Servlet

* Servlet Collaboration
  + We have two ways to collaborate with our servlet:
    1. RequestDispatcher
       - It is for receiving the request of the object that defines it and the send that request to a resource such as Html, Servlet or JSP file.
       - In RequestDispatcher we have use two method to define its interface.
         * forward() method – from the term forward, it is a method that forward the request of a servlet to a resource on the side of the server.
         * include() method – include the entire resource such as Html, servlet or JSP file in response.
    2. sendRiderect() method
       - This method is use to locate another resource and implement there our response.
* Servlet – Error Handling
  + Use to able to control the error that will be appearing in the Web client.
  + We can use the syntax <error-page/> elements.
  + Different <error-page/> element.
    1. <error-code>
       - It is optional and used as a valid error code.
    2. <exception-type>
       - It is optional and use just like java exception type as an exception in HTTP.
    3. <location>
       - Use to redirect our page to a resource whenever our page responds to an error. The destination or the location may be a servlet or JSP page.
  + Different type of error a Servelet we need to throw:
    1. ServletException
    2. IOException
    3. RuntimeException
    4. Other exception that needs servletException to wrap it up.
* <https://www.javamex.com/tutorials/servlets/http_status_code.shtml>
* DDL and DML sql statements:
  + Data definition language—is use to manipulate the structure of the database.
  + Data manipulation language—use to manipulate the data in our database.

PHP

* HEADER
  + Raw HTTP header will be send.
  + Two types of header calls:
    - We have the first type that have the “HTTP/” as the beginning statement, we use this type of calls often when we want to know the status code error that will be sent.

Ex. header(“HTTP/2.0 404 Forbidden”)

* + - The second type is the “Location:” returns the status code 302 which means redirect.

Ex. header(“Location:http://www.courseweb.com”);

* + Replace
    - It is optional, it ask whether he should replace the header if the previous header is the same as the present header or just add the header.
  + http\_response\_code
    - if the header is empty it will force the response to be in a specified value.
* Superglobal
  + - $GLOBALS
      * It references to the global scope helping us to use variable that is outside the function.
        + Ex.

$exp=”my name”

Function exmpl(){

echo $GLOBALS[“exp”];

}

* + $\_SERVER
  + $\_GET
  + $\_POST
  + $\_FILES
  + $\_COOKIE
  + $\_SESSION
  + $\_REQUEST
  + $\_ENV