**Variable:-**

This is the name given to memory location in java, it is compulsory to specify the data type of the variable. Datatype specifies what kind of data will be stored in a variable.

dataType variableName ;

**Relational Operators**

Relational operators can be used to compare two quantities.

The relational operators are given below

Operator Meaning

< less than

<= less than or equal to

> greater than

>= greater than or equal to

== equal to

!= not equal to

**1. Less than : -** This operator is used to check if a quantity is lesser than a different quantity. The result will be a boolean type (true or false)

**2. Less than or equal to: -** This operator is used to check if a quantity is lesser than or equal to a different quantity. The result will be a boolean type (true or false).

**3. Greater than:-** This operator is used to check if a quantity is greater than a different quantity. The result will be a boolean type (true or false).

**4. Greater than or equal to:-** This operator is used to check if a quantity is greater than or equal to a different quantity. The result will be a boolean type (true or false).

**5. Equal to :-**This operator is used to check if a quantity is equal to a different quantity. The result will be a boolean type (true or false).

**6. Not equal to :-**This operator is used to check if a quantity is not equal to a different quantity. The result will be a boolean type (true or false)

**Logical Operators**

|  |  |
| --- | --- |
| **Operation** | **Meaning** |
| && | logical AND |
| || | logical OR |
| ! | logical NOT |

**Control Statements**

Java provides 3 types of Control Statements

1. Decision Making Statements

2.Loops Statements

3.Jump Statements

**1.Decision-making Statements:**

Decision-making Statements decide which statement to execute and when.

Decision-making Statements evaluate the Boolean expression and control the program flow depending upon the result of the condition provided.

a) if Statements

if

if ..else

nested if

b) switch Statements

**2. Loops Statements**

**Loops:-** A loop statement allows us to execute a statement or group of statements multiple times.

**for loop :-**

Execute a sequence of statements multiple times and abbreviates the code that manages the loop variable.

**While loop:-**

Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body.

**do...while loop :-**

Like a while statement, except that it tests the condition at the end of the loop body.

**for-each loop :-**

This loop can be used very well with iteration over arrays and other such collections.

Syntax:

for(declaration : expression)

{

//Statements

}

**3.Jump Statements**

a) break Statement

The Java break statement is used to break loop or switch statement. It breaks the current flow of the program at specified condition.

In case of inner loop, it breaks only inner loop.

b) continue Statement

The Java continue statement is used to continue the loop.

It continues the current flow of the program and skips the remaining code at the specified condition. In case of an inner loop, it continues the inner loop only.

**Arrays**

Variables are capable of storing only one value, if we want to store multiple values under one variable name we can use arrays.

An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.

**Note:-** The disadvantage of arrays is their size is fixed and they can store values of only one datatype.

1. Single Dimensional Array

2. Multi Dimensional Array

**Array List:-**

This is a dynamic data structure which is capable of expanding or contracting based on the amount of data inserted, it can also store values of different datatypes.

List<String> l=new ArrayList<>();

**Note:- List<datatype>** accepts only specified data type

**Ex: List<String>**:- Accepts only string data

**List:-** accepts any datatype

|  |  |
| --- | --- |
| **SELENIUM** | **QTP** |
| Open source | Paid Tool |
| Works on all OS (Windows, OS X, Linux, Solaris,Mac) | Works only on Windows |
| Tests only Web applications | Tests web and desktop applications |
| Works on almost all browsers(IE, Firefox, Chrome, Safari, Opera) | Works on IE |
| Code can be made in any one of languages such as Java, C#, Ruby, Python, pearl, php | VB Script |
| Object Identification Options : Name ,Id , Xpath, CSS, Link text …etc | Object Identification Options : Object properties, Repository objects |
| Selenium does not  have such built in object repository, but object can be managed by using UI element user extension like Inspect option in All browsers | HP UFT comes with built in object repository.  Object repository development and maintenance is quite easy in HP ALM |
| IDE sometimes does not record some events | Recording is a little reliable |
| Set of Libraries, around 20MB (Need to include other supporting software) | Around 1.5GB |
| No direct Support | Support From HP |