Shell Scripting Cheat-sheet with Example

Ashish Singh · Follow

Published in AWS in Plain English
2 min read · Jan 17, 2024

Listen Share

Basic Structure:

```
#!/bin/bash
# This is a comment

# Variables
VAR_NAME="Hello"
echo $VAR_NAME

# User input
```

```
read -p "Enter your name: " USER_NAME
echo "Hello, $USER_NAME!"

# Conditional statements
if [ condition ]; then
    # code to execute if condition is true
elif [ another_condition ]; then
    # code to execute if another_condition is true
else
    # code to execute if all conditions are false
fi

# Loops
for item in "${array[@]}"; do
    # code to execute for each item in array
done

# Functions
function my_function() {
    # code for the function
}
my_function
```

File Operations:

```
# Check if a file exists
if [ -e "file.txt" ]; then
    echo "File exists."
fi

# Check if a directory exists
if [ -d "directory" ]; then
    echo "Directory exists."
fi

# Create a file
touch new_file.txt

# Copy a file
cp old_file.txt new_location/

# Move/Rename a file
mv old_file.txt new_name.txt

# Remove/Delete a file
rm unwanted_file.txt

# List files in a directory
ls -l
```

```
# Display file content
cat file.txt
```

String Operations:

```
# Concatenate strings
STRING1="Hello"
STRING2="World"
RESULT="$STRING1 $STRING2"
echo $RESULT

# String length
LENGTH=${#STRING1}
echo "Length of STRING1: $LENGTH"

# Substring
SUBSTRING=${STRING1:0:3}
echo "Substring of STRING1: $SUBSTRING"
```

Input/Output:

```
# Read from a file line by line
while IFS= read -r line; do
    echo "Line: $line"
done < input.txt

# Redirect output to a file
echo "Hello World" > output.txt

# Append to a file
echo "New line" >> output.txt
```

Command Line Arguments:

```
# Access command line arguments
echo "First argument: $1"
echo "Second argument: $2"
```

Math Operations:

```
# Perform arithmetic operations
result=$((5 + 3))
echo "5 + 3 = $result"
```

Examples:

Example 1: Simple Script

```
#!/bin/bash
# Simple script to greet the user
read -p "Enter your name: " USER_NAME
echo "Hello, $USER_NAME!"
```

Example 2: File Processing

```
#!/bin/bash
# Script to count the number of lines in a file
file="example.txt"
if [ -e "$file" ]; then
line_count=$(wc -l < "$file")
echo "The file $file has $line_count lines."
else
echo "File not found: $file"
fi
```

This is a basic cheat sheet to get you started. Shell scripting can become quite powerful and complex as you gain experience. Make sure to refer to the documentation for more advanced features and functionalities.

In Plain English 🚀

Thank you for being a part of the In Plain English community! Before you go:

- Be sure to **clap** and **follow** the writer 👏
- Follow us: X | LinkedIn | YouTube | Discord | Newsletter