	Instance Variable	**Static Variable**	**Local Variable**
Where it is Created	Inside a class but outside a method	Inside a class but outside a method	Inside a method or block
How it is Created	By declaring it inside a class but outside a method	By declaring it inside a class but outside a method with the static keyword	By declaring it inside a method or block
When it Should be Created	When you want to store state that is specific to an instance of the class	When you want to have a variable that is common to all instances of the class	When you want to store temporary state that is only needed within a certain method or block
How it is Accessed in Static Method	Cannot be accessed directly, needs an instance of the class	Can be accessed directly	N/A
How it is Accessed in Instance Method	Can be accessed directly	Can be accessed directly	N/A
How it is Accessed Within the Class	Depends upon instance or static method.	Can be accessed directly	Can be accessed directly within the method or block it is declared in
How it is Accessed Outside the Class	Can be accessed via an instance of the class, provided the variable is public	Can be accessed via the class, provided the variable is public	Cannot be accessed outside the method or block it is declared in
Description	Instance variables are non-static and are declared in a class outside any method, constructor or block	Static variables are declared similarly to instance variables but are declared using the static keyword	Local variables are declared within methods, constructors or blocks

Example	`public class Test {`
	`String instanceVar = "instance";`
	
	`}`
		`public class Test {`
		`static String staticVar =`
		`"static";`
		
		`}`
			`public class Test {`
			`public void testMethod() {`
			`String localVar = "local";`
			`}`
			`}`

Please note that the accessibility of these variables can be modified using access modifiers (private, public, protected).