ShellDecisionMaking

April 15, 2024

1 Shell Decision Making

Shell scripting supports several decision-making statements that can be used to form complex shell programs. Here are the main ones:

1. **If Statement**: The `if` statement is used to check a condition and execute a block of code if the condition is true.

```
a=10
b=20
if [ $a -lt $b ]; then
    echo "a is less than b"
fi  # Outputs: a is less than b
```

2. **If-Else Statement**: The `if-else` statement is used to check a condition and execute one block of code if the condition is true, and another block of code if the condition is false.

```
a=10
b=20
if [ $a -gt $b ]; then
    echo "a is greater than b"
else
    echo "a is not greater than b"
fi  # Outputs: a is not greater than b
```

3. **Else-If (elif) Statement**: The `elif` statement is used to check multiple conditions and execute the first block of code where the condition is true.

```
a=10
b=20
if [ $a -gt $b ]; then
    echo "a is greater than b"
elif [ $a -lt $b ]; then
    echo "a is less than b"
else
    echo "a is equal to b"
fi  # Outputs: a is less than b
```

4. **Case Statement**: The `case` statement is used to match one variable against several values.

```
a="Hello"
case $a in
```

```
    "Hello") echo "Hello, world!";;
    "Goodbye") echo "Goodbye, world!";;
    *) echo "Unknown greeting";;
esac  # Outputs: Hello, world!
```

5. **Select Statement**: The `select` statement is used to generate a menu from a list of items.

```
select option in "Option 1" "Option 2" "Option 3"; do
    echo "You selected $option"
    break
done
```

Note: Always use spaces around the brackets in the test conditions. For example, use `[ $a -lt $b ]` instead of `[$a -lt $b]`.

2 Thank You!