**Decision Making in Groovy**

In groovy, decision-making is used to check the condition and execute the statements. if the condition is true then the true block statement is executed and if the condition is false then the false block is executed.

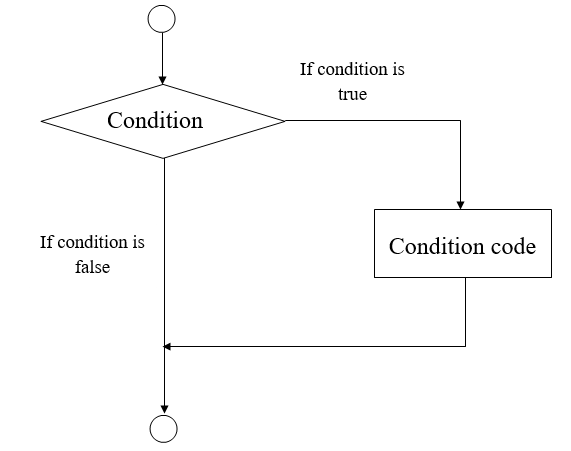
If statement

In groovy, if statement is used when we have only one condition. In if statement there is only true block. If the condition is true then the true block is executed. There is no false or else block in this statement.

Syntax:

1. **if**(condition) {
2. statement1
3. statement2
4. ...
5. }

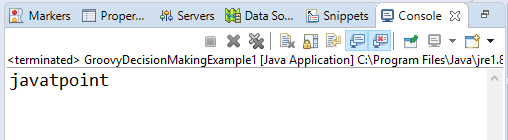
**Flow chart:**



Example 1:

1. **package** com.app
3. **class** GroovyDecisionMakingExample1 {
5. **static** **void** main(args) {
6. **int** a = 10
7. **if** (a<50) {
8. println("javatpoint");
9. }
11. }
12. }

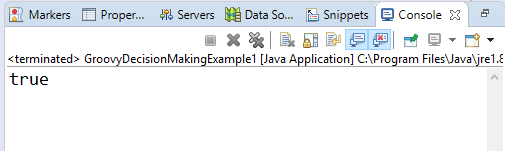
**Output:**



Example 2:

1. **package** com.app
3. **class** GroovyDecisionMakingExample2 {
5. **static** **void** main(args) {
6. def x = **false**
7. **if** ( !x ) {
8. x = **true**
9. println x
10. }
11. }
12. }

**Output:**



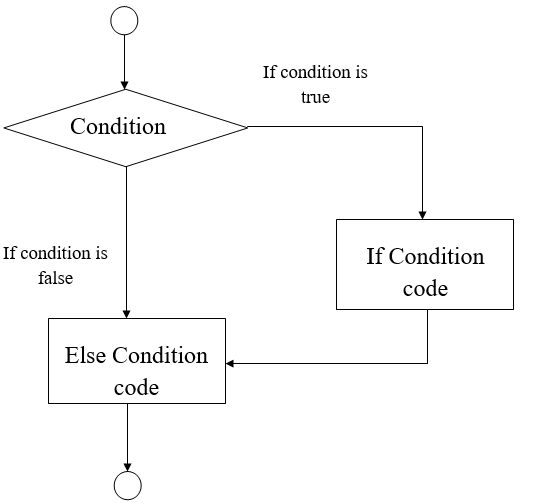
If else statement

In groovy, if else statement is used when only one condition. In if else statement we have a true as well as false or we can say else block. If the condition is true then the true block is executed otherwise else block is executed.

Syntax:

1. **if**(condition) {
2. statement #1
3. statement #2
4. ...
5. } **else**{
6. statement #3
7. statement #4
8. }

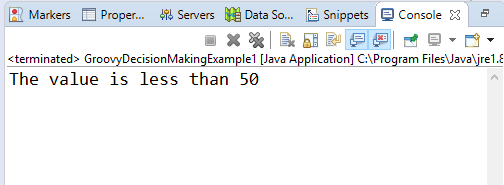
**Flow chart:**



Example 3:

1. **package** com.app
3. **class** GroovyDecisionMakingExample3 {
5. **static** **void** main(args) {
6. **int** a = 20
7. **if** (a<50) {
8. println("The value is less than 50");
9. } **else** {
10. println("The value is greater than 50");
11. }
12. }
13. }

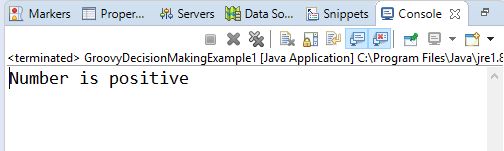
**Output:**



Example 4:

1. **package** com.app
3. **class** GroovyDecisionMakingExample4 {
5. **static** **void** main(args) {
6. **int** a = 40
7. **if** (a>0) {
8. println("Number is positive");
9. } **else** {
10. println("Number is negative");
11. }
12. }
13. }

**Output:**



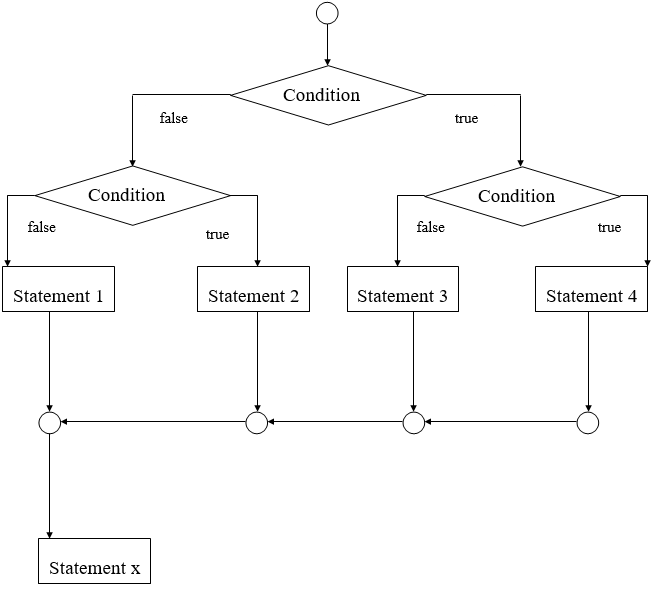
Nested If Statement

In groovy, Nested if statement is used when we have more than one condition.

Syntax:

1. **if**(condition) {
2. statement #1
3. statement #2
4. ...
5. } **else** **if**(condition) {
6. statement #3
7. statement #4
8. } **else** {
9. statement #5
10. statement #6
11. }

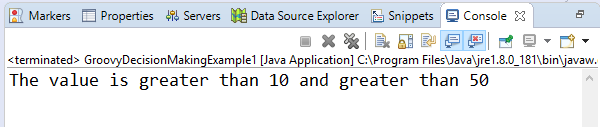
**Flow chart:**



Example 5:

1. **package** com.app
3. **class** GroovyDecisionMakingExample5 {
5. **static** **void** main(args) {
6. **int** a = 20
7. **if** (a>50) {
8. println("The value is less than 50");
9. } **else**
10. **if** (a>10) {
11. println("The value is greater than 10 and greater than 50");
12. } **else** {
13. println("The value of a is less than 10");
14. }
15. }
16. }

**Output:**



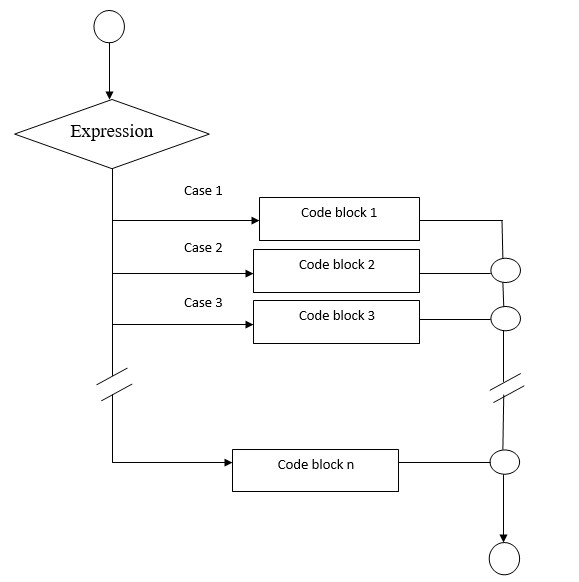
Switch Statement

In groovy, a switch is a multiway branch statement. It provides an easy way to execute different parts of the code which are based on the value of the expression.

Syntax:

1. **switch**(expression) {
2. **case** expression #1:
3. statement #1
4. ...
5. **case** expression #2:
6. statement #2
7. ...
8. **case** expression #N:
9. statement #N
10. ...
11. **default**:
12. statement #Default
13. ...
14. }

**Flow chart:**



Example 6:

1. **package** com.app
3. **class** GroovyDecisionMakingExample6 {
5. **static** **void** main(args) {
6. **int** a = 4
8. **switch**(a) {
10. **case** 1:
11. println("Monday");
12. **break**;
13. **case** 2:
14. println("Tuesday");
15. **break**;
16. **case** 3:
17. println("Wednesday");
18. **break**;
19. **case** 4:
20. println("Thursday");
21. **break**;
22. **case** 5:
23. println("Friday");
24. **break**;
25. **case** 6:
26. println("Saturday");
27. **break**;
28. **default**:
29. println("Sunday");
30. **break**;
31. }
32. }
33. }

**Output:**

