`. Make sure to give execute permissions to the script using the following command:

chmod +x create\_multiple\_directories.sh

```

Then, you can run the script using:

./create\_multiple\_directories.sh

This will create the specified directories at their respective paths. The script can handle any number of paths and directory names you define in the arrays.