

Rapport Atelier BootStrap : atelier 2

Développement web.

***Réaliser par* : *Encadré par*:**

* + Berbar Mohammed. Mr.Lotfi EL AACHAK

Objectif : L’objectif principal de ce Travail pratique et de se familiariser avec les concepts de l’API

Jquery.

# 1. Essayez d ‘implémenter le scripte base su Jquery pour valider les champs d ‘un formulaire d’inscription pour un site e-commerce, selon les règles suivantes :

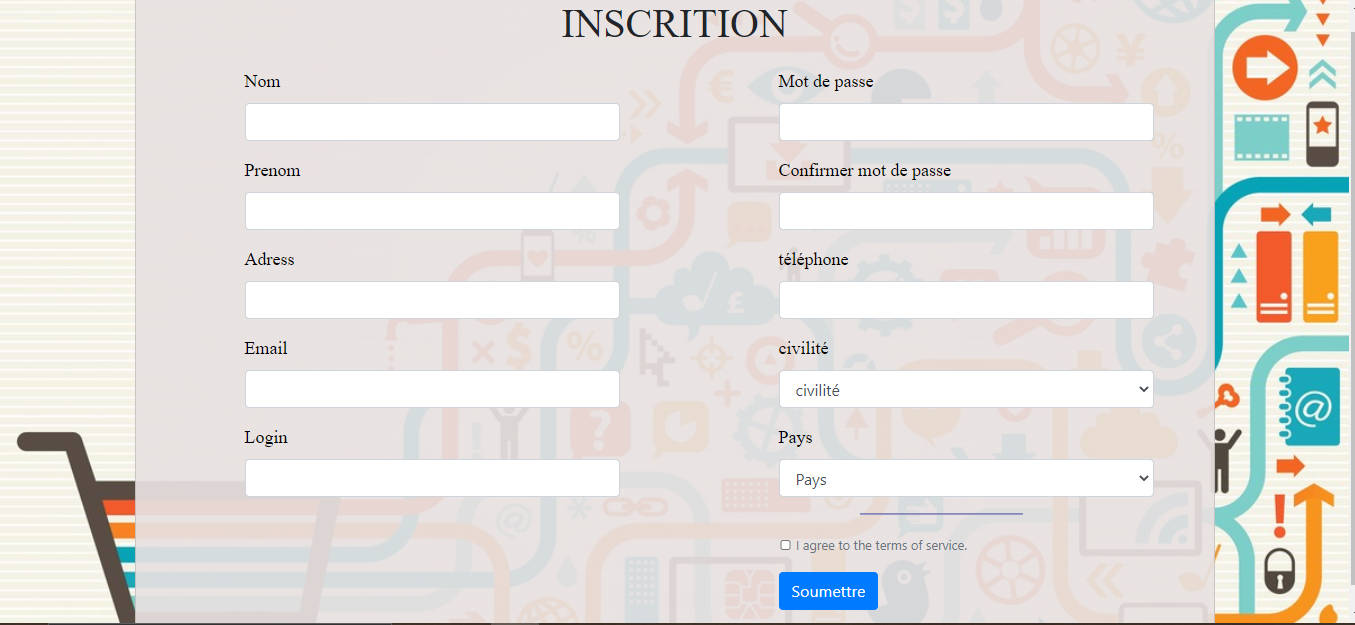
Les champs « nom , prénom, adresse, email , login , mot de passe, confirmer mot le passe,téléphone

,civilité, Pays » : Sont des champs obligatoires.

Le champ « email » doit être contrôlé par un regex email.

Le champ «téléphone» doit être contrôlé par une regex numéro de téléphone.

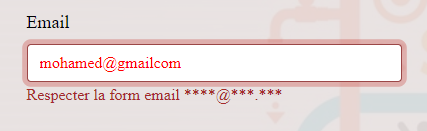
Les champs « mot de passe, confirmer mot le passe » doivent être identique.

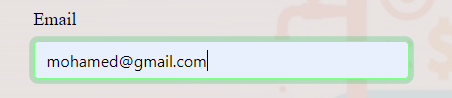


* Tous les champs sont obligatoire.

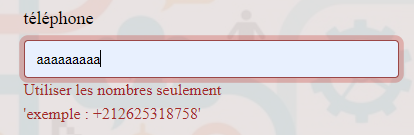
## 

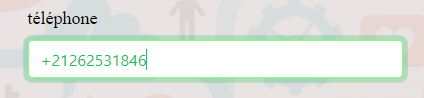
* Pour l’email



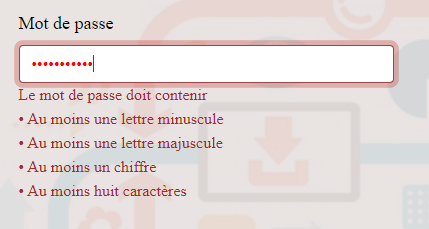


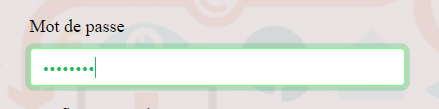
* Pour le téléphone





* Pour le mot de passe





## Réalisation de la page web.

### Partie jquery.

J’ai utilisé la bibliothèque jquery validation pour appliquer des regex au champs input.

* Partie rules dans le quelle on select les input en utilisant leur name a l’intérieur de chaque input sélecter on vérifier des conditions qui sont lié à des méthodes.
* jquery validation nous offre des méthodes prédéfinie come on peut ajouter nos propre

méthodes.

 rules: {

         Nom: {

            required:true,

            nowhitespace:true,

            lettersonly:true,

            minlength: 3,

         },

         Prenom: {

            required: true,

            lettersonly:true,

            minlength: 3,

            nowhitespace:true,

            lettersonly:true

         },

         Adress: {

            required: true,

            minlength: 10

         },

         Email: {

            required: true,

            reelemail:true

         },

         Login: {

            required: true,

            minlength: 5

         },

         Mot\_de\_passe:{

            strongPassword:true,

            required:true

         },

         Confirmer\_mot\_de\_passe:{

            required: true,

           equalTo: "input[name=Mot\_de\_passe]"

         },

         telephone:{

            required: true,

            numbersonly:true,

            minlength: 10

         },

         pays:{

            required: true

         },

         check:{

            required: true

         },

         civilité:{

            required: true,

         }

* Partie messages dans le quelle en peut attacher un message d’erreur a une condition d’un input si elle n’est pas vérifiée ou bien pour tous les conditions si elles sont pas vérifiées.

 messages: {

         Nom:{ required:"Veuillez fournir un nom d'au moins trois lettres"},

         Prenom: {required:"Veuillez fournir un prenom d'au moins trois lettres"},

         Adress: "Veuillez fournir un Adress d'au moins dix lettres",

         Email: {required:"L'email est incorrect"},

         Login:"Veuillez fournir un Login d'au moins cinq lettres",

         Adress: "Veuillez fournir un Adress d'au moins dix lettres",

         Mot\_de\_passe:{ required:"Le mot de pass est incorrect"},

         Confirmer\_mot\_de\_passe:"Le mot de pass est incorrect",

         telephone:{required:"Le numero de telephone est incorrect <br>esseyez de saisir au moins dix chifre ",

         minlength:"Veuillez fournir un Nombre de telephone <br>d'au moins dix nombres"},

         pays: "Choisir un pays",

         civilité:"Choisir une civilité",

         check:"Accepter les condition et les terme pour soumettre "

      },

* Method prédéfinie pour manipuler l’emplacement du messages d’erreur.

 errorPlacement:function(error,element){

         if(element.is(":checkbox")){

            error.appendTo(element.parents(".warn"));

         }else {

       error.insertAfter(element);

         }

      }

* « AddMethod » Méthode de jQuery pour ajouter des méthodes .
* //method pour verifier que le champ de password est remplie avec un mots de pass fort
* $.validator.addMethod('strongPassword', function(value, element) {
* var id='#'+element.id;
* if(/^(?=.\*[A-Z])(?=.\*[a-z])(?=.\*\d)(?=.\*[-+!\*$@%\_])([-+!\*$@%\_\w]{8,15})$/.test(value)){
* $(id).removeClass('warning');
* return /^(?=.\*[A-Z])(?=.\*[a-z])(?=.\*\d)(?=.\*[-+!\*$@%\_])([-+!\*$@%\_\w]{8,15})$/.test(value);
* }
* else{
* $(id).addClass('warning');
* return /^(?=.\*[A-Z])(?=.\*[a-z])(?=.\*\d)(?=.\*[-+!\*$@%\_])([-+!\*$@%\_\w]{8,15})$/.test(value); }
* },'Le mot de passe doit contenir  <br>• Au moins une lettre minuscule  <br>• Au moins une lettre majuscule  <br>• Au moins un chiffre  <br>• Au moins huit caractères' );
* //method pour verifier que le champ est n'est pes remplie evec des espaces
* $.validator.addMethod( "nowhitespace", function( value, element ) {
* var id='#'+element.id;
* if(/^\S+$/i.test( value )){
* $(id).removeClass('warning');
* }
* else{
* $(id).addClass('warning');}
* return /^\S+$/i.test( value );
* }, "N'est utiliser pas d'espace" );
* /\* -------------------------------------------------------------------------------- \*/
* //method pour verifier que le champ est remplie avec des lettres seulement
* $.validator.addMethod( "lettersonly", function( value, element ) {
* var id='#'+element.id;
* if(/^[a-z]+$/i.test( value )){
* $(id).removeClass('warning');
* }
* else{
* $(id).addClass('warning');}
* return  /^[a-z]+$/i.test( value );
* }, "Utiliser les lettres seulement" );
* //method pour verifier que le champ est remplie avec des nombres seulement
* $.validator.addMethod("numbersonly",function(value,element) {
* var id='#'+element.id;
* if(/^\+[0-9]+$/i.test( value )){
* $(id).removeClass('warning');
* }
* else{
* $(id).addClass('warning');}
* return  /^\+[0-9]+$/i.test( value );
* }, "Utiliser les nombres seulement<br>'exemple : +212625318758'" );
* //method pour verifier un email
* $.validator.addMethod("reelemail",function(value,element) {
* var id='#'+element.id;
* if(/^[^@]+@\w+(\.\w+)+\w$/.test( value )){
* $(id).removeClass('warning');
* }
* else{
* $(id).addClass('warning');}
* return  /^[^@]+@\w+(\.\w+)+\w$/.test( value );
* }, "Respecter la form email \*\*\*\*@\*\*\*.\*\*\*" );
* Des méthodes pour manipuler le border rouge des input et select.

//Si le champ est remplie ou bien vide enlever le rouge border

$("input").keyup(function(){

      $(this).removeClass('warning');

});

//Si tout les champs sont remplie enlever les rouges borders sinon ajouter les rouges borders

$("#submit").on('click',function(){

   $("input " ).each(function () {

      if($(this).val() === '') {

         $(this).addClass('warning');

      } else {

         $(this).removeClass('warning');

      }

   });

//Si un choix est selectionner enlever le rouge border sinon ajouter les rouges borders

   $("select" ).each(function () {

      if($(this).val()==null) {

         $(this).addClass('warning');

      }

   });

   //Si un choix est selectionner enlever le rouge border sinon ajouter les rouges borders

   $('select').change(function(){

      $("select" ).each(function () {

         if($(this).val()!=null) {

            $(this).removeClass('warning');

         }

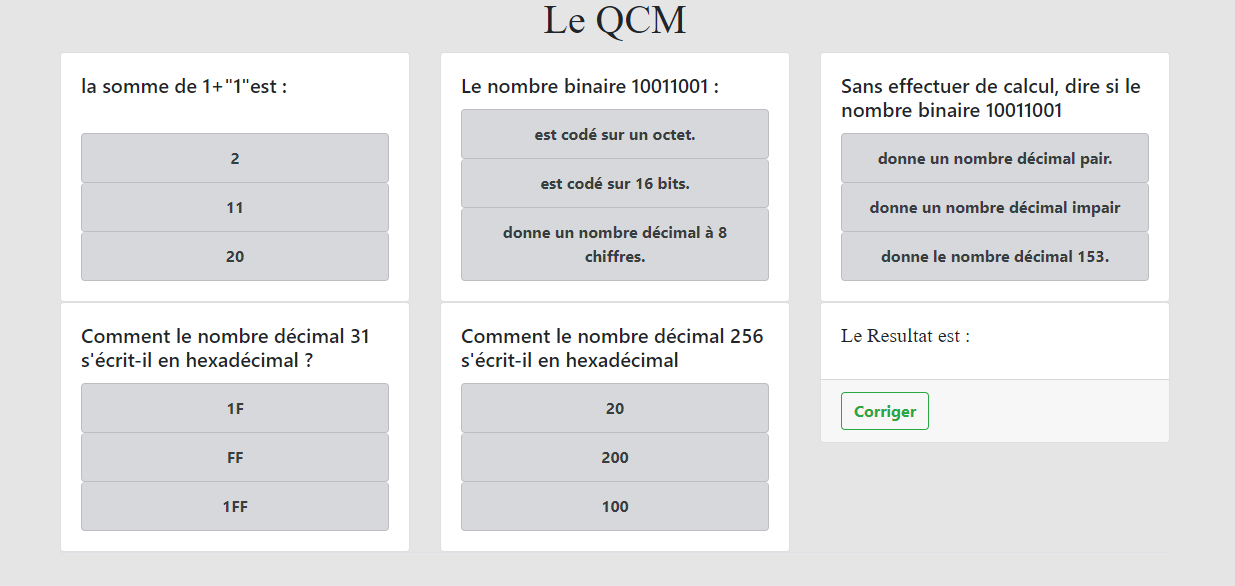
      });

   });

# 2. QCM cote client :

a. Créer une page contenant un QCM avec 5 questions qui auront chacune 3 réponses

possibles

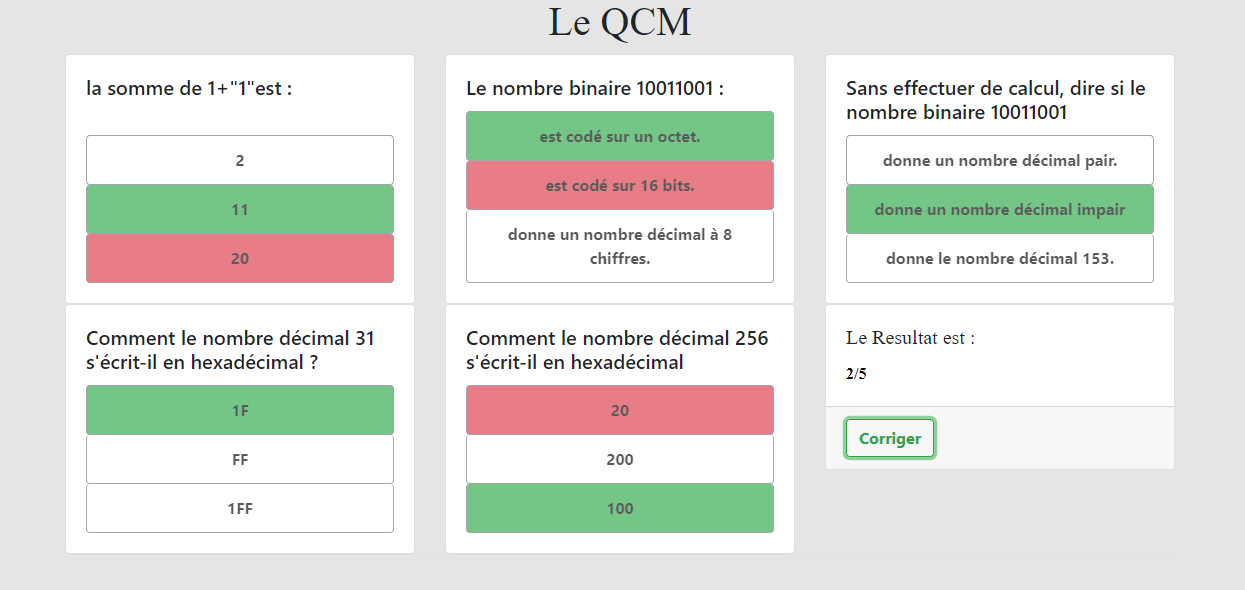


b. Ajouter un bouton « Corriger » qui, au lieu d’envoyer les réponses `a un serveur, les

vérifié cote client : pour chaque question, si la bonne réponse est sélectionne, la question apparaît en,vert, sinon elle apparaît en rouge.

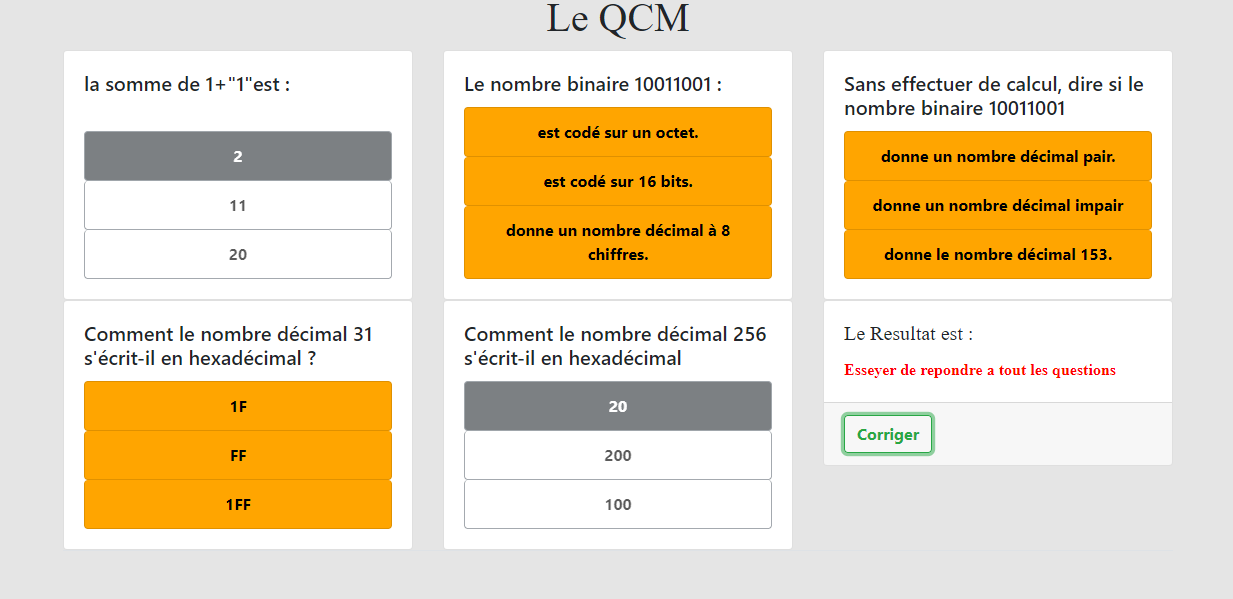
c. De plus (toujours après appui sur le bouton « Corriger »), lorsqu’une mauvaise réponse a

été sélectionnée, la réponse sélectionnée apparaît en rouge et la bonne réponse en vert. Lorsque la bonne réponse a été sélectionnée, elle apparaît en vert.

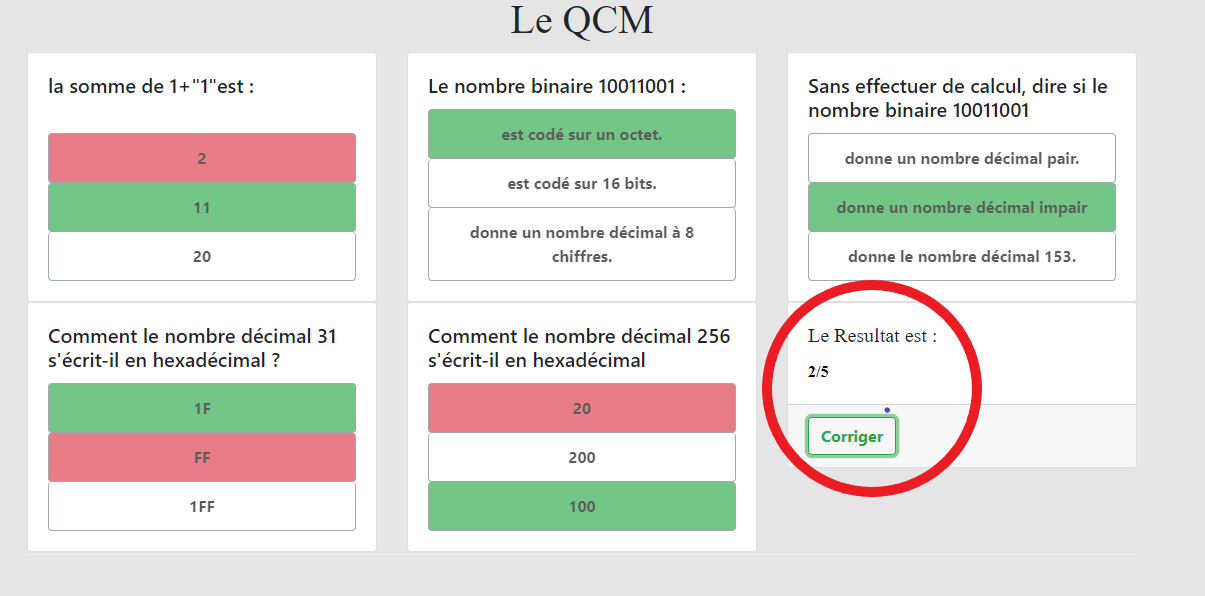


d. Lorsque l’utilisateur clique sur « Corriger » alors que certaines questions n’ont pas de

réponse sélectionnée, alors la correction n’a pas lieu et ces questions apparaissent en orange.



e. Apres la correction, un message apparaît sous le QCM avec la note obtenue.



## Réalisation de la page web.

### Partie jquery.

* Cette méthode et liée a l ‘évent click de la réponse choisie par l’utilisateur, quand l’utilisateur click une réponse on ajoute la classe check a cette élément grâce à cette classe on distingue les réponses choisie, après on désactive tous les réponses lié à la question.

 $('button').on('click', function () {

      var name = $(this).prop("name");

      var name\_selected = "[name='" + name + "']";

      if (name != 'corriger') {

         $(this).addClass('check');

         $(this).addClass('bg-dark');

         j++;

         jQuery(name\_selected).each(function () {

            if ($(this).hasClass('bg-orange')) {

               $(this).removeClass('bg-orange')

            }

            $(this).prop('disabled', true);

         });

      }

   });

* Quand l’utilisateur click sur le buttons corriger on cherche où se trouve la classe check

est on colore les buttons selon la rapponse et les conditions de l’exercice.

 $('#corriger').on('click', function () {

      jQuery('button').each(function () {

         if ($(this).prop("name") != 'corriger') {

            var locate = "[name='" + $(this).prop("name") + "']";

            if ($(".wrong\_2" + locate).hasClass("check") && j == 5) {

               $('.wrong\_2' + locate).addClass('bg-danger');

               $('.right' + locate).addClass('bg-success');

               $('.wrong\_2' + locate).removeClass('bg-dark');

            }

            if ($(".wrong\_1" + locate).hasClass("check") && j == 5) {

               $('.wrong\_1' + locate).addClass('bg-danger');

               $('.right' + locate).addClass('bg-success');

               $('.wrong\_1' + locate).removeClass('bg-dark');

            }

            if ($(".right" + locate).hasClass("check") && j == 5) {

               i++;

               $('.right' + locate).addClass('bg-success');

               $('.right' + locate).removeClass('bg-dark');

            }

            if (!$(".right" + locate).hasClass("check") && !$(".wrong\_1" + locate).hasClass("check") && !$(".wrong\_2" + locate).hasClass("check")) {

               $('.right' + locate).addClass('bg-orange');

               $('.wrong\_1' + locate).addClass('bg-orange');

               $('.wrong\_2' + locate).addClass('bg-orange');

            }

            $('.right' + locate).addClass('color');

            $('.wrong\_1' + locate).addClass('color');

            $('.wrong\_2' + locate).addClass('color');

         }

      });

- Si l’utilisateur a répondu a tous les questions on calcule sa note si non on lui demande de répondre a tous les questions.

 if (j == 5 && i <= 5) {

         $('#resultat').text(i + "/5");

         j = -1;

         $('#resultat').css({

            'color': 'black'

         });

      } else if (j == -1) {

         $('#resultat').text("Vous avez terminer le qcm");

         $('#resultat').css({

            'color': 'black'

         })

      } else {

         $('#resultat').text("Esseyer de repondre a tout les questions");

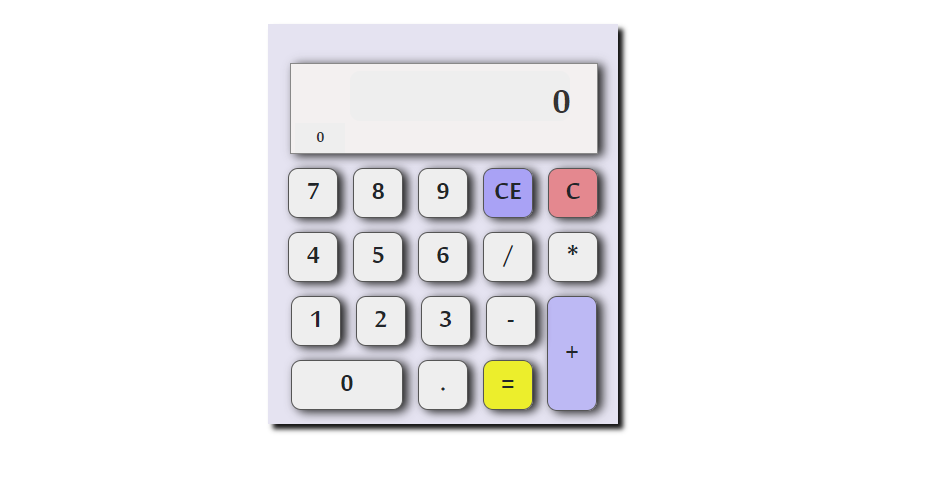
         $('#resultat').css({

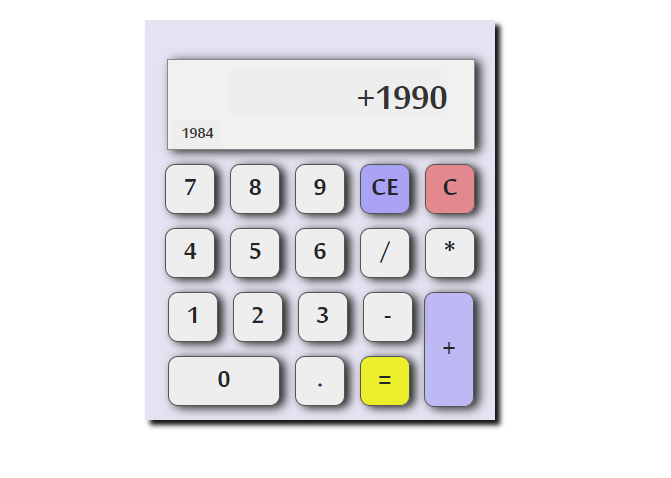
            'color': 'red'

         })

      }

# 3. Développer une calculatrice classique, en utilisant l’API Jquery :





## Réalisation de la page web.

### Partie jquery.

* Des function pour afficher le resultat sur la petite et la grand screen.
* updateScreen = function(displayValue) {
* $('#screen').html(displayValue);
* };
* updateSmallScreen = function(displayValue) {
* $('#small\_screen').html(displayValue);
* };
* Des methodes pour distingué les buttons clicker et pour faire les opeartions.

 //verifier si l'argument de la fonction est un nombre

  isNumber = function(value) {

    return !isNaN(value);

  };

  //verifier si l'argument de la fonction est un operateur

  isOperator = function(value) {

    return value === '/' || value === '\*' || value === '+' || value === '-';

  };

    //fair une operation 'operation' entre 'a' et 'b'

  operate = function(a, b, operation) {

    a = parseFloat(a);

    b = parseFloat(b);

    console.log(a, b, operation);

    if (operation === '+')return a + b;

    if (operation === '-') return a - b;

    if (operation === '\*') return a \* b;

    if (operation === '/'){

      return a / b;}

 };

* on clique sur un nombre , en stock sa valeur dans « currentEntry\_String » puis on clique sur l’opérateur et on affect la valeur de « currentEntry\_String »

a « prevEntry\_String » et l’opérateur a « operation » et on stock la deuxième valeur dans « currentEntry\_String » puis on fait l’opération entre « prevEntry\_String » et « currentEntry\_String ».

 //fonctionement de la calculatrice selon le button qu'on click

  $('.button').on('click', function(){

    var buttonPressed = $(this).html();

    if (buttonPressed === "C") {

      currentEntry\_String='0' ;

      updateScreen(currentEntry\_String);

      updateSmallScreen(currentEntry\_String);

    } else if (buttonPressed === "CE") {

      prevEntry\_String='0' ;

      updateScreen( prevEntry\_String);

    }  else if (buttonPressed === '.') {

      currentEntry\_String.substring( currentEntry\_String.length)

      currentEntry\_String+=buttonPressed;

      updateScreen(currentEntry\_String);

    } else if (isNumber(buttonPressed)) {

      if(currentEntry\_String=='0')

     {currentEntry\_String=buttonPressed;

      updateSmallScreen(currentEntry\_String);}

      else {

        currentEntry\_String+=buttonPressed;

      }

      updateScreen(currentEntry\_String);

    } else if (isOperator(buttonPressed)) {

      prevEntry\_String=currentEntry\_String;

      currentEntry\_String=buttonPressed;

      updateScreen(currentEntry\_String);

      operation=buttonPressed;

    } else if (buttonPressed === '=') {

      updateSmallScreen(prevEntry\_String);

      currentEntry\_String = operate(prevEntry\_String, currentEntry\_String.substring(1, currentEntry\_String.length), operation);

      console.log(currentEntry\_String);

      updateScreen(currentEntry\_String);

      prevEntry\_String=currentEntry\_String;

      updateSmallScreen(currentEntry\_String);

      operation = null;

    }

    if(currentEntry\_String.length>12)

    currentEntry\_String=currentEntry\_String.substring(0, 11);

  });