**What is the difference in creating an object as literal and normal object**

They both do the same thing, other than that the second one creates an object and adds a property to it. But literal notation takes less space in the source code. It's clearly recognizable as to what is happening, so using new object(), you are really just typing more and (in theory, if not optimized out by the JavaScript engine) doing an unnecessary function call.

These

person = new Object()

-or-

person = {

property1 : "Hello"

};

technically do not do the same thing. The first just creates an object. The second creates one and assigns a property. For the first one to be the same you then need a second step to create and assign the property.

**What is singletonset**

<https://www.javatpoint.com/singleton-design-pattern-in-java>

Singleton Pattern says that just**"define a class that has only one instance and provides a global point of access to it".**

In other words, a class must ensure that only single instance should be created and single object can be used by all other classes.

**What are the rules for immutability**

<https://docs.oracle.com/javase/tutorial/essential/concurrency/imstrat.html>

1. Don't provide "setter" methods — methods that modify fields or objects referred to by fields.
2. Make all fields final and private.
3. Don't allow subclasses to override methods. The simplest way to do this is to declare the class as final. A more sophisticated approach is to make the constructor private and construct instances in factory methods.
4. If the instance fields include references to mutable objects, don't allow those objects to be changed:

Don't provide methods that modify the mutable objects.

Don't share references to the mutable objects. Never store references to external, mutable objects passed to the constructor; if necessary, create copies, and store references to the copies. Similarly, create copies of your internal mutable objects when necessary to avoid returning the originals in your methods

**String Pool**

String Pool in java is a pool of Strings stored in Java Heap Memory. We know thatString is special class in java and we can create String object using new operator aswell as providing values in double quotes.String Pool is possible only because String is immutable in Java and itsimplementation of String interning concept. String pool helps in saving a lot of space for Java Runtime although it takes moretime to create the string.

**String Constant Pool**

The Java string constant pool is an area in heap memory where Java stores literal string values. The heap is an area of memory used for run-time operations. When a new variable is created and given a value, Java checks to see if that exact value exists in the pool. If not, it creates a new literal String.