

Introduction à la Simulation

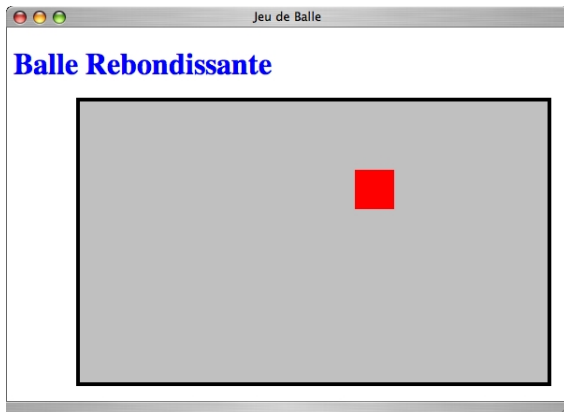
ou comment créer votre premier jeu vidéo

Jacques Duma



<http://math.et.info.free.fr/>
<http://ateliermathematique.free.fr/>

Aspect visuel souhaité



Page HTML :

«index.html»

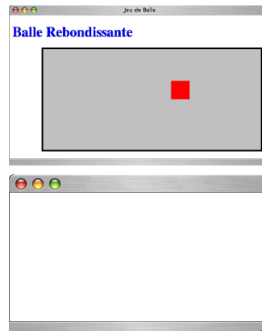
```
<html>
```

```
</html>
```

Page HTML :

«index.html»

```
<html>  
  <head>  
  
  </head>  
  <body>  
  
  </body>  
</html>
```

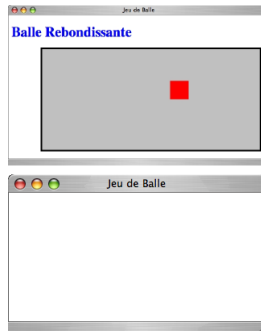


Page HTML :

«index.html»

```
<html>
  <head>
    <title>Jeu de Balle</title>
  </head>
  <body>

  </body>
</html>
```

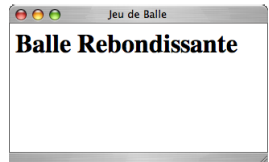


Page HTML :

«index.html»

```
<html>
  <head>
    <title>Jeu de Balle</title>
  </head>
  <body>
    <h1>Balle Rebondissante</h1>

    </body>
</html>
```



Page HTML :

«index.html»

```

<html>
  <head>
    <title>Jeu de Balle</title>
  </head>
  <body>
    <h1>Balle Rebondissante</h1>
    <div>
      CADRE
    </div>
  </body>
</html>

```



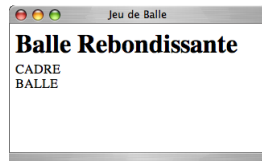
Page HTML :

«index.html»

```

<html>
  <head>
    <title>Jeu de Balle</title>
  </head>
  <body>
    <h1>Balle Rebondissante</h1>
    <div>
      CADRE
      <div>BALLE</div>
    </div>
  </body>
</html>

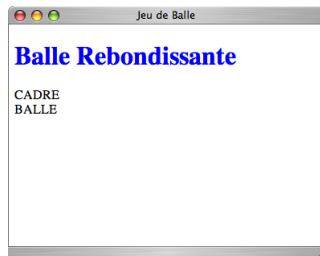
```



Feuille de styles en CSS :

«`style.css`»

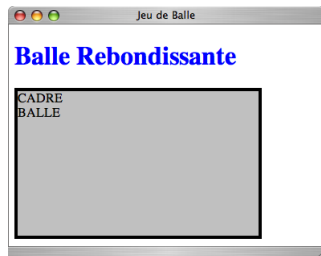
```
h1 { color : blue ; }
```



Feuille de styles en CSS :

«style.css»

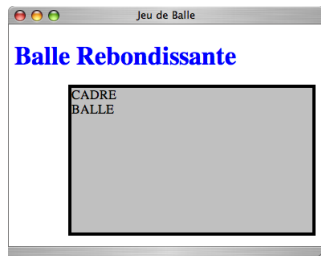
```
h1 { color : blue ; }  
#cadre {  
    width : 500px ; height : 300px ;  
    border : 4px black solid ;  
    background-color : silver ;  
}
```



Feuille de styles en CSS :

«style.css»

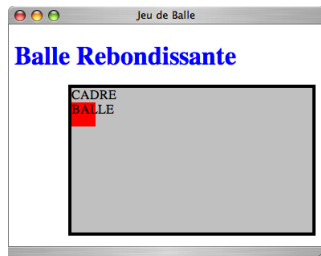
```
h1 { color : blue; }  
#cadre {  
    width : 500px; height : 300px;  
    border : 4px black solid;  
    background-color : silver;  
    position : fixed;  
    top : 75px; left : 75px;  
}
```



Feuille de styles en CSS :

«style.css»

```
h1 { color : blue; }  
#cadre {  
    width : 500px; height : 300px;  
    border : 4px black solid;  
    background-color : silver;  
    position : fixed;  
    top : 75px; left : 75px;  
}  
#balle {  
    width : 100px; height : 100px;  
    background-color : red;  
}  
}
```



Feuille de styles en CSS :

«style.css»

```

h1 { color : blue ; }
#cadre {
    width : 500px ; height : 300px ;
    border : 4px black solid ;
    background-color : silver ;
    position : fixed ;
    top : 75px ; left : 75px ;
}
#balle {
    width : 100px ; height : 100px ;
    background-color : red ;
    position : absolute ;
    top : 10px ; left : 10px ;
}

```



Feuille de styles en CSS :

«style.css»

```

h1 { color : blue; }
#cadre {
    width : 500px; height : 300px;
    border : 4px black solid;
    background-color : silver;
    position : fixed;
    top : 75px; left : 75px;
}
#balle {
    width : 100px; height : 100px;
    background-color : red;
    position : absolute;
    top : 10px; left : 10px;
}

```

CADRE et **BALLE** supprimés du texte HTML



Lien entre HTML et CSS :

```
<html>
  <head>
    <title>Jeu de Balle</title>
  </head>
  <body>
    <h1>Balle Rebondissante</h1>
    <div>
      CADRE
      <div> BALLE </div>
    </div>
  </body>
</html>
```

Lien entre HTML et CSS :

```
<html>
  <head>
    <title>Jeu de Balle</title>
    <link rel="stylesheet" href="style.css" type="text/css">
  </head>
  <body>
    <h1>Balle Rebondissante</h1>
    <div>
      CADRE
      <div> BALLE </div>
    </div>
  </body>
</html>
```


Lien entre HTML et CSS :

```
<html>
  <head>
    <title>Jeu de Balle</title>
    <link rel="stylesheet" href="style.css" type="text/css">
  </head>
  <body>
    <h1>Balle Rebondissante</h1>
    <div id="cadre">
      CADRE
      <div id="balle"> BALLE </div>
    </div>
  </body>
</html>
```

Lien entre HTML et CSS :

```
<html>
  <head>
    <title>Jeu de Balle</title>
    <link rel="stylesheet" href="style.css" type="text/css">
  </head>
  <body>
    <h1>Balle Rebondissante</h1>
    <div id="cadre">
      <div id="balle"> </div>
    </div>
  </body>
</html>
```

HTML : HyperText Markup Language

Langage à Balises pour l'HyperTexte

Balises pré-définies (tag)

html, head, link, body, h1, div,... etc.

HTML : HyperText Markup Language

Langage à Balises pour l'HyperTexte

Balises pré-définies (tag)

html, head, link, body, h1, div,... etc.

Structure : Texte écrit entre deux balises de même nom

`<html>` balise ouvrante

... ici du Texte éventuellement encadré par d'autres balises ...

`</html>` balise fermante

HTML : HyperText Markup Language

Langage à Balises pour l'HyperTexte

Balises pré-définies (tag)

html, head, link, body, h1, div,... etc.

Structure : Texte écrit entre deux balises de même nom

`<html>` balise ouvrante

... ici du Texte éventuellement encadré par d'autres balises ...

`</html>` balise fermante

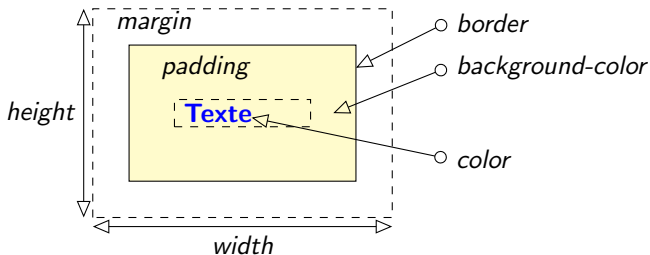
Liste Attribut-Valeur dans les balises ouvrantes

`<div id="balle">`

CSS : Cascading Style Sheets

Feuilles de style en cascade

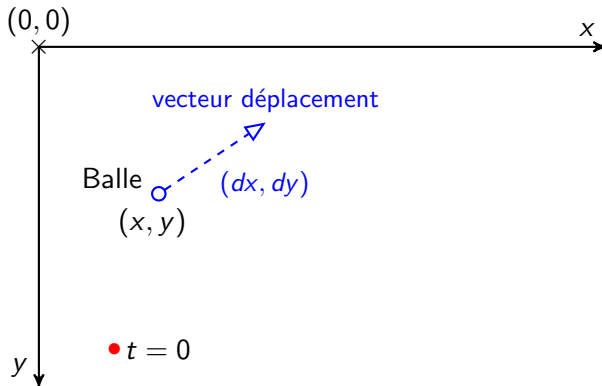
Géométrie des cases de texte : Le texte de chaque élément de la structure est affiché dans un rectangle paramétrable :



```
#balle { width: 100px; height: 100px; background-color: red;
position: absolute; top: 10px; left: 10px; }
```

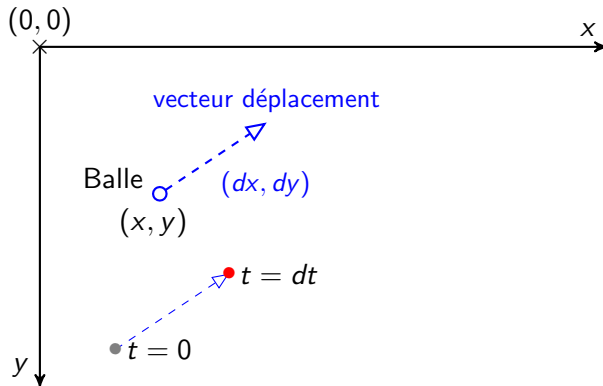
Animation : Affichage image par image

Une Animation consiste à changer périodiquement l'image affichée



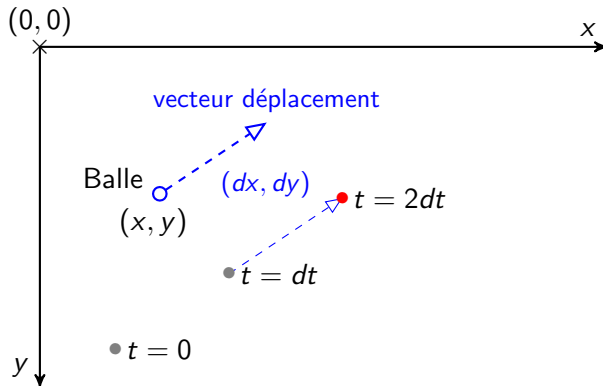
Animation : Affichage image par image

Une Animation consiste à changer périodiquement l'image affichée



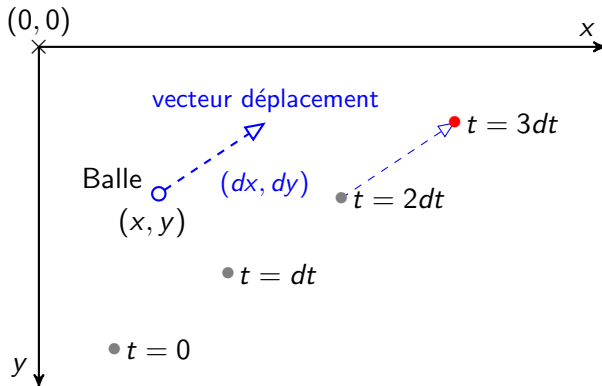
Animation : Affichage image par image

Une Animation consiste à changer périodiquement l'image affichée



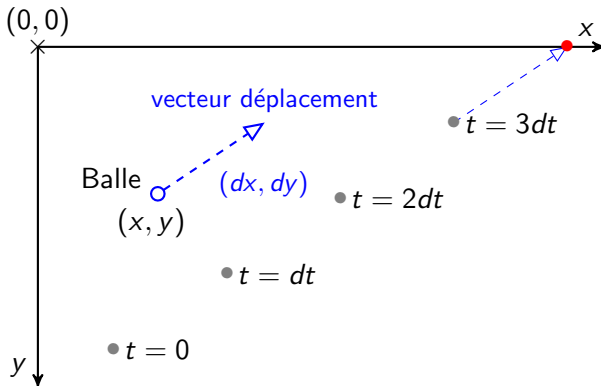
Animation : Affichage image par image

Une Animation consiste à changer périodiquement l'image affichée

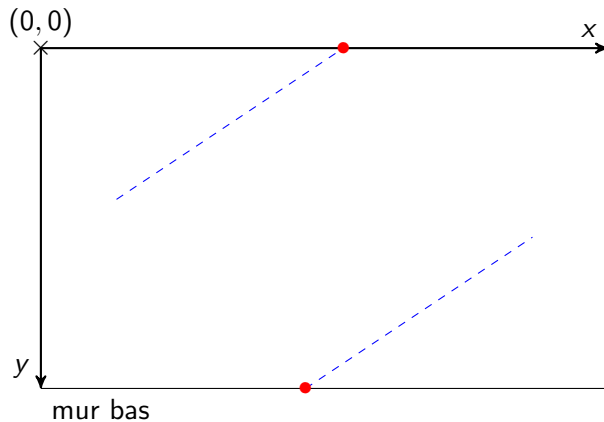


Animation : Affichage image par image

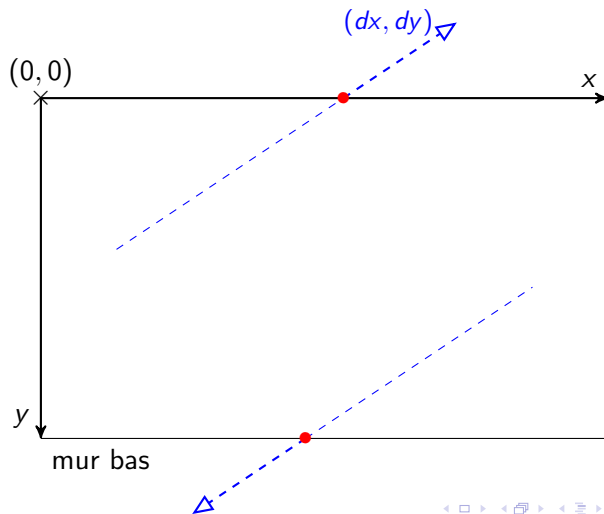
Une Animation consiste à changer périodiquement l'image affichée



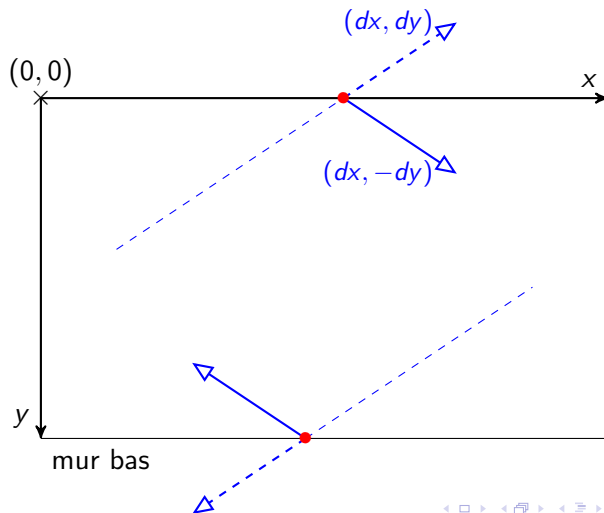
Symétrie par rapport à un axe horizontal



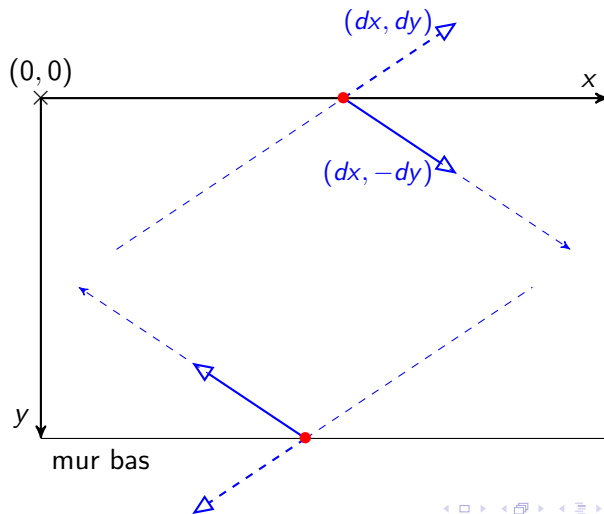
Symétrie par rapport à un axe horizontal



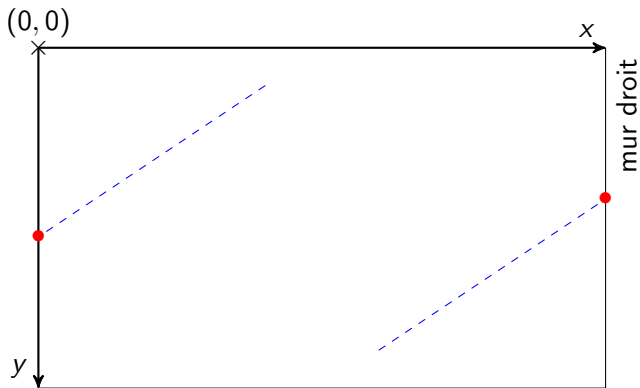
Symétrie par rapport à un axe horizontal



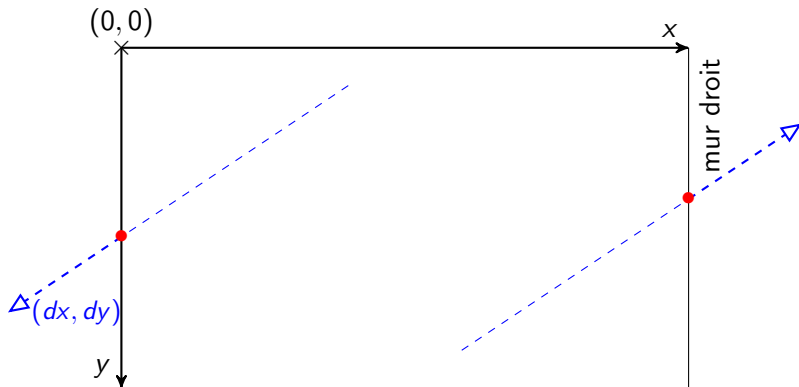
Symétrie par rapport à un axe horizontal



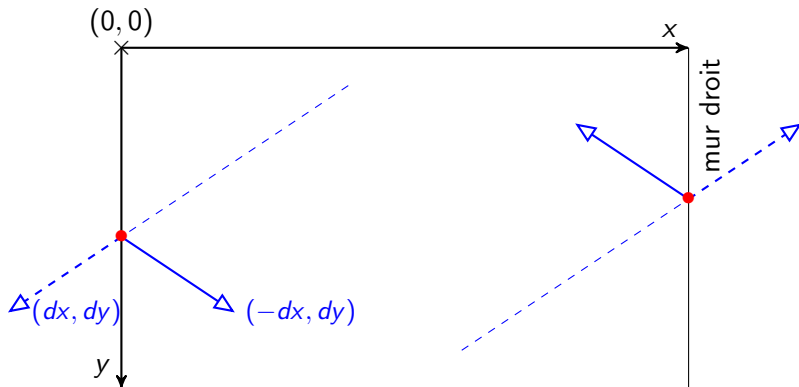
Symétrie par rapport à un axe vertical



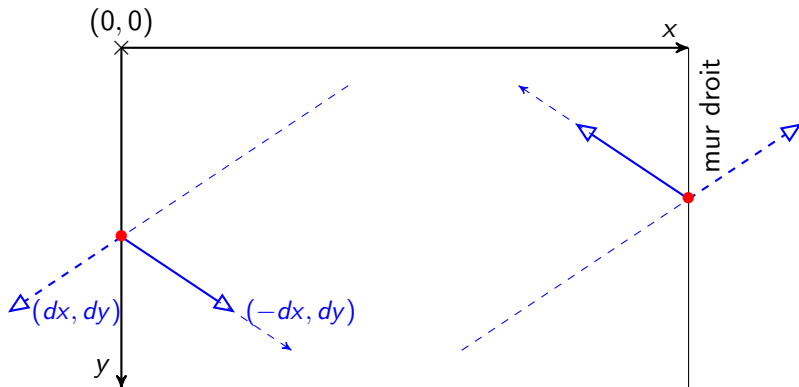
Symétrie par rapport à un axe vertical



Symétrie par rapport à un axe vertical



Symétrie par rapport à un axe vertical



Position, Vitesse, Temps, Mouvement

Mouvement

Le mouvement de la balle sur la page
sera simulé par un affichage image par image

Position, Vitesse, Temps, Mouvement

Mouvement

Le mouvement de la balle sur la page
sera simulé par un affichage image par image

Temps

Le temps écoulé entre deux images sera le nombre dt

Position, Vitesse, Temps, Mouvement

Mouvement

Le mouvement de la balle sur la page sera simulé par un affichage image par image

Temps

Le temps écoulé entre deux images sera le nombre dt

Vitesse

La vitesse est une variation de *position* par rapport au *temps*
le *déplacement* sera représentée par le vecteur (dx, dy)

Position, Vitesse, Temps, Mouvement

Mouvement

Le mouvement de la balle sur la page sera simulé par un affichage image par image

Temps

Le temps écoulé entre deux images sera le nombre dt

Vitesse

La vitesse est une variation de *position* par rapport au *temps*
le *déplacement* sera représentée par le vecteur (dx, dy)

Position

La *position* est représentée par les coordonnées (x, y)

JavaScript : Initialisation

«code.js»

```
// initialisation lors du chargement de la page HTML
```

```
div_balle = document.getElementById("balle");
```


JavaScript : Initialisation

«code.js»

```
// initialisation lors du chargement de la page HTML
```

```
div_balle = document.getElementById("balle");  
timer = setInterval(avancerBalleDunPas, dt);
```

JavaScript : Initialisation

«code.js»

```
var div_balle; // <div> de la balle
var timer;     // pour l'animation avec setInterval
```

```
// initialisation lors du chargement de la page HTML
function init() {
    div_balle = document.getElementById("balle");
    timer = setInterval(avancerBalleDunPas, dt);
}
```

JavaScript : Initialisation

«code.js»

```
var div_balle; // <div> de la balle
var timer;     // pour l'animation avec setInterval

function avancerBalleDunPas() {
    . . .
}

// initialisation lors du chargement de la page HTML
function init() {
    div_balle = document.getElementById("balle");
    timer = setInterval(avancerBalleDunPas, dt);
}
```

JavaScript : Animation de la Balle

«code.js»

```
// appel tous les dt milli secondes  
function avancerBalleDunPas() {
```

```
}
```

JavaScript : Animation de la Balle

«code.js»

```
// appel tous les dt milli secondes
function avancerBalleDunPas() {
    x = x + dx ;
    y = x + dy ;

}
```

JavaScript : Animation de la Balle

«code.js»

```
// appel tous les dt milli secondes
function avancerBalleDunPas() {
    x = x + dx;
    y = x + dy;
    if (x < 0 || x > murDroit) {
        dx = -dx; // rebond horizontal
    }
}
```

JavaScript : Animation de la Balle

«code.js»

```
// appel tous les dt milli secondes
function avancerBalleDunPas() {
  x = x + dx;
  y = y + dy;
  if (x < 0 || x > murDroit) {
    dx = -dx; // rebond horizontal
  } else if (y < 0 || y > murBas) {
    dy = -dy; // rebond vertical
  }
}
}
```

JavaScript : Animation de la Balle

«code.js»

```
// appel tous les dt milli secondes
function avancerBalleDunPas() {
    x = x + dx;
    y = y + dy;
    if (x < 0 || x > murDroit) {
        dx = -dx; // rebond horizontal
    } else if (y < 0 || y > murBas) {
        dy = -dy; // rebond vertical
    } else { // on change la position de la balle
        div_balle.style.left = x + "px"; // left : 5px
        div_balle.style.top = y + "px"; // top : 9px
    }
}
```


JavaScript : Déclaration des variables

«code.js»

```
var div_balle; // <div> de la balle  
var timer;     // pour l'animation avec setInterval
```

JavaScript : Déclaration des variables

«code.js»

```
var div_balle ; // <div> de la balle
var timer ;     // pour l'animation avec setInterval

var x = 250 ;   // balle au centre du cadre
var y = 150 ;   // 1/2*(largeur_cadre, hauteur_cadre)
```

JavaScript : Déclaration des variables

«code.js»

```
var div_balle ; // <div> de la balle
var timer ;    // pour l'animation avec setInterval

var x = 250 ;  // balle au centre du cadre
var y = 150 ;  // 1/2*(largeur_cadre, hauteur_cadre)

var dx = 5 ;   // vecteur du mouvement de la balle
var dy = 2 ;   // par exemple (5, 2)
```

JavaScript : Déclaration des variables

«code.js»

```
var div_balle; // <div> de la balle
var timer;     // pour l'animation avec setInterval

var x = 250;   // balle au centre du cadre
var y = 150;   // 1/2*(largeur_cadre, hauteur_cadre)

var dx = 5;    // vecteur du mouvement de la balle
var dy = 2;    // par exemple (5, 2)

var murDroit = 500 - 100; // largeur_cadre - largeur_balle
var murBas = 300 - 100;  // hauteur_cadre - hauteur_balle
```

JavaScript : Déclaration des variables

«code.js»

```
var div_balle; // <div> de la balle
var timer;     // pour l'animation avec setInterval

var x = 250;   // balle au centre du cadre
var y = 150;   // 1/2*(largeur_cadre, hauteur_cadre)

var dx = 5;    // vecteur du mouvement de la balle
var dy = 2;    // par exemple (5, 2)

var murDroit = 500 - 100; // largeur_cadre - largeur_balle
var murBas = 300 - 100;   // hauteur_cadre - hauteur_balle

var dt = 20; // temps en milli secondes entre chaque pas
```

HTML : Appel JavaScript

«index.html»

```
<html>
  <head>
    <title>Jeu de Balle</title>
    <link rel="stylesheet" href="style.css" type="text/css">
  </head>
  <body>
    <h1>Balle Rebondissante</h1>
    <div id="cadre">
      <div id="balle"></div>
    </div>
  </body>
</html>
```

HTML : Appel JavaScript

«index.html»

```
<html>
  <head>
    <title>Jeu de Balle</title>
    <link rel="stylesheet" href="style.css" type="text/css">
    <script src="code.js" type="text/javascript"></script>
  </head>
  <body>
    <h1>Balle Rebondissante</h1>
    <div id="cadre">
      <div id="balle"></div>
    </div>
  </body>
</html>
```

HTML : Appel JavaScript

«index.html»

```
<html>
  <head>
    <title>Jeu de Balle</title>
    <link rel="stylesheet" href="style.css" type="text/css">
    <script src="code.js" type="text/javascript"></script>
  </head>
  <body onload="init()">
    <h1>Balle Rebondissante</h1>
    <div id="cadre">
      <div id="balle"></div>
    </div>
  </body>
</html>
```


HTML : Clic sur la Balle

«index.html»

```
<html>
  <head>
    <title>Jeu de Balle</title>
    <link rel="stylesheet" href="style.css" type="text/css">
    <script src="code.js" type="text/javascript"></script>
  </head>
  <body onload="init()">
    <h1>Balle Rebondissante</h1>
    <div id="cadre">
      <div id="balle"></div>
    </div>
  </body>
</html>
```

HTML : Clic sur la Balle

«index.html»

```
<html>
  <head>
    <title>Jeu de Balle</title>
    <link rel="stylesheet" href="style.css" type="text/css">
    <script src="code.js" type="text/javascript"></script>
  </head>
  <body onload="init()">
    <h1>Balle Rebondissante</h1>
    <div id="cadre">
      <div id="balle" class="mobile"></div>
    </div>
  </body>
</html>
```

HTML : Clic sur la Balle

«index.html»

```
<html>
  <head>
    <title>Jeu de Balle</title>
    <link rel="stylesheet" href="style.css" type="text/css">
    <script src="code.js" type="text/javascript"></script>
  </head>
  <body onload="init()">
    <h1>Balle Rebondissante</h1>
    <div id="cadre">
      <div id="balle" class="mobile" onclick="clicBalle()"></div>
    </div>
  </body>
</html>
```

CSS : Modification de la Balle

«style.css»

```
.mobile { background-color : red; }
```



CSS : Modification de la Balle

«style.css»

```
.mobile { background-color : red; }
```



```
.immobile {  
    background-color : yellow;  
    border : 8px black dashed;  
}
```



CSS : Modification de la Balle

«style.css»

```
#balle {  
    position : absolute;  
    width : 200px;    height : 200px;  
}
```

```
.mobile { background-color : red; }
```



```
.immobile {  
    background-color : yellow;  
    border : 8px black dashed;  
}
```



JavaScript : Réaction de la Balle

«code.js»

```
// appel lors d'un clic sur la balle
function clicBalle() {

}
```

JavaScript : Réaction de la Balle

«code.js»

```
// appel lors d'un clic sur la balle
function clicBalle() {
    if (div_balle.className === "mobile") {
        clearInterval(timer); // Stop
        div_balle.className = "immobile";
    }
}
```


JavaScript : Réaction de la Balle

«code.js»

```
// appel lors d'un clic sur la balle
function clicBalle() {
  if (div_balle.className === "mobile") {
    clearInterval(timer); // Stop
    div_balle.className = "immobile";
  } else {
    timer = setInterval(avancerBalleDunPas, dt); // Go
    div_balle.className = "mobile";
  }
}
```

JavaScript : Réaction de la Balle

«code.js»

```
// appel lors d'un clic sur la balle
function clicBalle() {
  if (div_balle.className === "mobile") {
    clearInterval(timer); // Stop
    div_balle.className = "immobile";
  } else {
    timer = setInterval(avancerBalleDunPas, dt); // Go
    div_balle.className = "mobile";
  }
}
```

HTML CSS JavaScript

Technologies du Web côté Client

Tout Navigateur Internet interprète :

HTML

Langage de description du contenu textuel des pages

HTML CSS JavaScript

Technologies du Web côté Client

Tout Navigateur Internet interprète :

HTML

Langage de description du contenu textuel des pages

CSS

Langage de description de l'aspect visuel de l'affichage

HTML CSS JavaScript

Technologies du Web côté Client

Tout Navigateur Internet interprète :

HTML

Langage de description du contenu textuel des pages

CSS

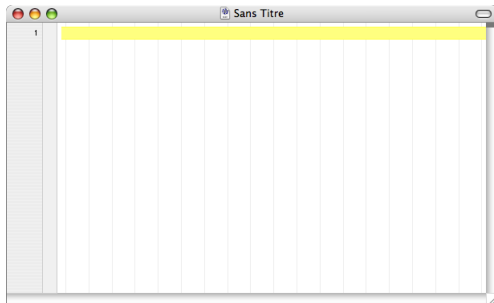
Langage de description de l'aspect visuel de l'affichage

JavaScript

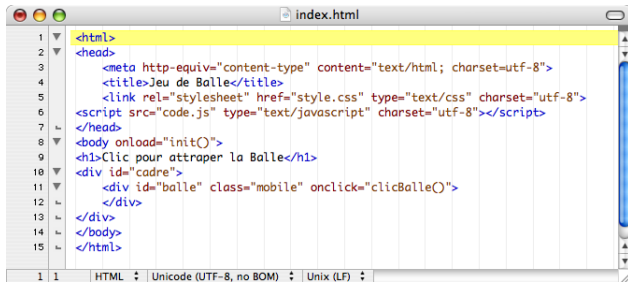
Langage de programmation des interactions du client sur l'interface

Vous pouvez le faire : Édition des codes sources

Il vous suffit d'un éditeur de texte pour créer trois fichiers



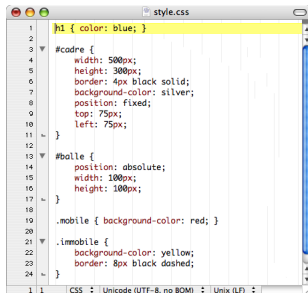
Vous pouvez le faire : Édition des codes sources



```
1 <html>
2 <head>
3   <meta http-equiv="content-type" content="text/html; charset=utf-8">
4   <title>Jeu de Balle</title>
5   <link rel="stylesheet" href="style.css" type="text/css" charset="utf-8">
6   <script src="code.js" type="text/javascript" charset="utf-8"></script>
7 </head>
8 <body onload="init()">
9   <h1>Clic pour attraper la Balle</h1>
10  <div id="cadre">
11    <div id="balle" class="mobile" onclick="clicBalle()">
12    </div>
13  </div>
14 </body>
15 </html>
```

- ▶ «index.html» la page HTML

Vous pouvez le faire : Édition des codes sources



```
1  h1 { color: blue; }
2
3  ▼ #cadre {
4      width: 500px;
5      height: 300px;
6      border: 4px black solid;
7      background-color: silver;
8      position: fixed;
9      top: 75px;
10     left: 75px;
11 }
12
13 ▼ #balle {
14     position: absolute;
15     width: 100px;
16     height: 100px;
17 }
18
19 .mobile { background-color: red; }
20
21 ▼ .immobile {
22     background-color: yellow;
23     border: 8px black dashed;
24 }
```

- ▶ «index.html» la page HTML
- ▶ «style.css» la feuille de style CSS

Vous pouvez le faire : Édition des codes sources

```

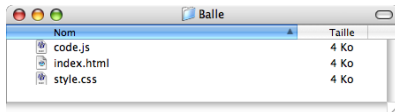
1  var div_balle; // <div> de la Balle
2  var timer;    // pour l'animation avec setInterval
3  var x = 250;  // balle au centre du cadre
4  var y = 150;  // 1/2*(largeur_cadre, hauteur_cadre)
5  var dx = 5;   // vecteur du mouvement de la balle
6  var dy = 2;   // par exemple (5, 2)
7  var dt = 20;  // temps en milli secondes entre chaque pas
8  var murDroit = 500 - 100; // largeur_cadre - largeur_balle;
9  var murBas = 300 - 100;  // hauteur_cadre - hauteur_balle;
10 // appel tous le dt milli secondes
11 function avancerBalleUnPas() {
12     x = x + dx;
13     y = y + dy;
14     if (x < 0 || x > murDroit) { // rebond horizontal
15         dx = -dx;
16     } else if (y < 0 || y > murBas) { // rebond vertical
17         dy = -dy;
18     } else { // on change la position de la balle
19         div_balle.style.left = x + "px";
20         div_balle.style.top = y + "px";
21     }
22 }
23 // appel lors d'un clic sur la balle
24 function clicBalle() {
25     if (div_balle.className === "mobile") {
26         clearInterval(timer); // Stop
27         div_balle.className = "immobile"; // on change l'aspect
28     } else {
29         timer = setInterval(avancerBalleUnPas, dt); // on repart
30         div_balle.className = "mobile"; // on change l'aspect
31     }
32 }
33 // initialisation lors du chargement de la page HTML
34 function init() {
35     div_balle = document.getElementById("balle");
36     timer = setInterval(avancerBalleUnPas, dt); // chaque instant dt
37 }

```

- ▶ «index.html» la page HTML
- ▶ «style.css» la feuille de style CSS
- ▶ «code.js» le code JavaScript

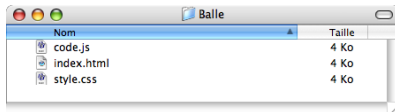
Vous pouvez le tester : Exécution du programme

- ▶ Placez ces 3 fichiers dans un même dossier : «Balle»



Vous pouvez le tester : Exécution du programme

- ▶ Placez ces 3 fichiers dans un même dossier : «Balle»



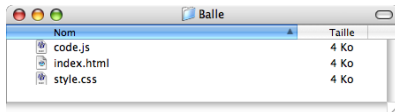
- ▶ Ouvrez alors «index.html» dans un Navigateur Internet.

Jouez :



Vous pouvez le tester : Exécution du programme

- ▶ Placez ces 3 fichiers dans un même dossier : «Balle»



- ▶ Ouvrez alors «index.html» dans un Navigateur Internet.

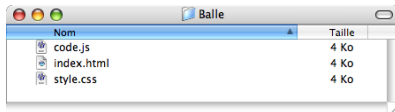
Jouez :



Clic

Vous pouvez le tester : Exécution du programme

- ▶ Placez ces 3 fichiers dans un même dossier : «Balle»



- ▶ Ouvrez alors «index.html» dans un Navigateur Internet.

Jouez :



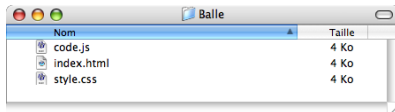
Clic



...

Vous pouvez le tester : Exécution du programme

- ▶ Placez ces 3 fichiers dans un même dossier : «Balle»



- ▶ Ouvrez alors «index.html» dans un Navigateur Internet.

Jouez :



Clic



...

Ensuite... donnez libre cours à votre imagination ...

Pour plus d'informations :

Vous trouverez les exemples présentés dans cet exposé sur le site :

Mathématiques et Informatique

<http://math.et.info.free.fr/>

... ainsi que d'autres exemples et des informations complémentaires

Et aussi de nombreux problèmes et exemples sur le site :

Atelier de Mathématiques «Laurent Schwartz»

<http://ateliermathematique.free.fr/>

