Data Validation Algorithms

An algorithm is a set of steps that can be followed to solve a problem or build something. They are usually written as a generic description of how to solve a problem. They are not written in a coding language or for a specific coding language. They are the pattern that is common to certain processes and problem solutions.

They are often used to describe the various steps or processes required to get to a solution. They are best used to guide you in what steps you will need in your code and in what order they will be performed. They are guidelines rather than absolute. You will need to fill in the details specific to your project and your language.

You can think of them as the general instructions to follow when working on that type of task or project.

The following are some basic algorithms used when performing data validation. These algorithms or instructions can be used for multiple languages. In the field of Web Development we would implement these algorithms with Javascript, PHP, SQL or other server side languages.

Form Validation Algorithm

This algorithm is the general set of steps that you would use when validating the data that comes from the fields on the form. This is often done in javascript on the client and PHP on the server. The following algorithm assumes that you are using a self-posting form in order to properly display and validate the form and its data.

if the form has been submitted (The user has filled out the form and hit the submit button)

{ The form data is ready to be validated

set the validForm flag to true (The flag tells us if the form is still valid after all the tests)

…

*Field validation tests go in this area*

…

if (validForm==true) If the flag is still true no errors were found, the form is valid

{

process form data into database The form data is good so process it

}

else The flag is false because errors were found in the data {

load the original data back into the form so the user can see their original form values.

load the error messages on the form so the user knows what to fix

display the form to the user with the original data and any error messages.

}

}

Else (The user did not submit the form, they need to see the form to enter their data)

{

display the form so the user can enter their data

}

Form Field Validation Algorithm

This algorithm or process is used for each field that needs validated on your form. These validations are often placed within a function. All the code needed to validate a field would be in that function. For example a function called validateName() would do all the validation for the Name field. This improves the readability and portability of your validations. A typical field validation follows the following steps.

clear the error messages for this validation. Set error message variable to ""

(Cleans up from previous errors and assumes there will not be an error)

check the variable for the field against the expected values of the field

(Use if statement and any specific functions or tools of the language in use)

if it meets those values

{

the field is valid

nothing else needs done

}

Else the field contains invalid data

{

the field is invalid

set the validForm=false (Set the validForm flag to false)

(A data validation error has been found. The form is no longer valid)

set the error message variable for this field to the appropriate message

(The user knows what is wrong and where)

}