**Issue with Future**

1. Imagine you have many threads in a pool (Ex: 20 web site content downloading in 20 threads).
2. We want to start processing contents of web sites as soon as any one content arrive, no matter which one is first.
3. But here's the problem: after submitting all the tasks, Future waits on the first contentFuture.get() .
4. In worst case scenario, if the first submitted page is slower, all the 19 results except the first one are ready for processing future has to wait on first result.
5. ExecutorCompletionService allows to provide a callback interface which is called once a task is completed. So we can return whatever the thread completed.

<https://dzone.com/articles/executorcompletionservice>

**CompletableFuture**

CompletableFuture extends the functionality of the Future interface with the possibility to notify callbacks once a task is done. This callback can be executed in another thread as the thread in which the CompletableFuture is executed.