Difference between Predicate and Function Interface

Before we get into the difference between Predicate<T> and Function<T, R> interface I would highly recommend you reading about Predicate<T> and Function<T, R> interface. Both of the interfaces are Functional Interfaces.

Ok, let us now see the difference between them.

Predicate<T> take input of type T and returns a Boolean result (boolean test(T t)). Essentially used as decision for some type T. filter() method take Predicate as input to filter out the collection using a test provided in Predicate.

Function<T, R> interface takes parameter of type T and returns the result of type R. This broadens the usage of Function rather than Predicate as Predicate always returns boolean. Function can be used anywhere when we want to transform an input to another value.

That’s all on difference between predicate and Function interface.