

Name: \_\_\_\_\_

Mark: \_\_\_\_\_ / 100

## Instructions

---

Previously, you created some script for form validation. For this assignment, you can use the same form, and even *some of the same JavaScript*. However, you will need to use regular expressions this time to validate the following:

- First name
- Last name
- Email Address
- Phone Number
- Postal Code
- URL
- Age

**Do not use Regular Expressions found on the Web.** Many of these may be incorrect, needlessly complex, or just do something different than what you want. Create your own RegEx, or use a combination of RegEx and other JS such as `split()`. You will receive a better mark if your RegEx is efficient and correct for the data you are trying to validate.

*Note: Many of the validations in this script will use validation **different than what you would use in “real life”**, such as an **underscore** in the postal code, **square brackets** in a phone number, or only part of an entire URL. These are added to this assignment **only so that students don’t copy/paste RegEx samples found online**. In the “real world”, you would not use an underscore in a postal code, square brackets in a phone number, or only check the first part of a URL.*

## Requirements

1. **First name and last name** validation. Be sure that there are no numbers or special characters in the first and last name.
2. **Email Address:** **You must use `split()`** to split the email address up at the "@" symbol, and the "." character, and check each part separately with regular expressions.
3. **Phone number:** Make sure to allow for square brackets, hyphens, and dots/periods. All these phone numbers should be allowed:

[204] 555-1212  
204.444.2323  
204-555-3434

Note: For this assignment, **only square brackets [] are allowed**. Conventional RegEx scripts found online will NOT work. You must create this yourself.

4. **Postal Code:** Validate a Canadian postal code (ie: R3C\_4C0) so that it follows letter, number, letter, UNDERSCORE, number, letter, number.

Note: For this assignment, you MUST use an **underscore**. Conventional RegEx scripts found online **will not work**.

5. **URL:** Make sure it starts with <http://> or <https://> **Do not check the rest of the URL.** Conventional URL checking scripts on the Web **will not work**.
6. **Age:** Make sure this data is comprised of only numerical characters.
7. **Negation:** Use at least one regular expression using negation (matching characters which are NOT in the pattern), and show the user with an alert or innerHTML at least one of those characters.
8. All the above need to use regular expressions and/or a combination of regular expressions and JavaScript string methods, to receive marks.

## Submission Guidelines

---

Upload your finished assignment to the specified OneDrive folder by the due date. If you have associated files (images, stylesheets, etc) **make sure to include these in your folder**. If you have a single HTML page, there is no need to upload a folder. Just the file.

*Code copied from other people, other sources, from course PDFs, or copied from in-class examples (**unless otherwise specified**) is plagiarism. Please refer to the “Academic Misconduct” section of your course outline.*

Folder/File name: “21129\_WDW01\_Assignment2(lastname\_Firstname)”

### Due Date and Time:

Wednesday, February 16, 2022 at 09:00 a.m.

## Marking Criteria

---

Marks for this project will be given as follows:

Criteria	Needs Work	Proficient	Excellent	Mark
----------	------------	------------	-----------	------

<b>First Name</b>	No RegEx for first name found.	RegEx is overly complex, or doesn't work properly.	First name validation as required using regex.	
	0	1-9	10	/ 10
Marking Comments				
<b>Last Name</b>	No regex for last name found.	RegEx is overly complicated, or doesn't work properly.	Last name validation as required using regex.	
	0	1-9	10	/ 10
Marking Comments				
<b>Email Address</b>	No regex for email address found.	Email address uses a single regex (no splitting) (5 marks)  Email address regex is overly complex or doesn't work properly. (1-19 marks)	Email address is split up as required, with regular expressions checking each part.	
	0	1-19	20	/ 20
Marking Comments				
<b>Phone number</b>	No regex for phone number found or <b>RegEx doesn't check for square brackets.</b>	RegEx is overly complicated, or doesn't work properly.	Phone number validation as required using regex.	
	0	1-9	10	/ 10
Marking Comments				
<b>Postal Code</b>	No regex for postal code found, or <b>RegEx doesn't check for the underscore.</b>	RegEx is overly complicated, or doesn't work properly.	Postal code validation as required, using regex.	
	0	1-9	10	/ 10
Marking Comments				

<b>URL</b>	No regex for URL found, or <b>RegEx</b> checks the rest of the URL.	RegEx is overly complicated, or doesn't work properly.	URL validation as required using regex. Check the protocol only.	
	0	1-9	10	/ 10
Marking Comments				
<b>Age</b>	No regex for age found.	RegEx is overly complicated, or doesn't work properly.	Age validation as required, using regex.	
	0	1-9	10	/ 10
Marking Comments				
<b>Negation</b>	No negation regex found.	Negation found, but incorrect characters are not shown to the user. (10 marks)  RegEx is overly complicated, or doesn't work properly.	At least one negation regex used, with feedback given to the user about which character is not allowed.	
	0	1-19	20	/ 20
Marking Comments				
			<b>Total Marks</b>	<b>/ 100</b>

Additional Comments	
---------------------	--