Completable future is used for non blocking. See this example

**public** Future<String> **calculateAsync**() **throws** InterruptedException {

CompletableFuture<String> completableFuture = **new** **CompletableFuture**<>(); Executors.newCachedThreadPool().submit(() -> { Thread.sleep(500); completableFuture.complete("Hello"); **return** null; }); **return** completableFuture;

}

When we call

CompletableFuture<String> completableFuture = **calculateAsync**();

We will get a CompletableFuture object. But it is not sure it will hold the value.

But when we call completableFuture.get() it will block to get “Hello” if needed.