Iterator vs Enumeration

Between Enumeration and Iterator, Enumeration is older, and it's there from JDK1.0, while iterator was introduced later. Iterator can be used with ArrayList, HashSet, and other collection classes. Another similarity between Iterator and Enumeration in Java is that the *functionality of the Enumeration interface is duplicated by the Iterator interface*.

The only major difference between Enumeration and iterator is Iterator has a `remove()` method while Enumeration doesn't. Enumeration acts as a Read-only interface because it has the methods only to traverse and fetch the objects, whereas by using Iterator we can manipulate the objects by adding and removing the objects from collection e.g. ArrayList.

Also, the **Iterator** is more secure and safe as compared to **Enumeration** because it does not allow other threads to modify the collection object while some thread is iterating over it and throws **ConcurrentModificationException.** This is by far the most important fact for me for deciding between Iterator vs Enumeration in Java.

In Summary, both Enumeration and Iterator will give successive elements, but Iterator is the new and improved version where method names are shorter and has a new method called remove. Here is a short comparison: