

Project 1: Common Gateway Interface

The program takes four parameters; origunits, convunits, numunits, and convfactor; and attempts to convert the numunits value from origunits to convunits, multiplied by convfactor, with the result printed in green. I used an object-oriented design to handle these parameters, their error messages, and html formatting. If any of these values are not properly supplied by the user, the offending parameter is printed in red and an error message is printed in bold red.

- 1) For each missing parameter, the program prints "<parameter> is missing" in red bold font (<parameter> is replaced with the actual name of the parameter).
- 2) The convunits parameter is printed in red, and the following error message is printed: "Conversion Units is missing".
- 3) The offending units parameter is printed in red, and the following error message is printed: "conversion not supported".
- 4) The following error message is printed: "Value to Convert is not a floating-point number".
- 5) The extra parameter is ignored. The conversion proceeds if the four expected parameters are properly supplied.

Bonus) GET causes the parameters and values to be stored in the browser history, because the parameters are part of the URL. POST does not do this. Also, since GET requires sending the parameters as part of the URL, the number/size of parameters that can be sent is finite (depending on the server); POST doesn't have this limitation.