1. **Unicode System**

Unicode is a universal international standard character encoding that is capable of representing most of the world's written languages.

Why java uses Unicode System?

Before Unicode, there were many language standards:

74 - > J

* ASCII (American Standard Code for Information Interchange) for the United States.
* ISO 8859-1 for Western European Language.
* KOI-8 for Russian.
* GB18030 and BIG-5 for chinese, and so on.

This caused two problems:

1. A particular code value corresponds to different letters in the various language standards.
2. The encodings for languages with large character sets have variable length. Some common characters are encoded as single bytes, other require two or more byte.

To solve these problems, a new language standard was developed i.e. Unicode System.

In unicode, character holds 2 byte, so java also uses 2 byte for characters.

Lowest value:\u0000

Highest value:\uFFFF

1. **Operators in java**

Operator in java is a symbol that is used to perform operations. For example: +, -, \*, / etc.

There are many types of operators in java which are given below:

* Unary Operator
* Arithmetic Operator
* shift Operator
* Relational Operator
* Bitwise Operator
* Logical Operator
* Ternary Operator
* Assignment Operator

|  |  |  |
| --- | --- | --- |
| Operator Type | Category | Precedence |
| Unary | Postfix | expr++ expr-- |
| Prefix | ++expr --expr +expr -expr ~ ! |
| Arithmetic | multiplicative | \* / % |
| Additive | + - |
| Shift | Shift | << - Left Shift  >> - Right Shift  >>> |
| Relational | Comparison | < > <= >= instanceof |
| Equality | == != |
| Bitwise | bitwise AND | & |
| bitwise exclusive OR | ^ |
| bitwise inclusive OR | | |
| Logical | logical AND | && |
| logical OR | || |
| Ternary | Ternary | ? : |
| Assignment | Assignment | = += -= \*= /= %= &= ^= |= <<= >>= >>>= |

1. **Control Statements**

**Java IF Statement Syntax:**

**if**(condition){

//code to be executed

}

**Java IF-else Statement Syntax:**

**if**(condition){

//code if condition is true

}**else**{

//code if condition is false

}

**Java IF-else-if ladder Statement Syntax:**

**if**(condition1){

//code to be executed if condition1 is true

}**else** **if**(condition2){

//code to be executed if condition2 is true

}

**else** **if**(condition3){

//code to be executed if condition3 is true

}

...

**else**{

//code to be executed if all the conditions are false

}

**Java Switch Statement Syntax:**

The Java switch statement executes one statement from multiple conditions. It is like if-else-if ladder statement.

**switch**(expression){

**case** value1:

//code to be executed;

**break**; //optional

**case** value2:

//code to be executed;

**break**; //optional

......

**default**:

code to be executed **if** all cases are not matched;

}

**Java For Loop**

**for**(initialization;condition;incr/decr){

//code to be executed

}

**Java For-each Loop Syntax:**

The for-each loop is used to traverse array or collection in java. It is easier to use than simple for loop because we don't need to increment value and use subscript notation.

It works on elements basis not index. It returns element one by one in the defined variable.

**for**(Type var:array){

//code to be executed

}

**Java Labeled For Loop**

labelname:

**for**(initialization;condition;incr/decr){

//code to be executed

}

**Java Infinitive For Loop**

**for**(;;){

//code to be executed

}

**Java While Loop**

**while**(condition){

//code to be executed

}

**Java Infinitive While Loop**

**while**(**true**){

//code to be executed

}

**Java do-while Loop**

**do**{

//code to be executed

}**while**(condition);

**Java Infinitive do-while Loop**

**do**{

//code to be executed

}**while**(**true**);

**Java Break Statement with Loop**

**public** **class** BreakExample {

**public** **static** **void** main(String[] args) {

**for** (**int** i = 1; i <= 10; i++) {

**if** (i == 5) {

**break**;

}

System.***out***.println(i);

}

}

}

**Java Break Statement with Inner Loop**

**public** **class** BreakExample2 {

**public** **static** **void** main(String[] args) {

**for** (**int** i = 1; i <= 3; i++) {

**for** (**int** j = 1; j <= 3; j++) {

**if** (i == 2 && j == 2) {

**break**;

}

System.***out***.println(i + " " + j);

}

}

}

}

**Java Switch Statement**

**switch**(expression){

**case** value1:

//code to be executed;

**break**; //optional

**case** value2:

//code to be executed;

**break**; //optional

......

**default**:

code to be executed **if** all cases are not matched;

}

E.g.

**public** **class** SwitchExample {

**public** **static** **void** main(String[] args) {

**int** number = 20;

**switch** (number) {

**case** 10:

System.***out***.println("10");

**break**;

**case** 20:

System.***out***.println("20");

**break**;

**case** 30:

System.***out***.println("30");

**break**;

**default**:

System.***out***.println("Not in 10, 20 or 30");

}

}

}

**Types of Java Comments**

There are 2 types of comments in java.

Single Line Comment

// This is single line comment

Multi Line Comment

/\* This is

Multi line

Comment \*/