

```

// Tapez votre code Javascript ici

// notes on définit les notes avec leurs fréquences

var dom = 262;
var re = 294;
var mi = 330;
var fa= 349;
var sol= 392;
var la = 440;
var si = 494;

// Piano Modifié
function enveloppe(t)
{
  if (t < 5000)
  {
    t = ((t / 44100) * Math.pow(0.3 - t / 44100, 8) / 0.05);
  }
  else {
    if (t < 25000)
    {
      t = ((5000 / 44100) * Math.pow(0.5- 5000 / 44100,8 ) / 0.075);
    }
    {
      t =(((t - 20000) / 44100) * Math.pow(0.65- (t - 20000) / 44100, 4) / 0.075);
    }
  }
  return (t);
}

function Harmonique(k)
{
  k= 1 / k;
  return (k);
}

function note( f ) // fréquence de la note en parametre e pour le la
{
  var son
  var data = Tableau(0);
  var amplitude = 0;
  for ( t = 0; t < 44100; t++)
  {
    var nb = 0;
    for (k = 1; k < 100; k *= 2)

```

```
    { /*Synthèse additive on multiplie par des valeurs correspondant à la décomposition le sinus en plusieurs sinus pour avoir un son coorrect*/
```

```
        nb = Math.sin(f * 360 * t / 44100 * 3.14 / 180) * Harmonique(k)+  
        0.5*Math.sin(f*2 * 360 * t / 44100 * 3.14 / 180)* Harmonique(k)+  
        0.2*Math.sin(f*5 * 360 * t / 44100 * 3.14 / 180) * Harmonique(k)+  
        0.25*Math.sin(f*4 * 360 * t / 44100 * 3.14 / 180) * Harmonique(k)  
        0.125*Math.sin(f*8 * 360 * t / 44100 * 3.14 / 180) * Harmonique(k)+  
        0.707*Math.sqrt(2)* Math.sin(f *360 * t / 44100 * 3.14 / 180) * Harmonique(k)+  
        0.888*Math.sin(f*9/8 *360 * t / 44100 * 3.14 / 180) * Harmonique(k);  
    }
```

```
data[t] = nb * enveloppe(t);  
if (amplitude < Math.abs(data[t]))  
{  
    amplitude = Math.abs(data[t]);  
}
```

```
for ( t = 0; t < 44100; t++)  
{  
    data[t] = Math.round(128 + 127 * data[t] / amplitude);  
}
```

```
son = CreerSon(data, 44100);
```

```
return son.play();
```

```
}
```

```
/* Partition, le tableau est défini et on attribue à chaque case, la note maintenant il reste plus qu'à le jouer */
```

```
function partition(k)
```

```
{ var part=[,sol,sol,sol,la,si,la,sol,si,la,la,sol,la,la,la,mi,mi,la,sol,fa,mi,re];  
    return part[k];  
}
```

```
/*fonction assemblage permet de clarifier les choses et de lire une note enfonction de la partition dans une seule fonction*/
```

```
function assemblage(n)  
{  
    return note(partition(n));  
}
```

```
/* appel récursif*/
```

```
function lecture(s)
```

```
{  
    if (s<22)
```

```
{  
    s=s+1; /*ici il faut incrémenter s avant d'exécuter la suite des évènements sinon bug*/
```

```
/*il est important d'appeler lecture() sous forme de chaîne sinon on a un message too much recursion!*/
```

```
setTimeout('lecture'+(''+s+)',800);
```

```
}  
return assemblage(s);
```

```
}
```

```
lecture(0);
```

