**Liste matériel pour développement web**

NetBeans : Pour coder en HTML/PHP/CSS

Bootstrap : pour adapter en mobile, pc, tablette

WampServer : Hébergement page web

Plugin PHP :

Apache :

MySQL :

**Bootstrap :**

Prérequis pour installer Bootstrap :

Copiez-collez la feuille de style <link>dans votre <head>avant toutes les autres feuilles de style pour charger notre CSS.

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">

Beaucoup de nos composants nécessitent l'utilisation de JavaScript pour fonctionner. Plus précisément, ils nécessitent [jQuery](https://jquery.com/) , [Popper.js](https://popper.js.org/) et nos propres plugins JavaScript. Placez les éléments suivants <script>vers la fin de vos pages, juste avant la </body>balise de fermeture, pour les activer. jQuery doit venir en premier, puis Popper.js, puis nos plugins JavaScript.

<script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6"

crossorigin="anonymous"></script>

Ensemble de la page :

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS -->

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">

<title>Hello, world!</title>

</head>

<body>

<h1>Hello, world!</h1>

<!-- Optional JavaScript -->

<!-- jQuery first, then Popper.js, then Bootstrap JS -->

<script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="anonymous"></script>

</body>

</html>

**Ou télécharger …**

Voir : <https://getbootstrap.com/docs/4.4/getting-started/download/>

<https://getbootstrap.com/docs/4.4/getting-started/introduction/>

**Plugin PHP :**

<https://wpformation.com/19-plugins-wordpress-2018/>

<https://wpformation.com/10-outils-faciliter-vie-wordpress/>

**Etapes pour développement Web**

**Page de connexion :**

<http://www.codeurjava.com/2016/12/formulaire-de-login-avec-html-css-php-et-mysql.html>

login.php:

<html>

<head>

<meta charset="utf-8">

<!-- importer le fichier de style -->

<link rel="stylesheet" href="style.css" media="screen" type="text/css" />

</head>

<body>

<div id="container">

<!-- zone de connexion -->

<form action="verification.php" method="POST">

<h1>Connexion</h1>

<label><b>Nom d'utilisateur</b></label>

<input type="text" placeholder="Entrer le nom d'utilisateur" name="username" required>

<label><b>Mot de passe</b></label>

<input type="password" placeholder="Entrer le mot de passe" name="password" required>

<input type="submit" id='submit' value='LOGIN' >

<?php

if(isset($\_GET['erreur'])){

$err = $\_GET['erreur'];

if($err==1 || $err==2)

echo "<p style='color:red'>Utilisateur ou mot de passe incorrect</p>";

}

?>

</form>

</div>

</body>

</html>

verification.php :

<?php

session\_start();

if(isset($\_POST['username']) && isset($\_POST['password']))

{

// connexion à la base de données

$db\_username = 'root';

$db\_password = 'mot\_de\_passe\_bdd';

$db\_name = 'nom\_bdd';

$db\_host = 'localhost';

$db = mysqli\_connect($db\_host, $db\_username, $db\_password,$db\_name)

or die('could not connect to database');

// on applique les deux fonctions mysqli\_real\_escape\_string et htmlspecialchars

// pour éliminer toute attaque de type injection SQL et XSS

$username = mysqli\_real\_escape\_string($db,htmlspecialchars($\_POST['username']));

$password = mysqli\_real\_escape\_string($db,htmlspecialchars($\_POST['password']));

if($username !== "" && $password !== "")

{

$requete = "SELECT count(\*) FROM utilisateur where

nom\_utilisateur = '".$username."' and mot\_de\_passe = '".$password."' ";

$exec\_requete = mysqli\_query($db,$requete);

$reponse = mysqli\_fetch\_array($exec\_requete);

$count = $reponse['count(\*)'];

if($count!=0) // nom d'utilisateur et mot de passe correctes

{

$\_SESSION['username'] = $username;

header('Location: principale.php');

}

else

{

header('Location: login.php?erreur=1'); // utilisateur ou mot de passe incorrect

}

}

else

{

header('Location: login.php?erreur=2'); // utilisateur ou mot de passe vide

}

}

else

{

header('Location: login.php');

}

mysqli\_close($db); // fermer la connexion

?>

principale.php :

<html>

<head>

<meta charset="utf-8">

<!-- importer le fichier de style -->

<link rel="stylesheet" href="style.css" media="screen" type="text/css" />

</head>

<body style='background:#fff;'>

<div id="content">

<!-- tester si l'utilisateur est connecté -->

<?php

session\_start();

if($\_SESSION['username'] !== ""){

$user = $\_SESSION['username'];

// afficher un message

echo "Bonjour $user, vous êtes connecté";

}

?>

</div>

</body>

</html>

Déconnexion.php :

<html>

<head>

<meta charset="utf-8">

<!-- importer le fichier de style -->

<link rel="stylesheet" href="style.css" media="screen" type="text/css" />

</head>

<body style='background:#fff;'>

<div id="content">

<a href='principale.php?deconnexion=true'><span>Déconnexion</span></a>

<!-- tester si l'utilisateur est connecté -->

<?php

session\_start();

if(isset($\_GET['deconnexion']))

{

if($\_GET['deconnexion']==true)

{

session\_unset();

header("location:login.php");

}

}

else if($\_SESSION['username'] !== ""){

$user = $\_SESSION['username'];

// afficher un message

echo "<br>Bonjour $user, vous êtes connectés";

}

?>

</div>

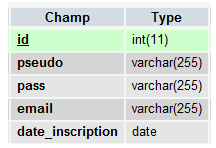
</body>

</html>

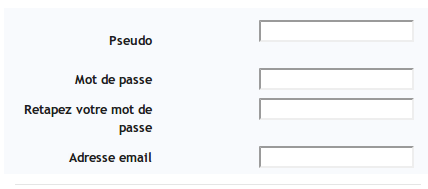
**création compte superviseur**

<https://openclassrooms.com/fr/courses/918836-concevez-votre-site-web-avec-php-et-mysql/917948-tp-creez-un-espace-membres>

Base de données :



Formulaire d’inscription :



**Base de données**

* Pour créneau

|  |  |  |  |
| --- | --- | --- | --- |
| Nom | Numéro de tel | Nombre de joueurs | Date |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Liste matériel pour développement sur Arduino**

* Arduino
* Pavé numérique
* Buzzer
* Pavé numérique
* Écran LCD

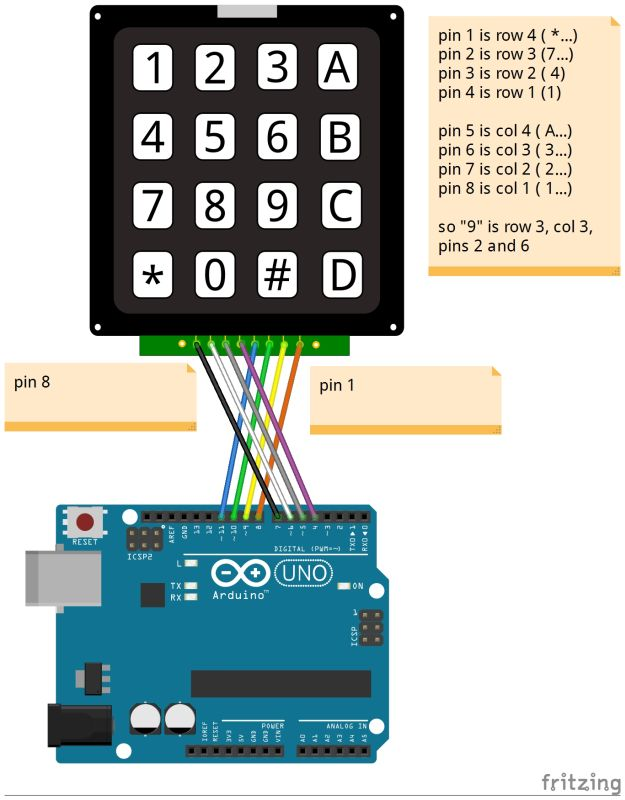
**Etapes pour développement sur Arduino**

**Installer logiciel Arduino sur pc Windows :**

<https://www.arduino.cc/en/Guide/Windows>

**Connecter pavé numérique sur Arduino :**

<https://knowledge.parcours-performance.com/clavier-projet-arduino/>



#include <Keypad.h>

const byte ROWS = 4; //four rows

const byte COLS = 4; //four columns

//define the cymbols on the buttons of the keypads

char keys[ROWS][COLS] = {

{'1', '2', '3', 'A'},

{'4', '5', '6', 'B'},

{'7', '8', '9', 'C'},

{'\*', '0', '#', 'D'}

};

byte rowPins[ROWS] = {11, 10, 9, 8}; //rows 1 to 4

byte colPins[COLS] = {7, 6, 5, 4}; //columns 4 to 1

//initialize an instance of class NewKeypad

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS);

void setup(){

Serial.begin(9600);

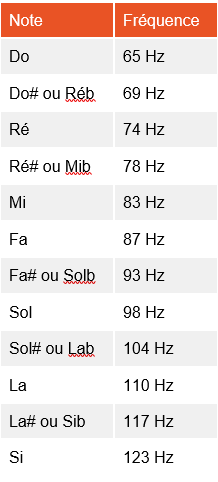
Serial.println("Setup done");

}

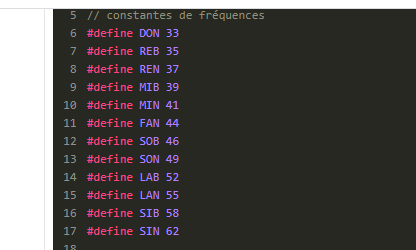
**Jouer la musique « au clair de la lune » sur le buzzer :**

Fréquence :

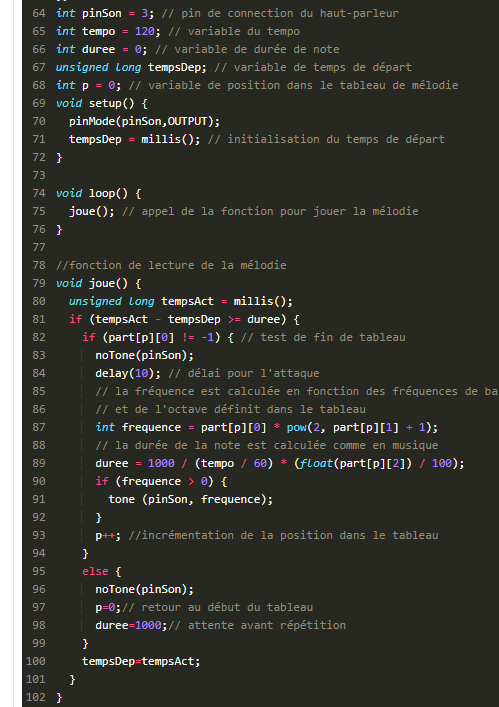
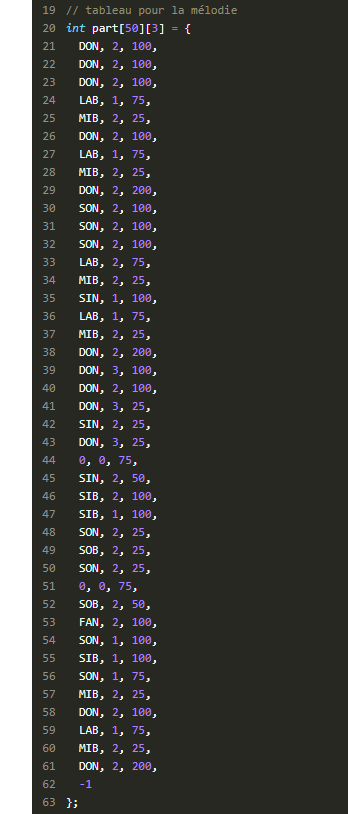
Note de musique au clair de la lune :



Sol sol sol la si la sol si la la sol  
Sol sol sol la si la sol si la la sol  
La la la la mi mi  
La sol fa mi ré  
Sol sol sol la si la sol si la la sol



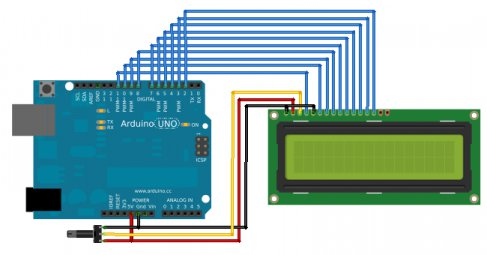
Code pour faire jouer la musique :

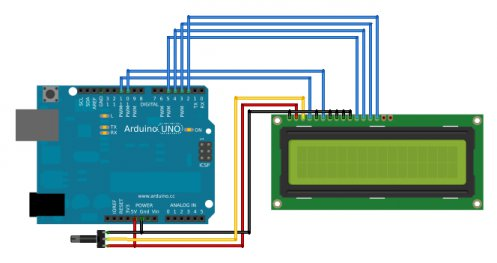


<https://openclassrooms.com/fr/courses/3290206-perfectionnez-vous-dans-la-programmation-arduino/3295331-generez-des-sons>

**Afficher énigme sur écran LCD**

Connecter Ecran LCD à l’Arduino :

8 fils :

4 fils :

#include "LiquidCrystal.h" *// ajout de la librairie*

*// Vérifiez les broches !*

LiquidCrystal lcd(11,10,9,8,7,6,5,4,3,2); *// liaison 8 bits de données*

LiquidCrystal lcd(11,10,5,4,3,2); *// liaison 4 bits de données*

**void** setup()

{

lcd.begin(16,2); *// utilisation d'un écran 16 colonnes et 2 lignes*

lcd.write(**"Salut ca zeste ?"**); *// petit test pour vérifier que tout marche*

}

**void** loop() {}

Code permettant d’activer le buzzer et d’afficher l’énigme sur l’écran  
  
#include <Password.h> // http://www.arduino.cc/playground/uploads/Code/Password.zip  
#include <Keypad.h>   // http://www.arduino.cc/playground/uploads/Code/keypad.zip  
  
#define green 12                   // LED rouge attachée à pin 12 de l'Arduino  
#define red 13                     // LED verte attachée à pin 13 de l'Arduino  
  
Password pwd = Password("1234");   // définition du mot de passe  
  
boolean alarm = false;             // variable stockant l'état de l'alarme  
                                   // false = alarme OFF | true = alarme ON  
boolean intruder = false;          // variable stockant l'état de l'intrusion  
                                   // false = pas d'intrus détecté | true = intrus détecté  
                                   
unsigned long previousMillis = 0;  // dernière fois que la LED rouge a clignoté  
int interval = 250;                // intervalle auquel faire clignoter la LED  
  
#define ROWS 4                                                       //  
#define COLS 3                                                       //  
char keys[ROWS][COLS] =                                              //  
{                                                                    //  
  {'1','2','3'},                                                     //  
  {'4','5','6'},                                                     // configuration  
  {'7','8','9'},                                                     // du clavier  
  {'\*','0','#'}                                                      //  
};                                                                   //  
byte rowPins[ROWS] = {8, 7, 6, 5};                                   //  
byte colPins[COLS] = {4, 3, 2};                                      //  
                                                                     //  
Keypad kpd = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS); //  
  
void setup()  
{  
  pinMode(red, OUTPUT);  
  pinMode(green, OUTPUT);  
   
  kpd.addEventListener(kpdEvent);  
}  
  
void loop()  
{  
  char Key = kpd.getKey();  
  // en cas d'intrus détecté :  
  if(intruder == true && alarm == true)  
  {  
    unsigned long currentMillis = millis();  
    // ... on regarde s'il est temps de faire clignoter la LED, c'est-à-dire  
    // s'il s'est passé suffisamment de temps depuis la dernière fois qu'on a vérifié  
    if(currentMillis - previousMillis > interval)  
    {  
      previousMillis = currentMillis;  
      digitalWrite(red, !digitalRead(red));  
    }  
  }  
  else  
  {  
    digitalWrite(red, LOW);  
  }  
}  
  
void kpdEvent (KeypadEvent Key)  
{  
  switch (kpd.getState())  
  {  
    case PRESSED :  
      switch (Key)  
      {  
        // appui sur '\*' -> vérification de la saisie en cours  
        case '\*' : checkPassword(); break;  
        // appui sur '#' -> réinitialisation de la saisie en cours  
        case '#' : pwd.reset(); break;  
        // sinon on ajoute le chiffre à la combinaison  
        default  : pwd.append(Key); break;  
      }  
    default : break;  
  }  
}  
  
void checkPassword(void)  
{  
  // on remet à zéro l'état du mot de passe  
  intruder = false;  
  // si le mot de passe est juste...  
  if (pwd.evaluate())  
  {  
    // ...on met à jour l'état de l'alarme : ON>OFF / OFF>ON  
    alarm = !alarm;  
     
    switch (alarm)  
    {  
      case true :  
      // la LED rouge signale la mise en service de l'alarme  
        digitalWrite(red, HIGH);  
        delay(2000);  
        digitalWrite(red, LOW);  
        pwd.reset();  // on remet à zéro la saisie  
        break;  
      case false :  
      // la LED verte signale l'extinction de l'alarme  
        digitalWrite(green, HIGH);  
        delay(2000);  
        digitalWrite(green, LOW);  
        pwd.reset();  // on remet à zéro la saisie  
        break;  
      default :  
        break;  
    }  
  }  
  // si le mot de passe est faux...  
  else  
 {  
   // ...on signale l'intrusion à la loop() qui déclenche l'alarme  
   // jusqu'à ce que le bon mot de passe soit rentré  
   intruder = true;  
 }  
 // on remet à zéro systématiquement après avoir vérifié pour ne pas avoir d'erreur  
 pwd.reset();  
}

[http://forum.snootlab.com/viewtopic.php?f=43&t=671#](http://forum.snootlab.com/viewtopic.php?f=43&t=671)