# Solutions-Exercise: jQuery

## 1.Increment Counter

function increment(selector){

let $container = $(selector);

let fragment = document.createDocumentFragment();

let $textAreaCounter = $('<textarea>');

let $incrementBtn = $('<button>Increment</button>');

let $addBtn = $('<button>Add</button>');

let $ulResults = $('<ul>');

$textAreaCounter.val(0);

$textAreaCounter.addClass('counter');

$textAreaCounter.attr('disabled', true);

$incrementBtn.addClass('btn');

$incrementBtn.attr('id', 'incrementBtn');

$addBtn.addClass('btn');

$addBtn.attr('id', 'addBtn');

$ulResults.addClass('results');

$($incrementBtn).on('click', incrementCounter);

$($addBtn).on('click', addToResults);

$textAreaCounter.appendTo(fragment);

$incrementBtn.appendTo(fragment);

$addBtn.appendTo(fragment);

$ulResults.appendTo(fragment);

$container.append(fragment);

function incrementCounter(){

$textAreaCounter.val(Number($textAreaCounter.val()) + 1);

}

function addToResults(){

let $li = $('<li>');

$li.text($textAreaCounter.val());

//let $li = $(`<li>${$textAreaCounter.val()}</li>`);

$ulResults.append($li);

}

}

|  |
| --- |
| function increment(selector) { |
|  | let container = $(selector); |
|  | let fragment = document.createDocumentFragment(); |
|  | let textArea = $("<textarea>"); |
|  | let incrementBtn = $("<button>Increment</button>"); |
|  | let addtBtn = $("<button>Add</button>"); |
|  | let list = $("<ul>"); |
|  |  |
|  | textArea.val(0); |
|  | textArea.addClass("counter"); |
|  | textArea.attr("disabled", true); |
|  |  |
|  | incrementBtn.addClass("btn"); |
|  | incrementBtn.attr("id", "incrementBtn"); |
|  | addtBtn.addClass("btn"); |
|  | addtBtn.attr("id", "addBtn"); |
|  |  |
|  | list.addClass("results"); |
|  |  |
|  | $(incrementBtn).on("click", function () { |
|  | textArea.val(+textArea.val() + 1); |
|  | }); |
|  | $(addtBtn).on("click", function () { |
|  | let li = $(`<li>${textArea.val()}</li>`); |
|  | li.appendTo(list); |
|  | }); |
|  |  |
|  | textArea.appendTo(fragment); |
|  | incrementBtn.appendTo(fragment); |
|  | addtBtn.appendTo(fragment); |
|  | list.appendTo(fragment); |
|  |  |
|  | container.append(fragment); |
|  | } |

|  |
| --- |
| function increment(string) { |
|  | let selector = $(string); |
|  |  |
|  | selector.append($('<textarea>').addClass('counter').attr('disabled', true).val('0')) |
|  | .append($('<button>').addClass('btn').attr('id', 'incrementBtn').text('Increment').on('click', increaseValue)) |
|  | .append($('<button>').addClass('btn').attr('id', 'addBtn').text('Add').on('click', appendToList)) |
|  | .append($('<ul>').addClass('results')); |
|  |  |
|  | function increaseValue() { |
|  | $('.counter').val(Number($('.counter').val()) + 1); |
|  | } |
|  |  |
|  | function appendToList() { |
|  | $('.results').append($('<li>').text($('.counter').val())); |
|  | } |
|  | } |

## 2.Timer

function timer(){

let $startTimerButton = $('#start-timer');

$startTimerButton.on('click', startTimer);

let $stopTimerButton = $('#stop-timer');

$stopTimerButton.on('click', stopTimer);

let isRunning = false;

let timerInterval;

let timeInSeconds = 0;

function startTimer(){

if(!isRunning){

timerInterval = setInterval(runTime, 1000);

isRunning = true;

}

}

function stopTimer(){

clearInterval(timerInterval);

isRunning = false;

}

function runTime(){

timeInSeconds++;

let hours = Math.trunc(timeInSeconds / 3600);

if(hours <= 9){

hours = '0' + hours;

}

let minets = Math.trunc((timeInSeconds % 3600) / 60);

if(minets <= 9){

minets = '0' + minets;

}

//let seconds = (timeInSeconds % 3600) % 60;

let seconds = timeInSeconds % 60;

if(seconds <= 9){

seconds = '0' + seconds;

}

$('#hours').text(hours);

$('#minutes').text(minets);

$('#seconds').text(seconds);

}

}

|  |
| --- |
| function timer() { |
|  | $("#start-timer").on("click", start); |
|  | $("#stop-timer").on("click", stop); |
|  | let timer; |
|  | let seconds = 0; |
|  | let started = undefined; |
|  |  |
|  | function step() { |
|  | seconds++; |
|  | $("#hours").text(pad(Math.trunc(seconds / 3600))); |
|  | $("#minutes").text(pad(Math.trunc((seconds % 3600) / 60))); |
|  | $("#seconds").text(pad(Math.trunc(((seconds % 3600) % 60) % 60))); |
|  | } |
|  |  |
|  | function start() { |
|  | if (!timer) { |
|  | timer = setInterval(step, 1000); |
|  | } |
|  | } |
|  |  |
|  | function stop() { |
|  | clearInterval(timer); |
|  | timer = undefined; |
|  | } |
|  |  |
|  | function pad(num) { |
|  | if (num <= 9) { |
|  | num = `0${num}` |
|  | } |
|  | return num; |
|  | } |
|  | } |

|  |
| --- |
| function timer() { |
|  | let time=0, intervalId, isRunning = false; |
|  | $('#start-timer').on('click', function () { |
|  | if(! isRunning){ |
|  | intervalId = setInterval(incrementTime, 1000); |
|  | isRunning = true; |
|  | } |
|  | }); |
|  |  |
|  | $('#stop-timer').on('click', function () { |
|  | clearInterval(intervalId); |
|  | isRunning = false; |
|  | }); |
|  |  |
|  | function incrementTime() { |
|  | time++; |
|  | $('#hours').text(("0" + Math.trunc(time/3600)).slice(-2)); |
|  | $('#minutes').text(("0" + Math.trunc((time/60)%60)).slice(-2)); |
|  | $('#seconds').text(("0" + Math.trunc(time%60)).slice(-2)); |
|  | } |
|  | } |

## 3.Form Validation

//the first zero test in Judge don't past

function validate(){

let $companyCheckBox = $('#company');

$companyCheckBox.on('change', showOrHideCompanySubForm);

let $submitButton = $('#submit');

$submitButton.on('click', validateForm);

function showOrHideCompanySubForm(){

if($(this).is(':checked')){

$('#companyInfo').css('display', 'block');

}

else{

$('#companyInfo').css('display', 'none');

}

}

function validateForm(event){

event.preventDefault();

let usernamePattern = /^[A-Za-z0-9]{3,20}$/;

//let passwordPattern = /^.{5,15}$/; //one test in Judge don't past

let passwordPattern = /^\w{5,15}$/;

let emaiPattern = /^.\*?@.\*?\..\*/;

let companyNumberPattern = /^[1-9][0-9]{3}$/;

let areValidFields = true;

let $username = $("#username");

let $password = $("#password");

let $confirmPassowrd = $("#confirm-password");

let $email = $("#email");

let $companyCheckbox = $("#company");

let $validDiv = $("#valid");

if($username.val().match(usernamePattern)){

$username.css("border", "none");

}

else {

$username.css("border-color", "red");

areValidFields = false;

}

if($password.val().match(passwordPattern)){

$password.css("border", "none");

}

else {

$password.css("border-color", "red");

areValidFields = false;

}

if($confirmPassowrd.val().match(passwordPattern) && $confirmPassowrd.val() === $password.val()){

$confirmPassowrd.css("border", "none");

}

else {

$confirmPassowrd.css("border-color", "red");

areValidFields = false;

}

if($email.val().match(emaiPattern)){

$email.css("border", "none");

}

else {

$email.css("border-color", "red");

areValidFields = false;

}

if($companyCheckbox.is(":checked")){

let $companyNumber = $("#companyNumber");

if($companyNumber.val().match(companyNumberPattern)){

$companyNumber.css("border", "none");

}

else {

$companyNumber.css("border-color", "red");

areValidFields = false;

}

}

if(areValidFields){

$validDiv.css("display", "block");

}

else {

$validDiv.css("display", "none");

}

}

}

|  |
| --- |
| //the first zero test don’t past  function validate() { |
|  | $('#company').on('change', showHideCompany); |
|  |  |
|  | $('#submit').on('click', function (ev) { |
|  | ev.preventDefault(); |
|  | let usernameRegex = /^[a-zA-Z0-9]{3,20}$/; |
|  | let passwordRegex = /^\w{5,15}$/; |
|  | let emailRegex = /@.\***\.**/; |
|  | let companyNumberRegex = /^[1-9]{1}[0-9]{3}$/; |
|  |  |
|  | let allFieldsValid = true; |
|  |  |
|  | if($('#username').val().match(usernameRegex)){ |
|  | $('#username').css('border', 'none'); |
|  | } else { |
|  | $('#username').css('border-color', 'red'); |
|  | allFieldsValid = false; |
|  | } |
|  |  |
|  | if($('#password').val().match(passwordRegex)){ |
|  | $('#password').css('border', 'none'); |
|  | } else { |
|  | $('#password').css('border-color', 'red'); |
|  | allFieldsValid = false; |
|  | } |
|  |  |
|  | if($('#email').val().match(emailRegex)){ |
|  | $('#email').css('border', 'none'); |
|  | } else { |
|  | $('#email').css('border-color', 'red'); |
|  | allFieldsValid = false; |
|  | } |
|  |  |
|  | if($('#confirm-password').val().match(passwordRegex) && $('#confirm-password').val().localeCompare($('#password').val()) == 0){ |
|  | $('#confirm-password').css('border', 'none'); |
|  | } else { |
|  | $('#confirm-password').css('border-color', 'red'); |
|  | allFieldsValid = false; |
|  | } |
|  |  |
|  | if($('#company').is(':checked')){ |
|  | if($('#companyNumber').val().match(companyNumberRegex)){ |
|  | $('#companyNumber').css('border', 'none'); |
|  | } else { |
|  | $('#companyNumber').css('border-color', 'red'); |
|  | allFieldsValid = false; |
|  | } |
|  | } |
|  |  |
|  | if(allFieldsValid){ |
|  | $('#valid').css('display', 'block'); |
|  | } else { |
|  | $('#valid').css('display', 'none'); |
|  | } |
|  |  |
|  | }); |
|  |  |
|  | function showHideCompany() { |
|  | if($(this).is(':checked')){ |
|  | $('#companyInfo').css('display', 'block'); |
|  | } else { |
|  | $('#companyInfo').css('display', 'none') |
|  | } |
|  | } |
|  | } |

**Problem 1 from Exam 18XII2018(“Gift Shop”)**

function solution() {

let $christmasGiftShop = $('#christmasGiftShop');

let $toyType = $('#toyType');

let $toyPrice = $('#toyPrice');

let $toyDescription = $('#toyDescription');

if ($toyType.val() && Number($toyPrice.val()) && $toyDescription.val()) {

let $giftDiv = $('<div>');

$giftDiv.addClass('gift');

let $img = $('<img src="gift.png"/>');

let $h2 = $('<h2>');

$h2.text($toyType.val());

let $p = $('<p>');

$p.text($toyDescription.val());

let $button = $('<button>');

$button.text(`Buy it for $${$toyPrice.val()}`);

$button.on('click', () => $giftDiv.remove());

$giftDiv.append($img, $h2, $p, $button);

$christmasGiftShop.append($giftDiv);

}

$toyType.val('');

$toyPrice.val('');

$toyDescription.val('');

}