# Assignment06 – Job Application for Cheesy Pete’s

Business is booming at Cheesy Pete’s. They need to hire some staff. That's why this job application form was created.

In this assignment you will write JavaScript to validate data provided on this form. You will use your knowledge of functions, strings and numbers to write this code.

## Overview

Important notes:

* What we are doing in this assignment is called client-side form validation, or just “input validation”. We are making sure that the data, provided by the user, on the client, conforms to our expectations before we send it off to “the cloud”.
* You won't go beyond form validation in this assignment. That is, your data won't \*really\* be sent to “the cloud”. It will just look that way.
* If you use Google to research the topic of form validation, you might discover that there is a whole language dedicated to this kind of task.  This language is called "Regular Expressions" or RegEx. ***Do not use RegEx for this assignment.*** For the sake of your learning, it is best that you stick to the limited subset of string methods presented in class. To be clear: students must use the subset of string functions covered in class this week.

Here is the form.

A white form with black text

AI-generated content may be incorrect.

## Instructions

1. Download assignment06.zip as provided by your instructor.
2. In the pizzahire.html file provided complete the following functions.

|  |  |
| --- | --- |
| **Function name** | **Comments** |
| validPizzaText | The validPizzaText function accepts one parameter. If the parameter has ***at least one*** visible piece of text in it, the validPizzaText function should return true. If the parameter is empty or all spaces, it should return false.  For example:  validPizzaText('') //returns false validPizzaText('x') //returns true validPizzaText('xyz123') //returns true validPizzaText('0') //returns true |
| validPizzaDigits | The validPizzaDigits function accepts two parameters: digits and size. The function should return true if the parameter digits contains nothing but digits, and has a length equal to size. Otherwise the function should return false.  For example: validPizzaDigits('321',3) //returns true  validPizzaDigits('321',4) //returns false   //because '123'   //has a length of 3  validPizzaDigits('99',3) //returns false  validPizzaDigits('',3) //returns false  validPizzaDigits('xyz',3) //returns false  validPizzaDigits('0123456789',10) //true |
| ValidPizzaEmail | The validPizzaEmail function accepts one parameter. If the parameter “looks like” and email address then the function should return true. If not, it should return false.  Students should implement the following **(greatly simplified)** definition of a valid email address.   1. A valid email address must contain exactly one @ character. 2. A valid email address may not begin with @ or end with @. 3. A valid email address must contain at least one dot “.”. Note that some email addresses may contain more than one dot.   Examples of *valid* email addresses:   * steve@test.com * Steve.Martin@test.com * martin@mail.test.com * steve.martin@mail.test.com   Examples of *invalid* email addresses:   * stevetest.com * @stevetest.com * stevetest.com@ * steve@martin@test.com * martin@com |

1. Then... when all the functions above are completed, tested, and working ... use those functions to complete the function called formPizzaValidation.

|  |  |
| --- | --- |
| **Function name** | **Comments** |
| formPizzaValidation | The function formPizzaValidation receives first name, last name, phone, and email address as parameters. It then calls the supporting functions validPizzaText, validPizzaDigits, validPizzaEmail as necessary. The formPizzaValidation function will return one of the following strings appropriately:   * Bad first name. * Bad last name. * Bad phone. * Bad email. * Got it. Thanks.   Your formPizzaValidation function will only return one error at a time. This is true even if there is more than one thing wrong with the input. This is OK for the purposes of this assignment. |

1. Check your work.

## Turn in your work

1. Upload your work to the class server.
2. Go to canvas and upload the URL to pizzahire.html using the corresponding assignment there.