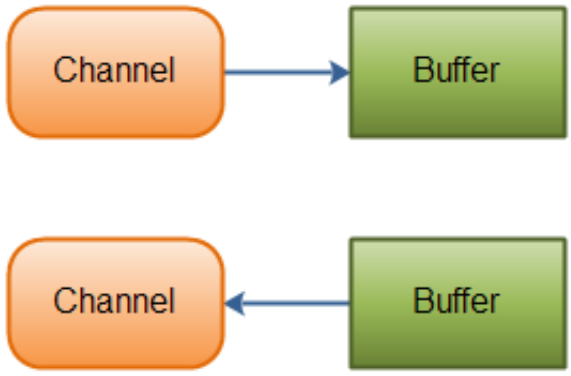
http://wiki.jikexueyuan.com/project/java-nio-zh/java-nio-channel.html

Java NIO Channel通道和流非常相似，主要有以下几点区别：

- 通道可以度也可以写，流一般来说是单向的（只能读或者写）。
- 通道可以异步读写。
- 通道总是基于缓冲区Buffer来读写。

正如上面提到的，我们可以从通道中读取数据到buffer；也可以中把buffer的数据写入到通道中。下面有个示意图：



Java NIO: Channels read data into Buffers, and Buffers write data into Channels

Channel的实现（Channel Implementations）

下面列出Java NIO中最重要的集中Channel的实现：

- FileChannel
- DatagramChannel
- SocketChannel
- ServerSocketChannel

FileChannel用于文件的数据读写。 DatagramChannel用于UDP的数据读写。 SocketChannel用于TCP的数据读写。 ServerSocketChannel允许我们监听TCP链接请求，每个请求会创建会一个SocketChannel.

Channel的基础示例（Basic Channel Example）

这有一个利用FileChannel读取数据到Buffer的例子：

```
RandomAccessFile aFile
        = new RandomAccessFile("data/nio-data.txt", "rw");

FileChannel inChannel = aFile.getChannel();
ByteBuffer buf = ByteBuffer.allocate(48);
int bytesRead = inChannel.read(buf);
while (bytesRead != -1) {
  System.out.println("Read " + bytesRead);
  buf.flip();
  while(buf.hasRemaining()){
      System.out.print((char) buf.get());
  }
  buf.clear();
  bytesRead = inChannel.read(buf);
}
aFile.close();
```

注意buf.flip()的调用。首先把数据读取到Buffer中，然后调用flip()方法。接着再把数据读取出来。在后续的章节中我们还会讲解先关知识。