*Explain what is meant by the stream abstraction. What is the relationship between streams and the observer pattern?  What are streams useful for modelling and when might you use them in Rich Web development?*

Streams are an abstraction used to model asynchronous data sources. They are a powerful technique when processing data if you don’t know the size or when it will arrive into your application. Examples of these include video data and log files. Processing these streams must be done in time-separated chunks in sequence or concurrently.

Streams and observer patterns have a particular relationship, whereby streams implement these observer patterns where data is extracted using the subscribe operation.

In Rich Web development, Streams can be useful when loading certain data. A good example of when you might use them in web development is when using subscribe operations. Such as if you were making changes to some code, the subscribe element will allow the events to occur, but only if new changes have been made.

*Assume that you are building an interface to an API in your Rich Web App. Describe in detail how you could use the RxJS library to handle asynchronous network responses to API requests. In your opinion, what are the benefits to using a streams library for networking over, say, promises? And what do you think are the downsides?*

There are many benefits to using Streams over promises, the first being if the result of a HTTP request to an asynchronous operation isn’t needed anymore, the subscription of the stream allows to cancel the subscription, while a promise will call a success or failed callback even when you don’t need the notification.

When dealing with streams, there are very few downsides to working with them. The main ones being the difficulty for beginners to understand the nature of the code. Another one would be the difficulty of setting up the environment, compared to other compilers, it takes that little bit more effort to set up.