# Problem 04 – Event handling with JQuery

# Session 2 Worksheet

**Exercise 3: Regular Expressions**

#### Regular Expressions are used to find certain patterns of text, for example, finding email addresses in a long text.

|  |  |  |
| --- | --- | --- |
| **RE** | **Explanation** | **Examples of valid entries** |
| **[0-5]**  **[0-5]{1}** | A digit between 0 and 5 | '3' or '1' |
| **[0-5]\*** | 0 or more digits between 0 and 5 | '321' or '' |
| **[0-5]+** | 1 or more digits between 0 and 5 | '321' or '1' |
| **[2-6]{4}** | Exactly 4 digits between 2 and 6 | '5435' or '2266' |
| **[a-z]\*** | 0 or more lowercase alphabets | 'hello' or '' or 'one' |
| **[A-Z]\*** | 0 or more uppercase alphabets | 'HELLO' or '' or 'ONE' |
| **\w{4}** | Exactly 4 characters long words | 'four' or 'LOOP' or 'Zone' |
| **\d{4,6}** | 4, 5 or 6 digits | '1243' or '663944' or '53521' |
| **[A-Z]{2}[0-9]{4}** | Two uppercase alphabets followed by 4 digits | 'AB2231' or 'GG2400' |
| **.+** | 1 or more digits or characters | 'a' or '9' or 'axy' or 'a1h' or 'x456' |

# Resource: <http://www.w3schools.com/jsref/jsref_obj_regexp.asp>

<https://www.youtube.com/watch?v=DRR9fOXkfRE> [11:52] (REGEX Tutorial Regular Expressions)

<https://www.youtube.com/watch?v=y4e9CYqtjMI> [12:17] (Tools for writing regular expressions)

Write down the regular expressions for the following formats and test it using Expresso Tool: <http://www.ultrapico.com/expresso.htm>

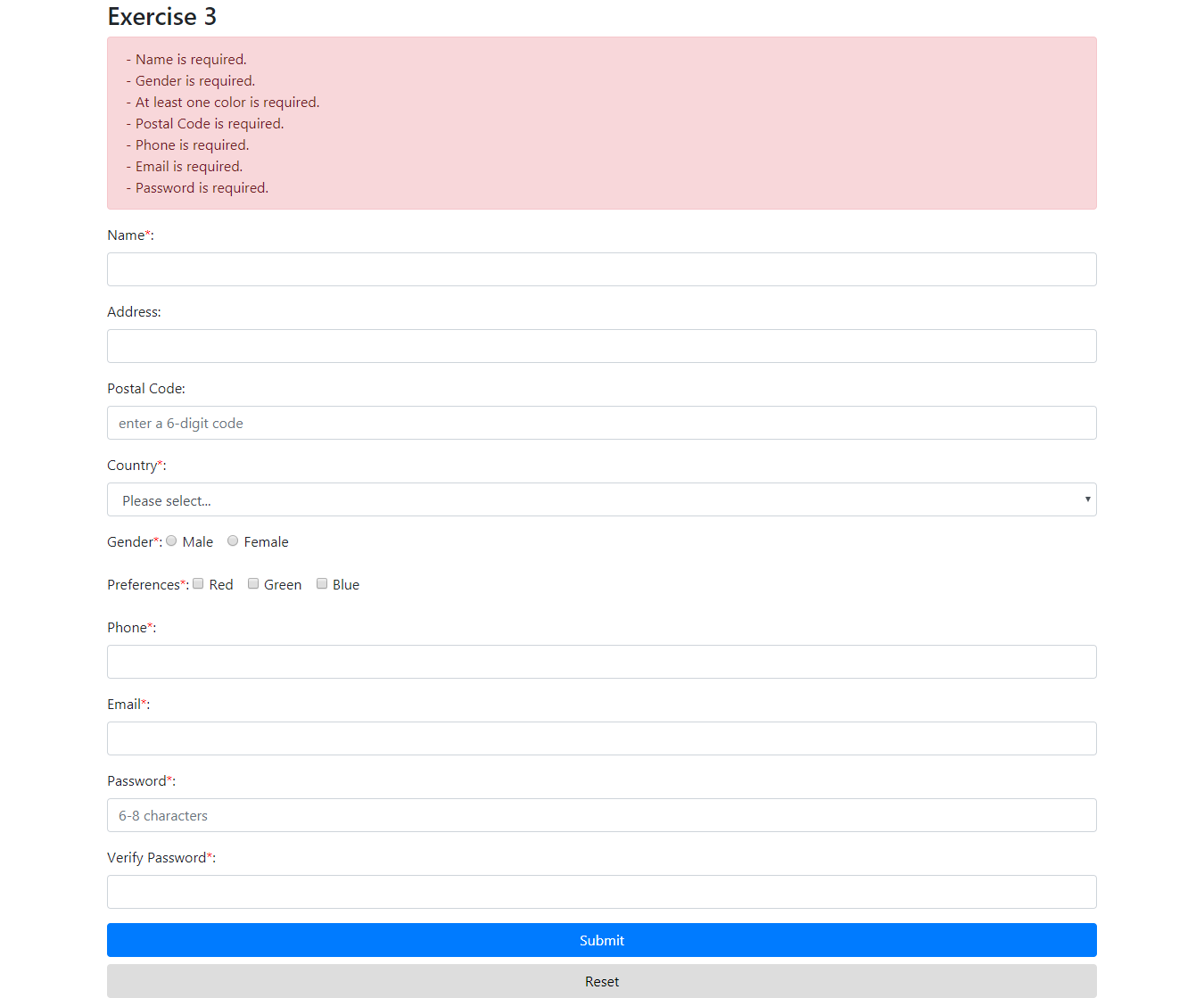
|  |  |
| --- | --- |
| **Requirement** | **Regular Expression** |
| Singapore mobile telephone number | (\+65\s)?[89]\d{7} |
| Singapore residential telephone number | (\+65\s)?6\d{7} |
| Singapore NRIC | [STFGstfg]\d{9}[A-Za-z] |
| A password that is between 6 and 12 characters and must contain a combination of letters, numbers only. E.g. “Apple32”, “MissionImpo1”, “ABC12345” | [a-Za-z0-9]{9,12} |
| A username which starts with either “T” or “G” followed by three lowercase letters. E.g. “Tabc”, “Gxyz”, “Trrr” | [TG][a-z]{3} |
| A student ID that contains 10 numbers. E.g. “0123456789”, “1111111111”, “3232323232” | [0-9]{10} |
| A Smart Watch Model code that starts with 3 digits followed by a dash and 4 digits. E.g. “993-2202”, “221-8989”, “334-7777” | [0-9]{3}-[0-9]{4} |
| All image files whose file names contain letters, digits and underscores and end with either “jpg”, “gif” or “png”. E.g. “abc\_123.jpg”, “photo1.gif”, “IMG2016\_01\_01.png” | \w+\.(jpg|gif|png) |
| Write the regular expression to validate a string of 4 to 8 numbers which must end with the number 6 or 8 E.g. “1238”, “543216”, “33445566”. | \w{3,8}[68] |
| Write the regular expression to validate a username with 5 alphabets followed by any number of numeric digits. E.g. “usr1”, “usr23”, “usr9999”. | [A-Za-z]{5}(usr1|usr23|usr9999) |
| Write the regular expression to validate a currency display text which starts with the dollar sign “$” followed by 2 numbers, then a dot (.) and ending with 2 numbers. E.g. “$10.89”, “$23.23”, “$77.10”. | \$\d{2}\.\d{2} |
| Write the regular expression to validate a string of minimum 6 characters which starts with an uppercase letter followed by lowercase letters or digits or dashes (-). For example, “A12f-5”, “Rd3216tjo”, “Z-hap9876543”. | [A-Z][a-z0-9-]{5,} |
| Write the regular expression to validate a 8 character long password that starts with “20” or “30” followed by any characters. E.g. “20$^erty”, “3002-s.s”, “20123456”. | (20|30).{6} |

# Exercise 3: Validating a form with regular expressions

Open exercise3.html and create a new JavaScript file exercise3.js. Link both files by making sure that this line is in exercise3.html.



When the Submit button is clicked, the form page will check for empty fields and display error messages on top of the form based on certain validation rules.



Write the jQuery code to validate the following fields and display the error message on top of the form.

Validation rules

* “Name”, “Country”, “Gender”, “Preferences”, “Phone”, “Email”, “Password” and “Verify Password” fields are mandatory
* Valid Postal Code format (6 digits)
* Valid Email format
* Valid password format (6 to 8 alphabets and/or digits (regardless of casing))

# The code below in exercise3.js shows you how to check if the postal code is empty and if it is in a valid postal code format.



Line 2 – Hide the red error box upon page load.

Line 3 – Create a submit event listener for the form

Line 5 – Create and initialise a “message” variable to store all error messages.

Line 7 – $("[name=postalcode]") retrieves the element of name “postalcode”, val() method is used to retrieve the value of the form field

Line 10 – test() method is a JavaScript method to check a value against a regular expression which is [0-9]{6} (6 digits). Regular expressions are enclosed with **/^** and **$/** symbols

Line 14 – Check if there are any errors by checking if the length of the string variable “message” is more than zero

Line 15 – Display the error message by using the html() method on the selector #alertBox

Line 16 – Display the red error box by using the show() method.

Line 18 – Prevent the form from submitting

Run exercise3.html to test out the desired effect.

# Exercise 3a: Validating all required text fields

In the same exercise3.js, type the jQuery code to make the rest of the form fields (Postal Code, Country, Phone, Email and Password) required fields. Display all error messages at the top of the form.

Copy and Paste code here

|  |
| --- |
| if (!$("#idName").val()) {message += "- Name is required.<br>";}if (!$("#selCountry").val()) {message += "- Country is required.<br>";}if(!$("#idPostal").val()) {message += "- Postal code is required.<br>";}if (!$("#idPhone").val()) {message += "- Phone is required.<br>";}if (!$("#idEmail").val()) {message += "- Email is required.<br>";}if (!$("#idPwd").val()) {message += "- Password is required.<br>";} |

# Exercise 3b: Validating the selection of radio buttons

In the same exercise3.js, type the jQuery code to make “Gender” a required field, meaning the user must select at least one gender type. Display the error message at the top of the form.

Hint: Use selector "[name=gender]:checked" and length method. If the value is zero, add the message "Please select gender" to the **message** variable

Copy and Paste code here

|  |
| --- |
| if (!$("input[type='radio']:checked").val()) {message += "- Gender is required.<br>";} |

# Exercise 3c: Validating the selection of check boxes

In the same exercise3.js, type the jQuery code to make “Preferences” a required field, meaning the user must select at least one preference. Display the error message at the top of the form.

Hint: Use selector ".chkboxlist:checked" and length method. If the value is zero, add the message "At least one color is required" to the **message** variable

Copy and Paste code here

|  |
| --- |
| if (!$("input[type='checkbox']:checked")) {message += "- At least one color is required.<br>";} |

# Exercise 3d: Validating the format of passwords

In the same exercise3.js, type the jQuery code to validate the “Password” field to ensure between 6-8 characters is entered. Display the error message at the top of the form.

Copy and Paste code here

|  |
| --- |
| else {let regex = new RegExp("/.{6,8}/g");if(!regex.test($(this).val()))message += "- Password must be 6 to 8 characters long.<br>";} |

# Exercise 3e: Validating the match of 2 text fields

In the same exercise3.js, type the jQuery code to make sure the “Verify Password” field matches what is entered in the “Password” field. Display the error message at the top of the form.

Copy and Paste code here

|  |
| --- |
| if($("#idVerifyPwd").val() != $("#idPwd").val()) {message += "- Password must be the same.<br>"} |

# Exercise 3g: Validating the format of email and phone

In the same exercise3.js, type the jQuery code to validate the “Email” and “Phone” fields. Display the error messages at the top of the form.

Copy and Paste code here

|  |
| --- |
| if (!$("#idPhone").val()) {message += "- Phone is required.<br>";} else {let regex = new RegExp("[89]\d{7}");if(!regex.test($("#idPhone").val())) {message += "- Phone number must be starting with 8 or 9 and containing a total of 8 digits.<br>";}}if (!$("#idEmail").val()) {message += "- Email is required.<br>";} else {let re = /^(([^<>()[\]\\.,;:\s@\"]+(\.[^<>()[\]\\.,;:\s@\"]+)\*)|(\".+\"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/;if(!re.test($("#idEmail").val())) {message += "- Email must be entered correctly.<br>"}} |