Javadoc Tutorial

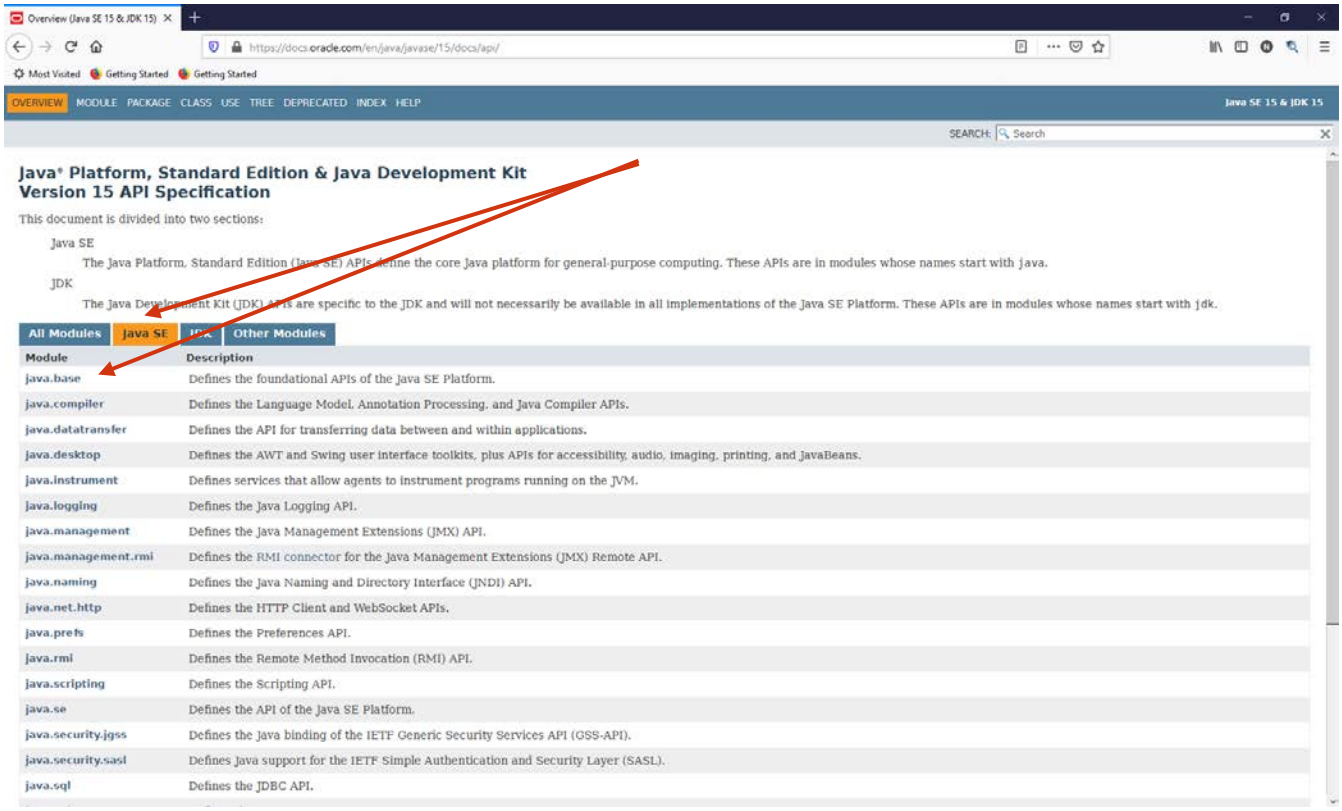
How to access and use the documentation for the Java SE Development Kit, or JDK - Versions 9 and later – for this tutorial, we will use JDK Version 15

Topic	Details
Overview	In this tutorial, you will become familiar with the basic features of the documentation for the JAVA SE Development Kit, or JDK.
Key Concepts	• Access the Javadocs API • Explore Modules • Explore Packages, Classes, Constructors, and Methods • Utilize the Search feature of Javadocs
Difficulty	Beginner – This tutorial is appropriate for someone learning Java
Duration	30 minutes
Notes	This tutorial was built using JDK Version 15

1. Access and *bookmark* this link:
    a. **Java Platform, Standard Edition & JDK Version 15 API Specification**: https://docs.oracle.com/en/java/javase/15/docs/api/index.html

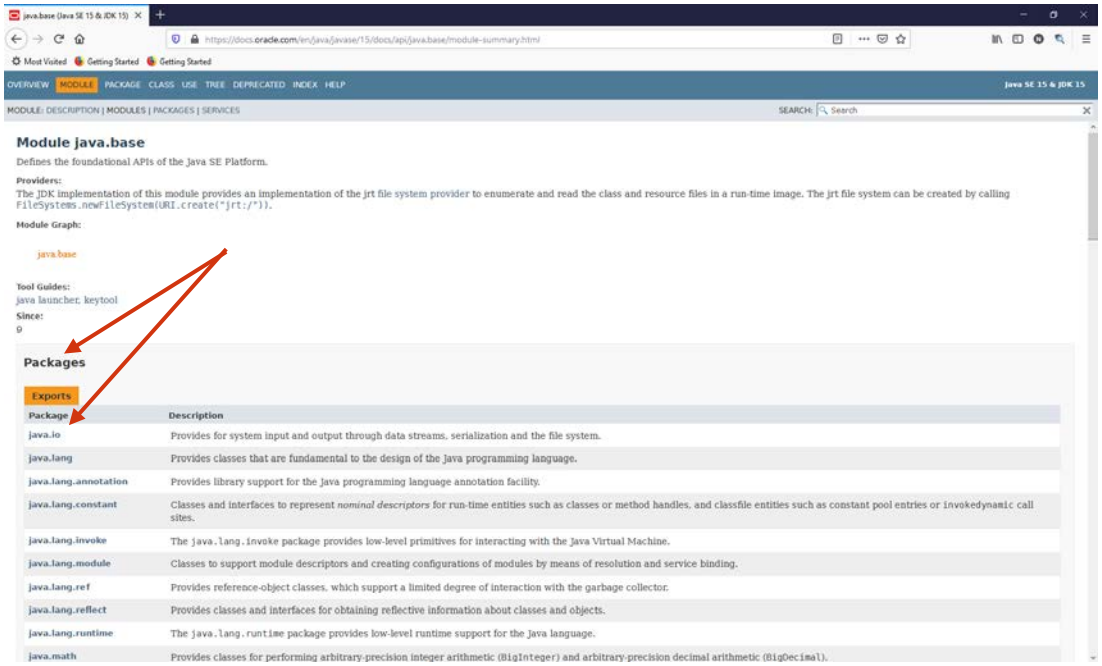
2. The documentation we will use in this course is under **Java SE** – Select the **Java SE** tab

3. The classes are grouped by module - in this course, you will be working with classes in the **java.base** module – Select the **java.base** link



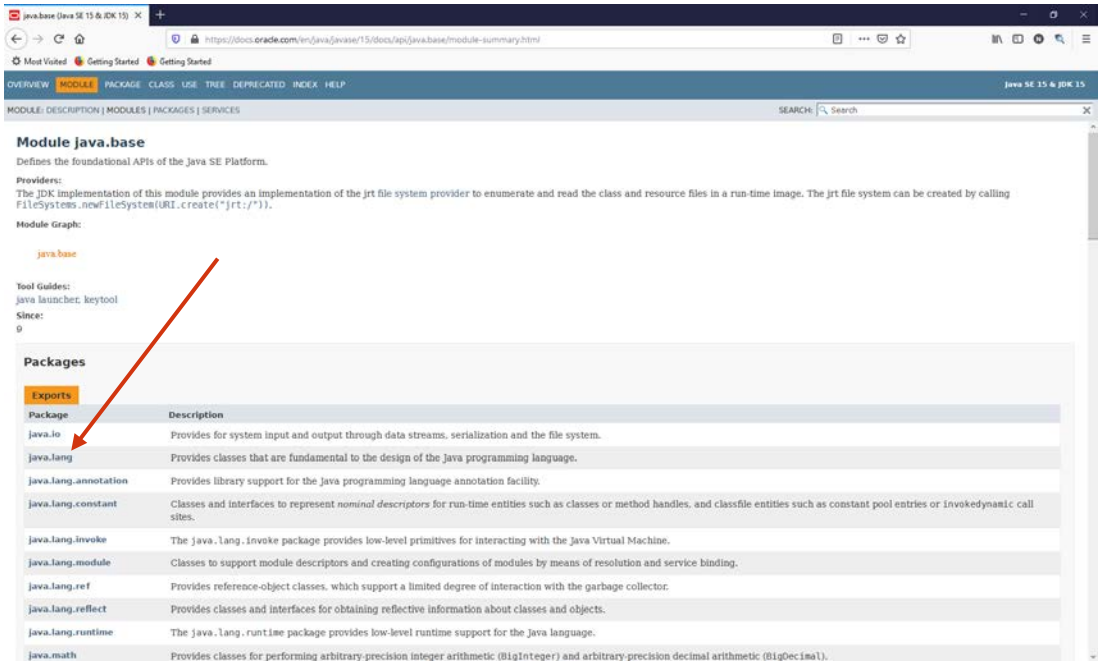
4. Within the **java.base** module is a list of packages, most of which begin with "java."

5. For example, note **java.io** that contains the classes that you use for Input and Output tasks - No need to select this



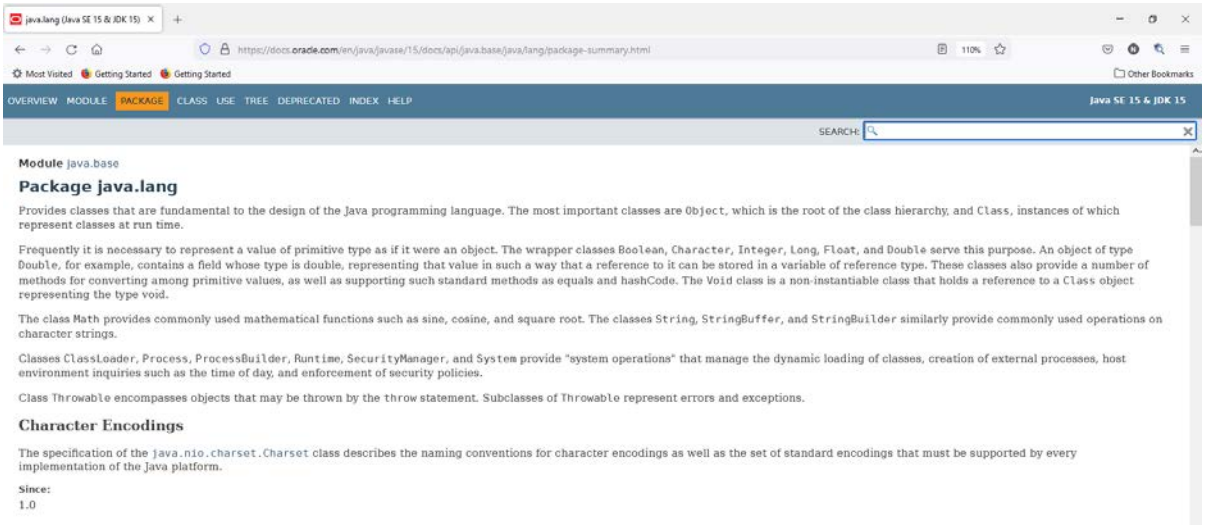
6. The **java.lang** package contains the classes that are fundamental to the Java language, and you will use these often

   a. Classes include **Boolean**, **Integer**, **Object**, and **String**



7. Select the **java.lang** package

8. The **Javadoc** for a package begins with a description of the package



9. This is followed by the **Interface Summary** for Interfaces that are included in the package

Interface Summary

Interface	Description
Appendable	An object to which char sequences and values can be appended.
AutoCloseable	An object that may hold resources (such as file or socket handles) until it is closed.
CharSequence	A CharSequence is a readable sequence of char values.
Cloneable	A class implements the Cloneable interface to indicate to the Object.clone() method that it is legal for that method to make a field-for-field copy of instances of that class.
Comparable<T>	This interface imposes a total ordering on the objects of each class that implements it.
Iterable<T>	Implementing this interface allows an object to be the target of the enhanced for statement (sometimes called the "for-each loop" statement).
ProcessHandle	ProcessHandle identifies and provides control of native processes.
ProcessHandle.Info	Information snapshot about the process.
Readable	A Readable is a source of characters.
Runnable	The Runnable interface should be implemented by any class whose instances are intended to be executed by a thread.
StackWalker.StackFrame	A StackFrame object represents a method invocation returned by StackWalker.
System.Logger	System.Logger instances log messages that will be routed to the underlying logging framework the LoggerFinder uses.
Thread.UncaughtExceptionHandler	Interface for handlers invoked when a Thread abruptly terminates due to an uncaught exception.

10. This is followed by the **Class Summary** – a list of classes that are included in the package

Class Summary

Class	Description
Boolean	The Boolean class wraps a value of the primitive type boolean in an object.
Byte	The Byte class wraps a value of primitive type byte in an object.
Character	The Character class wraps a value of the primitive type char in an object.
Character.Subset	Instances of this class represent particular subsets of the Unicode character set.
Character.UnicodeBlock	A family of character subsets representing the character blocks in the Unicode specification.
Class<T>	Instances of the class Class represent classes and interfaces in a running Java application.
ClassLoader	A class loader is an object that is responsible for loading classes.
ClassValue<T>	Lazily associate a computed value with (potentially) every type.
Compiler	Deprecated, for removal: This API element is subject to removal in a future version. *JIT compilers and their technologies vary too widely to be controlled effectively by a standardized interface.*
Double	The Double class wraps a value of the primitive type double in an object.
Enum<E extends Enum<E>>	This is the common base class of all Java language enumeration types.
Enum.EnumDesc<E extends	A nominal descriptor for an enum constant

11. Finally followed by enumerations, and exceptions that are included in the package (You will learn about these later in the course)

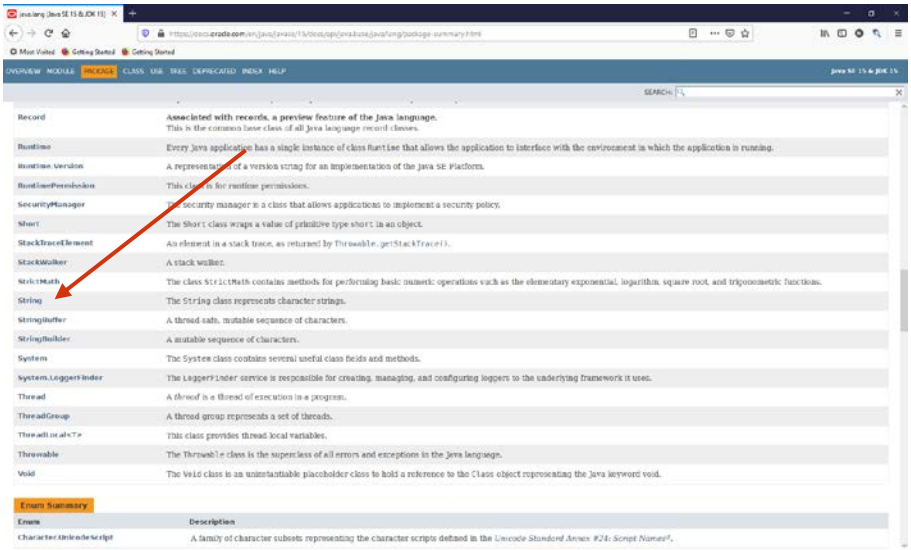
Enum Summary

Enum	Description
Character.UnicodeScript	A family of character subsets representing the character scripts defined in the *Unicode Standard Annex #24: Script Names*.
ProcessBuilder.Redirect.Type	The type of a ProcessBuilder.Redirect.
StackWalker.Option	Stack walker option to configure the stack frame information obtained by a StackWalker.
System.Logger.Level	System loggers levels.
Thread.State	A thread state.

Exception Summary

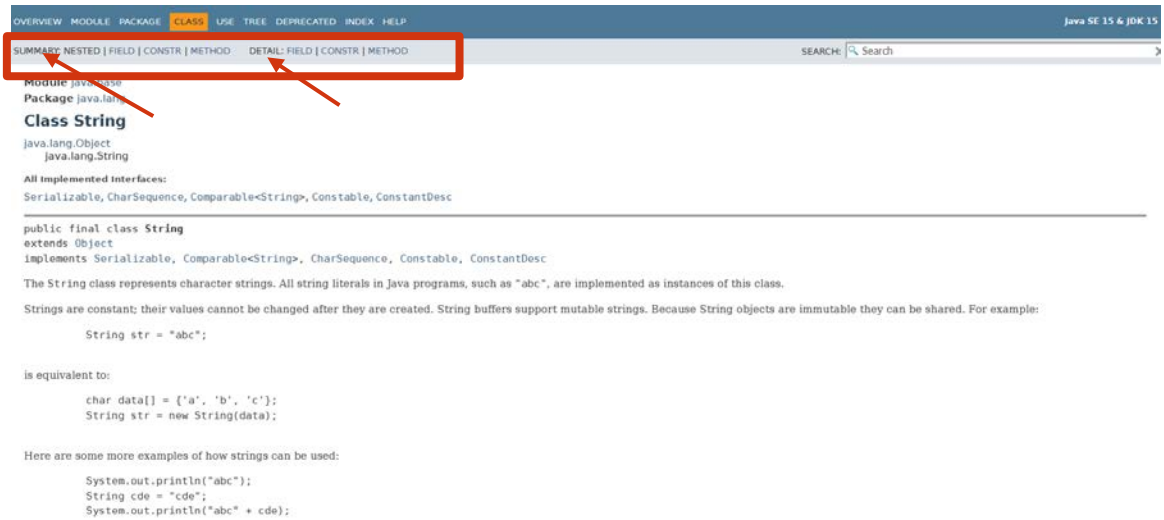
Exception	Description
ArithmeticException	Thrown when an exceptional arithmetic condition has occurred.
ArrayIndexOutOfBoundsException	Thrown to indicate that an array has been accessed with an illegal index.
ArrayStoreException	Thrown to indicate that an attempt has been made to store the wrong type of object into an array of objects.
ClassCastException	Thrown to indicate that the code has attempted to cast an object to a subclass of which it is not an instance.
ClassNotFoundException	Thrown when an application tries to load in a class through its string name using: The forName method in class Class.
CloneNotSupportedException	Thrown to indicate that the clone method in class Object has been called to clone an object, but that the object's class does not implement the Cloneable interface.

12. Under the Class Summary, select the **String** class

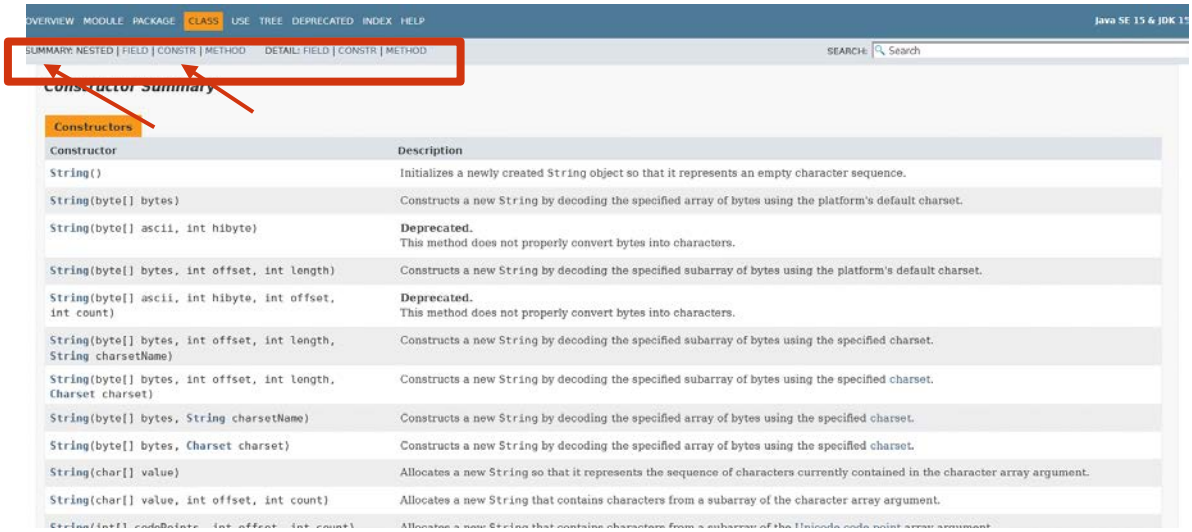


13. Let's look at the features that you will work with the most often

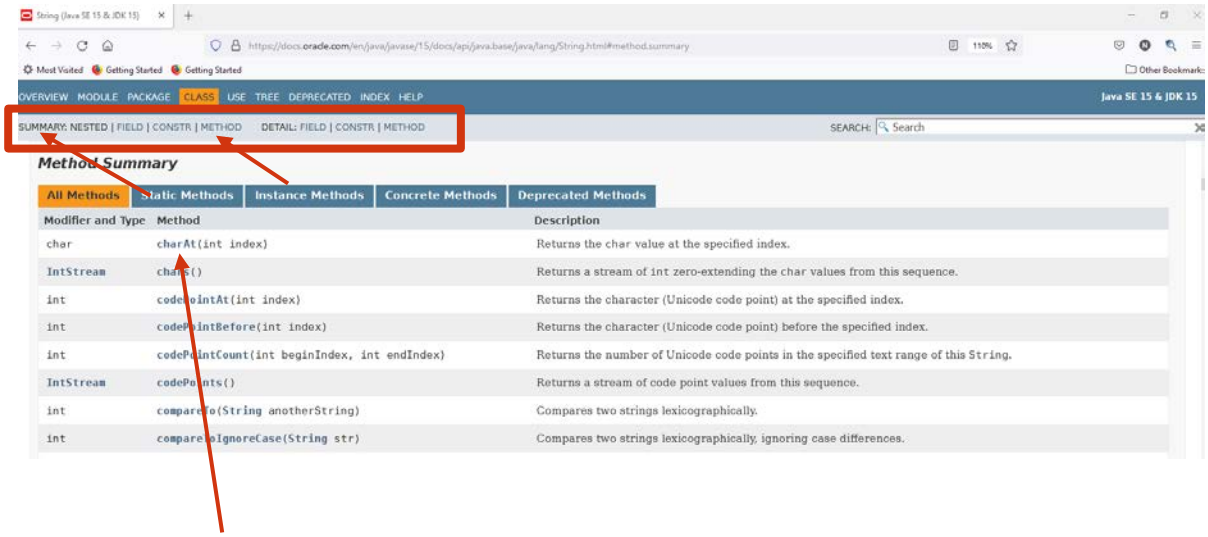
14. In the blue ribbon at the top of the page you see **SUMMARY** and **DETAIL –** these links provide simpler navigation for a class



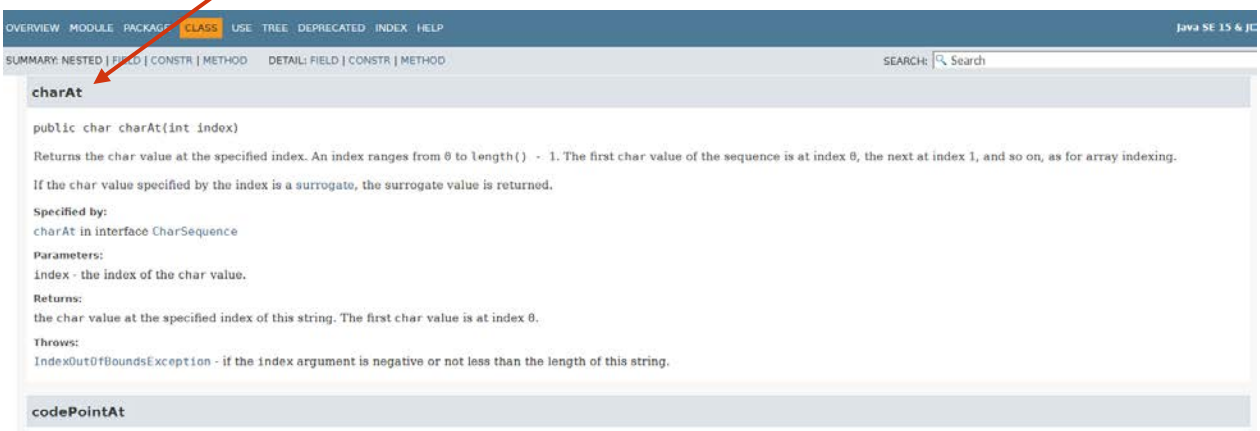
15. To view a **Summary** of the **String** class constructors, click the **CONSTR** link next to **SUMMARY**



16. Constructors are used to create objects, and this list shows many ways that you can construct a **String**
17. You will learn more about constructors later in the course
18. To see a summary of the String class's methods, click the **METHOD** link next to **SUMMARY**



19. Click on a method in the **Method Summary** to see that method's details, including a description of the method, its parameters, its return value, and so on (You can also select the **METHOD** link next to the **DETAIL** to view the method's details)

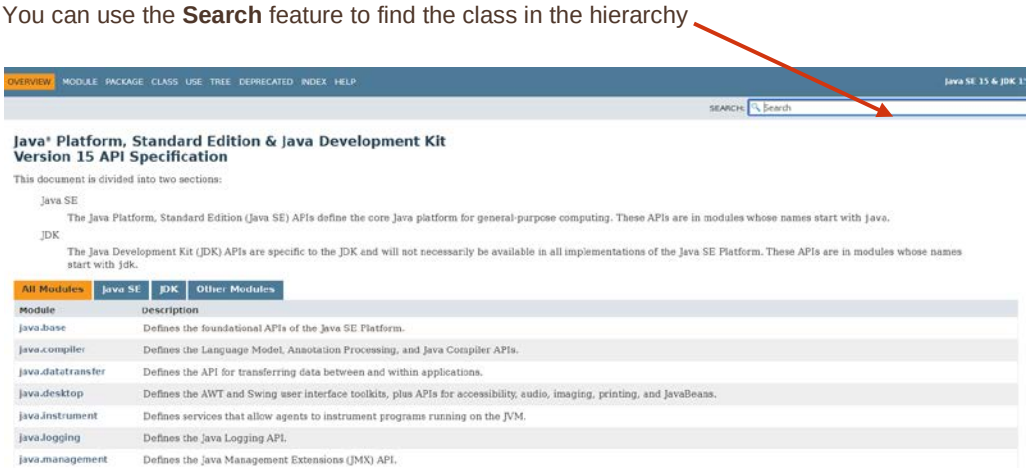


20. Methods are how you tell the object to do something, and you will learn how to use them later in the course

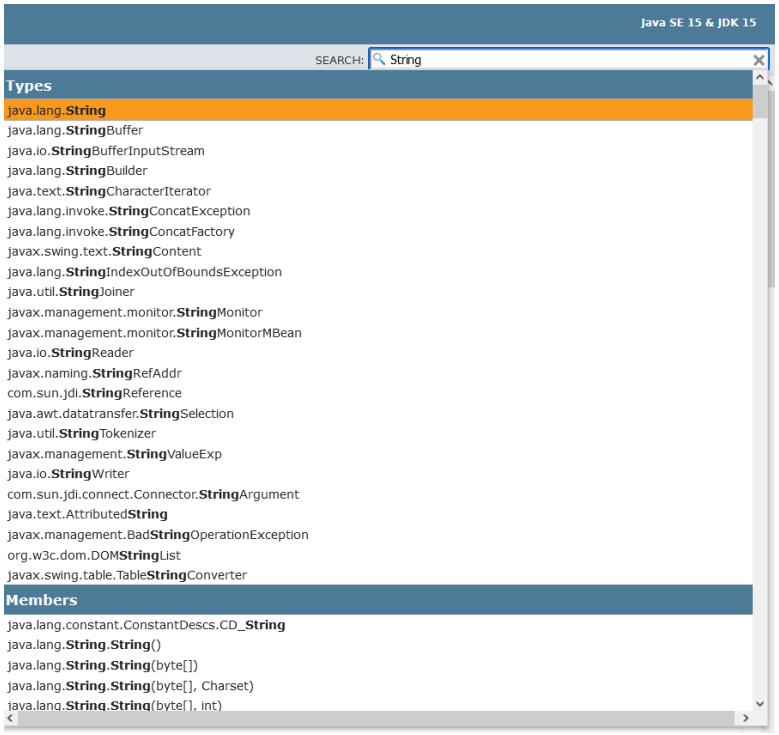
21. For now, just start to familiarize yourself with what is in the documentation

22. But what if you know the name of a class, but do not know what module or package that it is in??

23. You can use the **Search** feature to find the class in the hierarchy



24. Try this for the **String** class – note that there are MANY entries that include the word **String -** this is common for search results – you will learn as you become more familiar with the **Java API** which classes you require for your programming projects



25. As a Java programmer, the **JavaDoc** is an incredibly important tool to have in your toolbox - You will find yourself referring to the **Javadoc** again and again!

26. Again, be sure to bookmark the main documentation page!