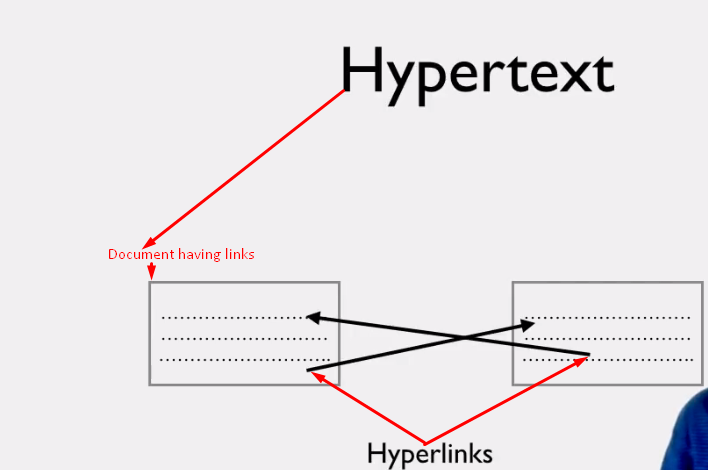
1. 
2. The concept of REST is very closely related to HTTP.
3. REST is inspired a lot by the concept of HTTP.
4.   
   So, it’s not surprise that lot of the ideas behind REST make good use of the ideas and concepts of HTTP.

So, to understand REST, you must have a little understanding of HTTP itself.

1. **HTTP**: Hyper Text Transfer Protocol. You can think of Protocol as a language or mechanism for communication. HTTP is a way to exchange data online.
   1. **Hypertext**: The stuff that you transfer/exchange in HTTP is called **hypertext** (Document having links to other documents). Hence, the name HTTP.   
      **Hyperlink**: Hypertext contains logical links to other text in the same document and other document.   
      **H**yper**t**ext **M**arkup **L**anguage: A very common and popular language in which we write hypertext having hyperlinks.
2. Let’s now move to HTTP Concepts that inspired REST instead of focusing on HTTP which is beyond the scope of this course.
   1. **Resource Locations:** REST API also has URL addresses too just like the Web page has. REST API calls return just core data and not meant for human directly.   
      The practice in RESTful API is to have **resourced-based addresses rather than action-based addresses.**
3. Next let’s talk about **HTTP Methods**. As we have decided what address is, now how to interact with it?
   * 1. HTTP has methods which are verbs which you can use to interact with URLs. 🡺 GET, POST.  
        A good RESTful API design makes right choice for HTTP method.  
        The method chosen by a developer depends on the URL’s functionality.
4. Next let’s talk about **Metadata.**
   1. **You** get not only data as response but also metadata.
      1. **HTTP Status Code: 200** (If everything is fine), 404 (If resource not found), 500 (If error on server).
      2. **Why these status codes:** As the client is a machine which when receives the response must have some way to know about the status of the response that it has got. So that, it can take some proper action.
5. Next let’s talk about the “**Format of the Message**”: As message exchange b/w the client and server can be of any of **Text, JSON, XML** etc. so how server knows about the kind of data received from the client in the request and how client knows about the format of the data in the response?  
   **Answer**: There is one header called **content-type as metadata** which is the format in which message is 🡺 **text/html, application/json**  
   **Content Negotiation:** Suppose a RESTful API is able to send XML as well as JSON formatted data as response. Depending on the client’s request for a particular content type, the WS server would send the data in that format.
6. **Summary**: We talked about the RESTful WS influenced by HTTP.
   1. Resource based URIs.
   2. Right HTTP Method.
   3. Right HTTP Status Code.
   4. Right Message headers: Such as content-type when responding back.