ゼロからわかる JavaScript 超入門　たのしいプログラミング

「河西朝雄著　技術評論社」

の補足記事

すでに紙媒体で出版した上記著作に関し頁数の関係から載せられなかった以下の内容を提供します。

１０章　各種 Visual ツールを作ってみよう
１１章　最終課題（各種ゲーム）

原稿としては 2009 年の脱稿時にできていてお蔵入りしていたものをここで公開することとします。

2012 年 12 月　河西　朝雄

１０章　各種 Visual ツールを作ってみよう

実際の Web ページで見かける各種 Visual ツールの作り方を説明します。

<input>タグで作るボタンでなく、イメージを使った Visual ボタン、ある領域にマウスが入ったらその領域の色が変わる地図、月単位のカレンダー、ページを切り替えるタブなどです。

本書の１章～７章はプログラミングの初心者が視覚的に興味のある題材を使って効率よく JavaScript や DynamicHTML の技術を学べるような構成になっています。このため DynamicHTML の使い方（ラケットゲーム、ドットアートなど）としては特殊な部類に入ります。この章で説明する内容は Web ページに応用できる本来の DynamicHTML の内容となっています。特にタブの例はスタイルシートの各プロパティを最も豊富に使っています。

10-1　Visual ボタン

　バーを 3 つのイメージ(normalbar.png,prevbar.png,nextbar.png)で切り替えます。前の写真部、センター部、次の写真部は 1 つのイメージに収まっているので、各部の領域を判定する場合は onMouseOver イベントでなく onMouseMove イベントで行います。イメージ外に出た場合は onMouseOut イベントで行います。

normalbar.png

prevbar.png

nextbar.png

「例題 10-1」

　バーへマウスが入ったときその位置で「前の写真」領域か「次の写真」領域かで対応するバーを表示します。

```
<html>
<head>
<script type="text/javascript">
    function over(event)
    {
        var obj=document.getElementById("bar");
        if (event.clientX<242)          //100+142   ←①
            obj.src="prevbar.png";
        else if (event.clientX>432) //100+332   ←②
            obj.src="nextbar.png";
        else   ←③
            obj.src="normalbar.png";
    }
    function out()   ←④
    {
        var obj=document.getElementById("bar");
        obj.src="normalbar.png";
    }
</script>
</head>
<body>
<img id="bar" src="normalbar.png" onMouseMove="over(event)" onMouseOut="out()"
```

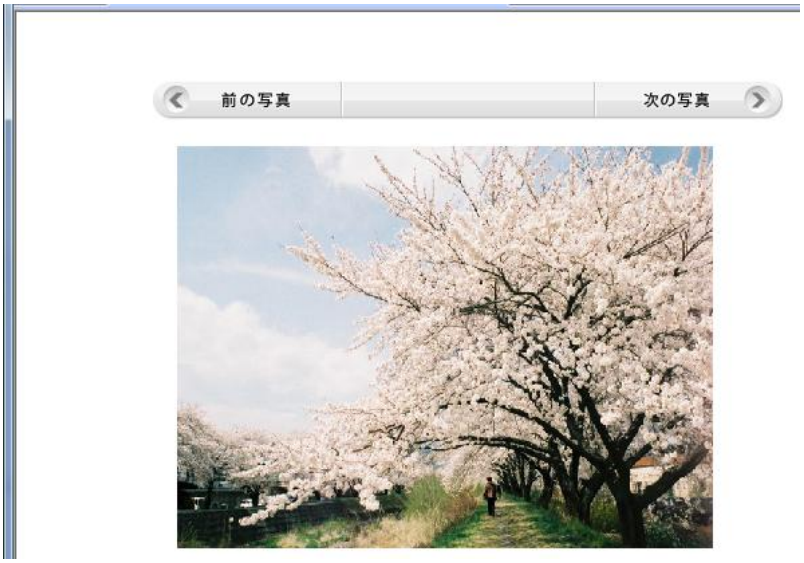
style="position:absolute;left:100;top:50">
</body>
</html>

①マウス位置が「前の写真位置」のとき
②マウス位置が「次の写真位置」のとき
③それ以外の位置のとき
④マウスがバーから離れたとき

「練習問題 10-1」
　バーのクリックで前の写真、次の写真を表示しなさい。写真のファイル名は配列 photo に格納するものとする。写真番号を Number,写真の枚数を Max で管理する。

```
<html>
<head>
<script type="text/javascript">
    var Number=0,Max=4;
    var photo=new
Array("balloon.jpg","flower.jpg","hikari.jpg","matsuri.jpg","sakura.jpg");
    function next(event)
    {
        if (event.clientX<242)
            Number--;
        else if (event.clientX>432)
            Number++;
        if (Number<0)
            Number=Max;
        if (Number>Max)
           Number=0;
        var obj=document.getElementById("srcimg");
        obj.src=photo[Number];
    }
    function over(event)
    {
        var obj=document.getElementById("bar");
```

```
        if (event.clientX<242)          //100+142
            obj.src="prevbar.png";
        else if (event.clientX>432) //100+332
            obj.src="nextbar.png";
        else
            obj.src="normalbar.png";
    }
    function out()
    {
        var obj=document.getElementById("bar");
        obj.src="normalbar.png";
    }
</script>
</head>
<body>
<img id="bar" src="normalbar.png" onClick="next(event)" onMouseMove="over(event)"
onMouseOut="out()" style="position:absolute;left:100;top:50">
<img src="balloon.jpg" id="srcimg"
style="position:absolute;left:120;top:100;width:400;height:300">
</body>
</html>
```



10-2　地図

地図を領域で分け、その領域にマウスが入ったらその領域の色を変えるようにします。

「例題 10-2」

日本地図を北海道から九州・沖縄の 9 つの領域に分け、その領域の色を変えた地図 map1.png～map9.png と、どこも色が変わっていない地図 map0.png を用意します。

```
<html>
<head>
<script type="text/javascript">
    function disp(event)
    {
        var x=event.clientX;
        var y=event.clientY;
        var obj=document.getElementById("map");
        if (290<x && x<380 && 10<y && y<80)   ←①
            obj.src="map1.png";
        else if (260<x && x<300 && 90<y && y<155)   ←②
            obj.src="map2.png";
        else if (230<x && x<260 && 164<y && y<200)
            obj.src="map3.png";
        else if (200<x && x<230 && 150<y && y<180)
            obj.src="map4.png";
        else if (188<x && x<222 && 174<y && y<203)
            obj.src="map5.png";
        else if (160<x && x<177 && 160<y && y<200)
            obj.src="map6.png";
        else if (95<x && x<150 && 160<y && y<180)
            obj.src="map7.png";
        else if (115<x && x<145 && 185<y && y<205)
            obj.src="map8.png";
        else if (60<x && x<90 && 170<y && y<220)
            obj.src="map9.png";
        else   ←③
            obj.src="map0.png";
    }
</script>
</head>
```

<body>
<img id="map" src="map0.png" style="position:absolute;left:0;top:0" onMouseMove="disp(event)">
</body>
</html>

①北海道の領域のとき
②東北の領域のとき
③どこの領域にも入らないとき

「練習問題 10-2」
例題 10-2 は地図の配置位置をクライアント画面の（0,0）と仮定してプログラムにしていましたが、地図の配置位置に依存しないで領域の判定ができるようにしなさい。地図の配置位置の top,left を取得し、そこからの位置で領域の判定を行う。

```
<html>
<head>
<script type="text/javascript">
    function disp(event)
    {
        var x=event.clientX;
        var y=event.clientY;
        var obj=document.getElementById("map");
        var top=parseInt(obj.style.top);
        var left=parseInt(obj.style.left);
        if (left+290<x && x<left+380 && top+10<y && y<top+80)
            obj.src="map1.png";
        else if (left+260<x && x<left+300 && top+90<y && y<top+155)
            obj.src="map2.png";
        else if (left+230<x && x<left+260 && top+164<y && y<top+200)
            obj.src="map3.png";
```

```
            else if (left+200<x && x<left+230 && top+150<y && y<top+180)
                obj.src="map4.png";
            else if (left+188<x && x<left+222 && top+174<y && y<top+203)
                obj.src="map5.png";
            else if (left+160<x && x<left+177 && top+160<y && y<top+200)
                obj.src="map6.png";
            else if (left+95<x && x<left+150 && top+160<y && y<top+180)
                obj.src="map7.png";
            else if (left+115<x && x<left+145 && top+185<y && y<top+205)
                obj.src="map8.png";
            else if (left+60<x && x<left+90 && top+170<y && y<top+220)
                obj.src="map9.png";
            else
                obj.src="map0.png";
    }
</script>
</head>
<body>
<img id="map" src="map0.png" style="position:absolute;left:100;top:50"
onMouseMove="disp(event)">
</body>
</html>
```

10-3　カレンダー

　カレンダーを作るには次のような手順で考えます。

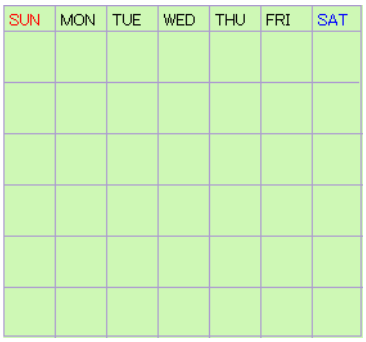
①月の 1 日の曜日を調べる。week に 0(日)～6(土)が得られる。

```
var Year=today.getFullYear();
var Month=today.getMonth();
var day=new Date(Year,Month,1);
var week=day.getDay();
```

②閏年か調べる。

```
if ((Year % 4)==0 && (Year % 100)!=0 || (Year % 400)==0)
    mt[1]=29;
else
    mt[1]=28;
```

③カレンダーとして表示。次のような calendar.png 上に 1～31 を表示する。表示位置は①で求めた week の値を開始位置として使う。



「例題 10-3」

　今日の月のカレンダーを作ります。

```
<html>
<head>
<script type="text/javascript">
    function mannen()
    {
        var mt=new Array(31,28,31,30,31,30,31,31,30,31,30,31);
        var i,x,y,tag="";
        var today=new Date();
        var Year=today.getFullYear();
        var Month=today.getMonth();
        var day=new Date(Year,Month,1);   ←①
        var week=day.getDay();   ←②
```

```
        if ((Year % 4)==0 && (Year % 100)!=0 || (Year % 400)==0)   ←③
            mt[1]=29;
        else
            mt[1]=28;

        tag+="<font    size='3'    style='position:absolute;left:60;top:20'>"+Year+"   年
"+(Month+1)+"月</font>";   ←④
        x=week*40+45;y=65;   ←⑤
        for (i=1;i<=mt[Month];i++){   ←⑥
            tag+="<font                                                    size='3'
style='position:absolute;left:"+x+";top:"+y+"'>"+i+"</font>";   ←⑦
            x+=40;   ←⑧
            if ((week+i)%7==0){   ←⑨
                x=45;
                y+=40;
            }
        }
        var obj=document.getElementById("calendar");
        obj.innerHTML=tag;   ←⑩
    }
</script>
</head>
<body onLoad="mannen()">
<img src="leftbtn.png" style="position:absolute;left:40;top:18">
<img src="rightbtn.png" style="position:absolute;left:160;top:18">
<img id="n1" src="calendar.png" style="position:absolute;left:40;top:40">
<div id="calendar"></div>
</body>
</html>
```

①Year 年 Month 月 1 日の Date オブジェクトの作成
②Year 年 Month 月 1 日の曜日（0：日～6：土）の取得
③閏年なら 2 月の日を 29 にする
④年、月の表示。Month は 0（1 月）～11（12 月）の値なので、+1 した値を表示
⑤1 日の表示位置を x,y に設定
⑥1 日から月の終わりの日まで繰り返す
⑦日を示す数字を（x,y）位置に表示
⑧表示位置を横に進める
⑨土曜日なら、次の行先頭に移す
⑩tag に追加してきた HTML タグを タグの内部 HTML として置き換える

「練習問題 9-3」

左ボタンで前の月を表示し、右ボタンで次の月を表示できるようにしなさい。これに伴い今日の月を表示する場合 init 関数を最初に呼び出し Year と Month を求める。back 関数で月を前に戻し、next 関数で月を次に進める。

```
<head>
<script type="text/javascript">
    var Year,Month;
    function init()
    {
        var today=new Date();
        Year=today.getFullYear();
        Month=today.getMonth();
        mannen();
    }
    function back()
    {
        if (Month==0){
```

```
            Year--;
            Month=11;
        }
        else
            Month--;
        mannen();
    }
    function next()
    {
        if (Month==11){
            Year++;
            Month=0;
        }
        else
            Month++;
        mannen();
    }
    function mannen()
    {
        var mt=new Array(31,28,31,30,31,30,31,31,30,31,30,31);
        var i,x,y,tag="";
        var day=new Date(Year,Month,1);
        var week=day.getDay();

        if ((Year % 4)==0 && (Year % 100)!=0 || (Year % 400)==0)
            mt[1]=29;
        else
            mt[1]=28;

        tag+="<font size='3' style='position:absolute;left:60;top:20'>"+Year+" 年
"+(Month+1)+"月</font>";
        x=week*40+45;y=65;
        for (i=1;i<=mt[Month];i++){
            tag+="<font size='3'
style='position:absolute;left:"+x+";top:"+y+"'>"+i+"</font>";
            x+=40;
```

```
                if ((week+i)%7==0){
                    x=45;
                    y+=40;
                }
            }
            var obj=document.getElementById("calendar");
            obj.innerHTML=tag;
        }
</script>
</head>
<body onLoad="init()">
<img src="leftbtn.png" style="position:absolute;left:40;top:18" onClick="back()">
<img src="rightbtn.png" style="position:absolute;left:160;top:18" onClick="next()">
<img id="n1" src="calendar.png" style="position:absolute;left:40;top:40">
<div id="calendar"></div>
</body>
</html>
```

10-4　タブ

　タブのクリックでページを切り替えます。タブは<a>タグ、ページは タグにスタイルシートを指定して作ります。

　スタイルシートは<style type="text/css">～</style>に記述します。

a {}の中に<a>タグのスタイルシートを、div {}の中に タグのスタイルシートを記述します。さらに a.tab1 {}の中に指定したスタイルシートは<a class="tab1">のように class を指定した個々の<a>タグに適用されます。

　スタイルシートの各プロパティの意味は以下の通りです。

display：表示方法。block,inline,list-item,none

overflow：オーバーフローした内容の表示方法。none,clip,scroll

clear：文字の回り込みのクリア。none,left,right,both

float：文字の回り込み。left,right,none

text-align：テキストの配置。left,right,center,justify

border：枠の幅、スタイル、色

padding：枠と内容の上、右、下、左間隔

color：文字色

background-color：背景色

width：スタイルシートの幅

height：スタイルシートの高さ

指定する長さの単位は以下が指定できます。

em　　使用されている文字の高さ

pt　　ポイント

px　　ピクセル

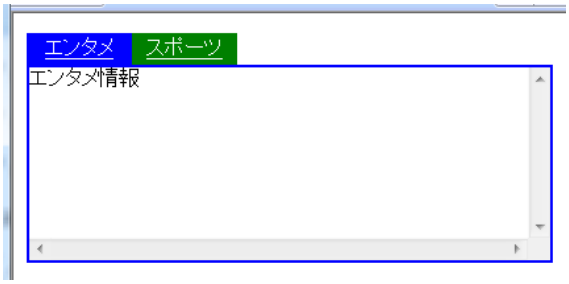
「例題 10-4」

　タブのクリックでページを切り替える関数 change を作ります。タブ上にマウスが入るとハンドマークにするために<a>タグを用いているのですが、<a>タグのクリックでリンク先にジャンプさせないために、onClick イベントの呼び出しを行った後の「return false;」は必要です。

```
<html>
<head>
<script type="text/javascript">
    function change(tab) {   ←①
        document.getElementById("tab1").style.display="none";
        document.getElementById("tab2").style.display="none";
```

```
            document.getElementById(tab).style.display="block";
        }
</script>
<style type="text/css">
a {   ←②
    display:block;width:5em;float:left;
    padding:3px;text-align:center;color:white;
}
a.tab1 {background-color:blue;}   ←③
a.tab2 {background-color:green;}

div {   ←④
    width:400px;height:150px;clear:left;overflow:scroll;
}
div.page1 { border:2px solid blue;}   ←⑤
div.page2 { border:2px solid green;}
</style>
</head>
<body onLoad="change('tab1')">
<a href="" class="tab1" onClick="change('tab1');return false;">エンタメ</a>   ←⑥
<a href="" class="tab2" onClick="change('tab2');return false;">スポーツ</a>
<div id="tab1" class="page1">   ←⑦
    エンタメ情報
</div>
<div id="tab2" class="page2">
    スポーツ情報
</div>
</body>
</html>
```

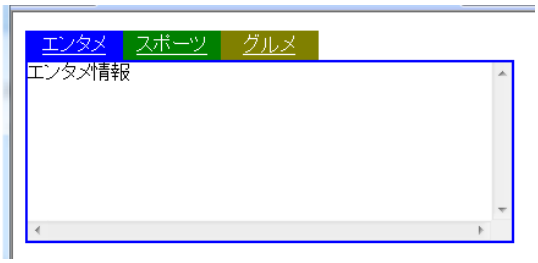


①クリックしたタブのページだけを表示する関数。display プロパティが”block”のページだ

けが表示される。他のページの display プロパティを”none”にすることで、そのページの縦・横サイズは 0 になるので、複数のページを同じ位置に重ね合わせることができる。
②タブの共通形状。float:left の指定でタブを横に配置できる。
③各タブの個々の形状（背景色）。
④ページの共通形状。clear:left で横表示をクリアしてタブの下に配置することができる。
⑤各ページの個々の形状（枠の色）。
⑥タブを 2 つ配置
⑦ページを 2 つ配置
「練習問題 10-4」
タブを 3 つに増やしなさい。タブの色は olive とする。

```
<html>
<head>
<script type="text/javascript">
    function change(tab) {
        document.getElementById("tab1").style.display="none";
        document.getElementById("tab2").style.display="none";
        document.getElementById("tab3").style.display="none";
        document.getElementById(tab).style.display="block";
    }
</script>
<style type="text/css">
a {
    display:block;width:5em;float:left;
    padding:3px;text-align:center;color:white;
}
a.tab1 {background-color:blue;}
a.tab2 {background-color:green;}
a.tab3 {background-color:olive;}
div {
    width:400px;height:150px;clear:left;overflow:scroll;
}
div.page1 { border:2px solid blue;}
div.page2 { border:2px solid green;}
div.page3 { border:2px solid olive;}
</style>
</head>
```

```
<body onLoad="change('tab1')">
<a href="" class="tab1" onClick="change('tab1');return false;">エンタメ</a>
<a href="" class="tab2" onClick="change('tab2');return false;">スポーツ</a>
<a href="" class="tab3" onCclick="change('tab3');return false;">グルメ</a>
<div id="tab1" class="page1">
    エンタメ情報
</div>
<div id="tab2" class="page2">
    スポーツ情報
</div>
<div id="tab3" class="page3">
    グルメ情報
</div>
</body>
</html>
```



「章末問題 10-1」

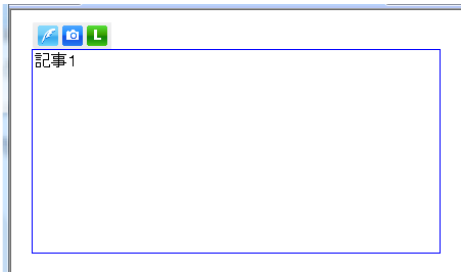
ツールバーのイメージを 3 領域に区分し、各領域にマウスが入れば、カーソルをハンドマークにし、ページに表示する内容を領域に合わせて変えなさい。領域のイメージ内の x 座標は 5,29,53 とし、幅は 20 とする。イメージの配置位置（left,top）から領域の範囲を計算する。

5　29　53
↓　↓　↓

```
<head>
<script type="text/javascript">
    function disp(event)
    {
        var map=document.getElementById("map");
        var page=document.getElementById("page");
        var left=parseInt(map.style.left);
        var top=parseInt(map.style.top);
        var x=event.clientX;
        var y=event.clientY;

        if (left+5<x && x<left+25){
            map.style.cursor="hand";
            page.innerHTML="記事 1 ";
        }
        else if (left+29<x && x<left+49){
            map.style.cursor="hand";
            page.innerHTML="記事 2 ";
        }
        else if (left+53<x && x<left+73){
            map.style.cursor="hand";
            page.innerHTML="記事 3 ";
        }
        else {
            map.style.cursor="default";
            page.innerHTML="";
        }
```

```
            if (y<=top+2 || top+28<=y){   // 上下に離れたとき
                map.style.cursor="default";
                page.innerHTML="";
            }
        }
</script>
</head>
<body>
<img id="map" src="toolbar.png" style="position:absolute;left:20;top:10"
 onMouseMove="disp(event)">
<div id="page" style="position:absolute;left:20;top:38;width:400;height=200;
 border-style:solid;border-color:blue;border-width:1;padding:2 2 2 2"></div>
</body>
</html>
```



「章末問題 10-2」

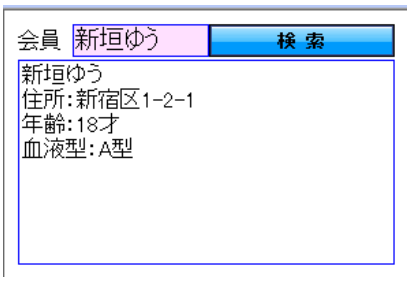
テキストボックスに入力した名前と一致する人のデータを検索して表示します。テキストボックスに適用するスタイルシートを.box { }に記述し、結果を表示する領域に適用するスタイルシートを.page { }に記述します。会員のデータは２次元配列 girl[][]に名前、住所、年齢、血液型が格納されています。

オブジェクトの上をマウスが通過したときにハンドマークにする簡単な方法は「onMouseOver="this.style.cursor='hand'"」です。

```
<html>
<head>
<style type="text/css">
.box {
      width:100;height:24;
      border-style:solid;border-color:blue;border-width:1;
      background-color:#ffe0ff;
```

```
        font-size:18;
}
.page {
        width:276;height=150;
        border-style:solid;border-color:blue;border-width:1;
        padding:2 2 2 2;
}
</style>
<script type="text/javascript">
        function search()
        {
                var girl=new Array(3);
                girl[0]=new Array("新垣ゆう","新宿区 1-2-1","18 才","A 型");
                girl[1]=new Array("長沢みちる","千代田区 1-2-1","22 才","B 型");
                girl[2]=new Array("綾瀬はるな", "港区 1-2-1","19 才","O 型");
                var namae=document.getElementById("namae");
                var result=document.getElementById("result");
                for (var i=0;i<girl.length;i++){
                        if (girl[i][0]==namae.value){
                                result.innerHTML="名前："+girl[i][0]+"<br>"+
                                                "住所："+girl[i][1]+"<br>"+
                                                "年齢："+girl[i][2]+"<br>"+
                                                "血液型："+girl[i][3]+"<br>";
                                return;
                        }
                }
                result.innerHTML="見つかりませんでした<br>";
        }
</script>
</head>
<body>
<form style="position:absolute;top:10;left:10">
<div style="position:absolute;top:4;left:0;width:40">会員</div>
<input id="namae" class="box" type="text"    style="position:absolute;top:0;left:40">
<img src="searchbar.png" style="position:absolute;top:0;left:140"
 onMouseOver="this.style.cursor='hand'" onClick="search()">
```

</form>
</body>
</html>



１１章　最終課題（各種ゲーム）

９章のリバーシゲームで実用的なプログラムを作るスキルが付いたと思います。この章ではさらに、自分で考えて実用的に使えるプログラムを作るためのヒントとなるサンプルとして、ブロックくずし、マインスイーパー、迷路の３例題を説明します。

ブロックくずしは６章のラケットゲームを元にそこにブロックを置きます。マインスーパーは、練習問題 7-2、リバーシーゲームのマスを開ける手法を応用します。迷路はここまで説明しなかったテクニックがいくつかあります。マインスーパーと迷路では再帰を使用しています。

発展課題は各例題に対し、追加または変更された箇所（赤色）とその周辺部のみを掲載しています。プログラムの全リストは技術評論社の以下の Web ページで入手することができます。

11-1　ブロックくずし

6 章のラケットゲームにブロックを置いたブロックくずしを作ります。

ブロックを 4×6 個配置し、ブロックの情報（0:なし、1:あり）を 2 次元配列 M[][]に格納します。ブロックの高さを 10、幅を 40 とし配置開始位置を(100,100)から y 方向は 20、x 方向は 50 単位で配置します。

ボールの位置を By,Bx とすると、その位置にあるブロックに対応する配列の添字 i,j は次式で得られます。次式では By,Bx が 100 未満の場合も i,j が 0 になってしまい、ブロックのないところでもあると判断されてしまうので、if 文で By,Bx は 100 以上という条件を付けてあります。

```
i=parseInt((By-100)/20);// ボール位置から、ブロックの位置を計算する
j=parseInt((Bx-100)/50);
if (100<=Bx && 100<=By && 0<=i && i<=3 && 0<=j && j<=5 &&
M[i][j]==1){   // ブロックがあれば
```

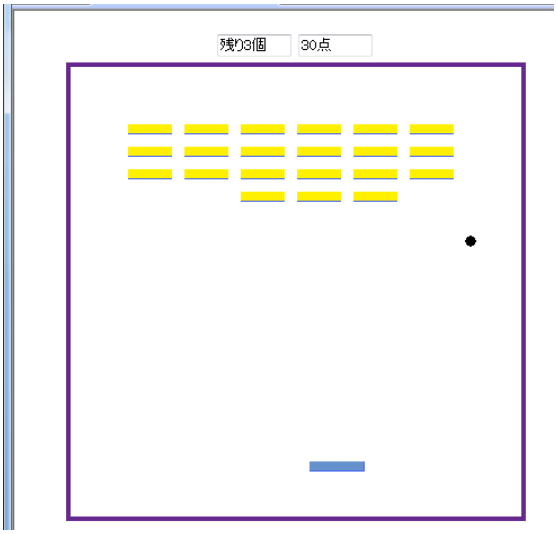
ボールが上に移動している場合と下に移動している場合でブロックに当たったと判定する位置は異なります。正確に判定するにはもう少し細かな条件を付ける必要がありますが、ここでは大ざっぱな判定をしています。

・block1.html

```
<html>
<head>
<script type="text/javascript">
    var By=200,Bx=100,Dy=10,Dx=10;   // ボール位置と移動量
    var Ry=400,Rx=200;                    // ラケットの位置
    var Point=0,BallN=3,TimeID;       // 得点、ボール残数、タイム ID
    var M=new Array(4);                   // ブロック情報配列
    function move()
    {
        var i,j,n;
        TimeID=setTimeout("move()",100);
        var ball=document.getElementById("ball");
        By+=Dy;Bx+=Dx;
        ball.style.top=By;ball.style.left=Bx;
        if (Bx<=50 || Bx+10>=450)
            Dx=-Dx;
        if (By<=50 || (Rx-25-10<=Bx && Bx<=Rx-25+50 && By+10==Ry))
```

```
            Dy=-Dy;
        i=parseInt((By-100)/20);  // ボール位置から、ブロックの位置を計算する
        j=parseInt((Bx-100)/50);
        if (100<=Bx && 100<=By && 0<=i && i<=3 && 0<=j && j<=5 &&
M[i][j]==1){  // ブロックがあれば
            n=i*6+j;        // ブロックの ID を求める
            document.getElementById("block"+n).style.visibility="hidden";  // 非表示
にする
            M[i][j]=0;                          // 配列情報を 0 にすることで、ブロックな
し状態を示す
            Dy=-Dy;
            Point+=10;                          // 得点の加算と表示
            document.getElementById("PointMsg").value=Point+"点";
        }
        if (By>420){
            BallN--;                    // ボールの残数を減らす
            document.getElementById("BallMsg").value="残り"+BallN+"個";
            clearTimeout(TimeID);  // ボールを停止する
            if (BallN>0){
                Bx=parseInt(Math.random()*30)*10+50;By=200;
                setTimeout("move()",1000);  // 1 秒後に再開
            }
        }
    }
    function racket(event)
    {
        var bar=document.getElementById("bar");
        Rx=event.clientX;
        if (50<=Rx-25 && Rx+25<=450)
            bar.style.left=Rx-25;
    }
    function init()
    {
        var i,j,n=0,tag="";
        var block=document.getElementById("block");
        window.resizeTo(550,700);  // ウィンドウサイズを変更
```

```
        for (i=100;i<=160;i+=20){   //   ブロックの配置
            for (j=100;j<=350;j+=50){
                tag+="<img id='block"+n+"' src='blockbar.png'
style='position:absolute;left:"+j+";top:"+i+"'>";
                n++;
            }
        }
        block.innerHTML=tag;
        for (i=0;i<4;i++){          // 4 行 6 列のブロックの情報を表す 2 次元配列
            M[i]=new Array(6);
            for (j=0;j<6;j++){
                M[i][j]=1;          // i 行 j 列のブロックがあれば 1
            }
        }
        Bx=parseInt(Math.random()*30)*10+50;By=200;
        move();
    }
</script>
</head>
<body onMouseMove="racket(event)">
<form style="position:absolute;left:180;top:20">
<input name="BallMsg" type="text" size="10" value="残り 3 個">
<input name="PointMsg" type="text" size="10" value="0 点">
</form>
<img src="wall.png" style="position:absolute;left:46;top:46">
<img id="ball" src="ball.png" style="position:absolute;left=-10;top=0">
<img id="bar" src="bar.png" style="position:absolute;left:200;top:400" onClick="init()">
<div id="block"></div>
</body>
</html>
```



「発展課題」

次の機能を追加しなさい。

・ボールが下に移動しているときにブロックに当たったという判断をブロックの直前位置にしなさい。Dy が正のときは配列の仮想位置を y 方向に-10 した位置とする。

・全面クリア（Point が 240 の倍数）するたびにブロックを再表示し、タイムアウト時間を-20 にしてボールの移動スピードを速くする。

・block2.html

```
var Level=100;
function move()
{
    var i,j,n;
    TimeID=setTimeout("move()",Level);
    var ball=document.getElementById("ball");
    By+=Dy;Bx+=Dx;
    ball.style.top=By;ball.style.left=Bx;
    if (Bx<=50 || Bx+10>=450)
        Dx=-Dx;
    if (By<=50 || (Rx-25-10<=Bx && Bx<=Rx-25+50 && By+10==Ry))
        Dy=-Dy;
    if (Dy<0)
        i=parseInt((By-100)/20);  // ボール位置から、ブロックの位置を計算する
    else
        i=parseInt((By-90)/20);
    j=parseInt((Bx-100)/50);
    if (100<=Bx && 90<=By && 0<=i && i<=3 && 0<=j && j<=5 &&
```

```
M[i][j]==1){  // ブロックがあれば
            n=i*6+j;        // ブロックの ID を求める
            document.getElementById("block"+n).style.visibility="hidden";  // 非表示にする
            M[i][j]=0;                          // 配列情報を 0 にすることで、ブロックなし状態を示す
            Dy=-Dy;
            Point+=10;                          // 得点の加算と表示
            document.getElementById("PointMsg").value=Point+"点";
            if (Point%240==0) {
                clearTimeout(TimeID);
                Bx=parseInt(Math.random()*30)*10+50;By=200;
                Dy=10;
                if (Level>=40) Level-=20
                init();
            }
        }
        if (By>420){
            BallN--;                    // ボールの残数を減らす
            document.getElementById("BallMsg").value="残り"+BallN+"個";
            clearTimeout(TimeID);  // ボールを停止する
            if (BallN>0){
                Bx=parseInt(Math.random()*30)*10+50;By=200;
                setTimeout("move()",1000);  // 1 秒後に再開
            }
        }
    }
```

11-2　マインスイーパー

おなじみのマインスイーパーを作ります。ここで作るゲームは以下のような内容です。

クリックした位置に機雷があれば、すべて機雷を表示し、機雷でないならその位置だけ開きます。マスを開けるところは練習問題 7-2、リバーシーゲームと同じ要領です。発展問題で、接近度数 0 のマスを開く再帰関数 visit を作ります。

minesweeper は（機雷除去の）掃海艇、または地雷除去装置の意味です。

配列 M[][]に機雷の情報、機雷の接近度数を格納します。(1,1)－(N,N)を実際のマスと対応させます。配列の外枠要素は visit 関数で再帰呼び出しを行う際に、配列の外に進まないようにするためのものです。このように配列の範囲を越えて探索を行なわないように見張りをしているものを番兵と呼びます。

配列 M[][]には機雷があれば-1、外枠要素は-2、それ以外の要素はその要素の隣接するところに機雷がある個数（接近度数）を格納します。例えば M[3][1]の要素は機雷が 2 個隣接しているので 2 となります。隣接する機雷がないところは 0 となります。

マスを開いたときに表示するイメージは接近度数 0,1,2・・に対し num0.png,num1.png,num2.png・・を使います。num1.png 以後は数字入りですが num0.png は数字は入っていません。

・機雷の接近度数をカウントアップする関数 count

機雷の接近度数を示す数値は、機雷を中心とする周辺 8 箇所を+1 することでカウントアップして求めます。このときその周辺 8 箇所の中で機雷のある要素と外枠の要素はカウントアップしません。

・mine1.html

```
<html>
<head>
<script type="text/javascript">
    var N,Bomb,W=20;        // マスの数、機雷の数、マスのサイズ
    var M;
    function yx(event)  // クリックイベント処理関数
    {
        var y,x,n,obj;
        y=Math.floor((event.clientY-50)/W+1);
        x=Math.floor((event.clientX-50)/W+1);
        if (M[y][x]==-2) return;  // visit で訪問してあれば戻る
        n=(y-1)*N+(x-1);
        obj=document.getElementById("img"+n);
        if (M[y][x]==-1){  // 機雷のときは全ての機雷を表示
            for (i=1;i<=N;i++){
```

```
                for (j=1;j<=N;j++){
                    if (M[i][j]==-1){
                        n=(i-1)*N+(j-1);
                        obj=document.getElementById("img"+n);
                        obj.src="bomb.png";
                    }
                }
            }
        }
        else
            obj.src="num"+M[y][x]+".png";            // 機雷でないとき
    }
    function count()   // 接近度数カウント関数
    {
        var i,j;
        for (i=1;i<=N;i++){
            for (j=1;j<=N;j++){
                if (M[i][j]==-1){
                    for (dx=-1;dx<=1;dx++){
                        for (dy=-1;dy<=1;dy++){
                            if   ((dx!=0   ||   dy!=0)   &&   (M[i+dy][j+dx]!=-1   &&
M[i+dy][j+dx]!=-2))
                                M[i+dy][j+dx]++;
                        }
                    }
                }
            }
        }
    }
    function init()
    {
        var i,j,x,y,n=0,tag="";
        var mine=document.getElementById("mine");
        N=parseInt(document.getElementById("level").value); // parseInt 必要
        Bomb=N;
        for (y=50;y<50+W*N;y+=W){
```

```
            for (x=50;x<50+W*N;x+=W){
                tag+="<img src='sea.png' id='img"+n+"' style='position:absolute;left:"+x+";top:"+y+"' onClick='yx(event)'>";
                n++;
            }
        }
        mine.innerHTML=tag;
        M=new Array(N+2);
        for (i=0;i<N+2;i++){
            M[i]=new Array(N+2);
            for (j=0;j<=N+1;j++){
                if (i==0 || i==N+1 || j==0 || j==N+1)
                    M[i][j]=-2;    // 外枠要素。番兵
                else
                    M[i][j]=0;
            }
        }
        for (i=0;i<Bomb;i++){   // y,x 位置のマスに機雷を最大で Bomb 個置く
            x=Math.floor(Math.random()*N+1);
            y=Math.floor(Math.random()*N+1);
            M[y][x]=-1;
        }
        count();
    }
</script>
</head>
<body>
<form style="position:absolute;left:50;top:10">
マスの数<input id="level" type="text" size="4">
<input type="button" value="表示" onClick="init()">
</form>
<div id="mine"></div>
</body>
</html>
```

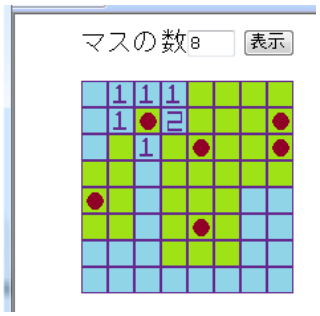
「発展課題」

接近度数 0 のマスをクリックしたらそれにつながる全ての接近度数 0 のマスを開く関数

visit を作りなさい。　「8-2 の 3. 閉路の探索」の vist 関数を元に作ります。上下左右に進む方向のマスの内容が 0 なら再帰的に訪問します。訪問した印として-2 を置きます。

・mine2.html

```
function visit(i,j)　// 接近度数 0 のマスを開ける関数
{
    var n,obj;
    n=(i-1)*N+(j-1);
    obj=document.getElementById("img"+n);
    obj.src="num"+M[i][j]+".png";
    M[i][j]=-2;　// 訪問した印
    if (M[i][j+1]==0) visit(i,j+1);          // 右位置への訪問
    if (M[i+1][j]==0) visit(i+1,j);          // 下位置への訪問
    if (M[i][j-1]==0) visit(i,j-1);          // 左位置への訪問
    if (M[i-1][j]==0) visit(i-1,j);          // 上位置への訪問
}
function yx(event)　// クリックイベント処理関数
{
     ・
     ・
    if (M[y][x]==-2) return;　// visit で訪問してあれば戻る
    n=(y-1)*N+(x-1);
    obj=document.getElementById("img"+n);
    if (M[y][x]==-1){　// 機雷のときは全ての機雷を表示
         ・
         ・
    }
    else if (M[y][x]==0)
        visit(y,x);                                  // 接近度数 0 のマスのときの処理
    else
        obj.src="num"+M[y][x]+".png";          // 接近度数 1 以上のマスのときの処理
}
```


マスの数8
表示

11-3　迷路

縦 NY、横 NX の各マスに迷路配列 M[][] を対応させ、その各要素に上壁の有無、右壁の有無、訪問の有無の３つの情報を持たせます。

（i,j）位置から進む方向を乱数で１：右、２；下、３左、４上の中から選択し、１マス進むことにします。その方向に進めるかは、そのマスがまだ未訪問である場合とします。これを進む方向がなくなるまで再帰的に繰り返します。全てのマスに上壁、右壁をつけて置き、進む方向の壁を取り去るという方法で迷路を作ります。通過に伴う壁の取り崩しは、次の要領で行います。

①右へ進む場合は、今いる位置の右壁をとる
②下へ進む場合は、進む位置の上壁をとる
③左へ進む場合は、進む位置の右壁をとる
④上へ進む場合は、今いる位置の上壁をとる

具体的には、たとえば M[i][j]位置の右壁をとるには次のようにします。

M[i][j]&=0xd;　　↓第 1 ビット

0x0d のビットパターンは「1101」で、これとの AND をとることにより、第 1 ビットだけを 0 にすることができます。

迷路配列の初期値は、外壁に番兵の 15（再トライ禁止、訪問済み、右壁あり、上壁あり）を置き、各マスには 3（未訪問、右壁あり、上壁あり）を置きます。

壁の長さを W、迷路の左上隅位置を(0,0)、迷路の入り口と出口の配列要素を（Si,Sj)と(Ei,Ej)とします。配列に i,j 位置に対応するマスの y,x 位置は以下の式で得られます。

y=(i-1)*W;

x=(j-1)*W;

この y,x 位置に「M[i][j] & 1」が真なら上壁イメージ（upwall.png）を、「M[i][j] & 2」が真なら右壁イメージ（rightwall.png）を配置します。

キー入力で迷路をたどります。経路を示す直線は id を h0～h99、v0～v99 まで付けた水平線イメージ（hline.png）と垂直線イメージ（vline.png）を使って描画します。

右に移動するときは oldY,oldX 位置に hline.png を配置してから oldX を+20 します。左に移動するときは oldX を-20 した後でその位置に hline.png を配置します。

☆キー入力イベント

キー入力イベント処理は onKeyPress アトリビュートに指定します。

矢印キーはキーイベントを取得できないので E（上)、S（左)、D（右)、X（下）キーを使います。入力されたキーのコードは event.keyCode で得られます。E は 115、S は 100、

D は 101、X は 120 という値になります。

E

S D

X

☆ビット演算子

プログラミング言語では、人間の感覚に合わせて 10 進数で表しますすが、実はコンピュータの世界では 2 進数が基本になっていてます。2 進数は 0 と 1 で表します。2 進数 4 桁の数値に対応する 10 進数値、16 進数値を表 11-1 に示します。2 進数 4 桁が 16 進数 1 桁に対応します。JavaScript では 16 進数値を示すのに先頭に 0x を付け、0xd のように表します。

表 11-1　2 進数、10 進数、16 進数の関係

2 進数	10 進数	16 進数
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	a
1011	11	b
1100	12	c
1101	13	d
1110	14	e
1111	15	f

ビット演算子は、数値を 2 進表現したときの「1」または「0」の各ビットごとの論理演算を行います。論理演算には AND、OR があります。ビット a と b を AND,OR した結果(真理値表)

を表 11-2 に示します。JavaScript では AND 演算を&演算子 OR 演算を|演算子で行います。

表 11-2　AND と OR の真理値表

AND(論理積)

a	b	a&b
0	0	0
0	1	0
1	0	0
1	1	1

OR(論理和)

a	b	A|b
0	0	0
0	1	1
1	0	1
1	1	1

たとえばある数と「0x0d」との AND をとることの効果を考えてみましょう。「0x0d」は 2 進数で表すと「1101」となります。ある数がたとえば「1011」と「1110」の場合について「0x0d」と AND をとると以下のようになります。つまり、「1101」と AND をとることは、「1101」の 0 の位置に対応するビットを強制的に「0」にすることになります。このような操作をマスクと呼びます。

```
   1011                  1110
&  1101               &  1101
   1001                  1100
     ↑強制的に 0 にされる。    ↑強制的に 0 にされる。
```

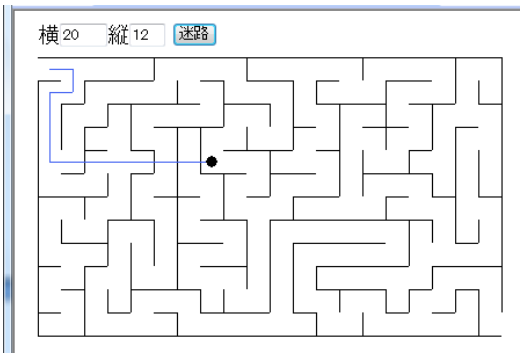
・maze1.html

```
<html>
<head>
<script type="text/javascript">
    var NY,NX,Si,Sj,Ei,Ej;  // マスの縦と横、入口位置、出口位置
    var W=20;               // 壁の長さ
    var M;          // 迷路配列
    var oldY,oldX,H,V; // 移動の前の位置、水平線と垂直線の id 番号
    function move(event)
    {
        var obj,mouse;
        if (event.keyCode==115){
            oldX-=20;//左
            obj=document.getElementById("h"+H);H++;
            obj.style.top=oldY;obj.style.left=oldX;
```

```
    }
    else if (event.keyCode==100){ //右
        obj=document.getElementById("h"+H);H++;
        obj.style.top=oldY;obj.style.left=oldX;
        oldX+=20;
    }
    else if (event.keyCode==101){
        oldY-=20;//上
        obj=document.getElementById("v"+V);V++;
        obj.style.top=oldY;obj.style.left=oldX;
    }
    else if (event.keyCode==120){//下
        obj=document.getElementById("v"+V);V++;
        obj.style.top=oldY;obj.style.left=oldX;
        oldY+=20;
    }
    mouse=document.getElementById("mouse");
    mouse.style.top=oldY-5;mouse.style.left=oldX-5;
}
function genmaze(i,j)   // 迷路の作成
{
    var n;
    M[i][j]|=4;
    while (M[i][j+1]==3 || M[i+1][j]==3 || M[i][j-1]==3 || M[i-1][j]==3){
        n=Math.floor(Math.random()*4)+1;
        if (n==1 && M[i][j+1]==3){
            M[i][j]&=0xd;
            genmaze(i,j+1);
        }
        else if(n==2 && M[i-1][j]==3){
            M[i][j]&=0xe;
            genmaze(i-1,j);
        }
        else if(n==3 && M[i][j-1]==3){
            M[i][j-1]&=0xd;
            genmaze(i,j-1);
```

```
            }
            else if(n==4 && M[i+1][j]==3){
                M[i+1][j]&=0xe;
                genmaze(i+1,j);
            }
        }
    }
    function meiro()
    {
        var i,j,tag="";
        var maze=document.getElementById("maze");
        NY=parseInt(document.getElementById("NY").value);
        NX=parseInt(document.getElementById("NX").value);
        Si=1;Sj=1;Ei=NY;Ej=NX;
        oldY=10;oldX=10;H=0;V=0;
        M=new Array(NY+2);
        for (i=0;i<NY+2;i++){
            M[i]=new Array(NX+2);
        }
        for (i=0;i<NY+2;i++){
            for (j=0;j<NX+2;j++){
                if (i==0 || j==0 || i==NY+1 || j==NX+1)
                    M[i][j]=15;
                else
                    M[i][j]=3;
            }
        }
        genmaze(Ei,Ej);
        M[Ei][Ej]&=0xd;
        for (i=1;i<=NY;i++){
            for (j=1;j<=NX;j++){
                x=(j-1)*W;
                y=(i-1)*W;
                if (M[i][j] & 1){
                    tag+="<img                                    src='upwall.png'
style='position:absolute;left:"+x+";top:"+y+"'>";
```

```
                    }
                    if (M[i][j] & 2){
                        tag+="<img                                          src='rightwall.png'
style='position:absolute;left:"+(x+W)+";top:"+y+"'>";
                    }
                }
            }
            tag+="<img                                                      src='mazeh.png'
style='position:absolute;left:0;top:"+(NY*20)+";height:1;width:"+(NX*20)+"'>"; //  下壁
            tag+="<img                                                      src='mazev.png'
style='position:absolute;left:0;top:20;width:1;height:"+(NY*20-20)+"'>";          //  左壁
            for (i=0;i<100;i++){
                tag+="<img                  "+"id='h"+i+"'"+"               src='hline.png'
style='position:absolute;left:-40;top:0'>";
                tag+="<img                  "+"id='v"+i+"'"+"               src='vline.png'
style='position:absolute;left:-40;top:0'>";
            }
            tag+="<img id='mouse' src='mouse.png' style='position:absolute;left:5;top:5'>";
            maze.innerHTML=tag;
        }
</script>
</head>
<body onKeyPress="move(event)">
<form style="position:absolute;left:20;top:10">
横<input id="NX" type="text" size="5">縦<input id="NY" type="text" size="3">
<input type="button" value="迷路" onClick="meiro()">
</form>
<div id="maze" style="position:absolute;left:20;top:40"></div>
</body>
</html>
```



「発展課題」

先の例題では、壁を突き抜けて進んでしまうので次の機能を追加しなさい。

・進めない位置を判定

たとえば左に進場合はその位置(i,j)の左のマスの右壁の有無を調べることになるので、(M[i][j-1]&2)==0 なら右壁が無いことになります。

・逆戻りしたら消す

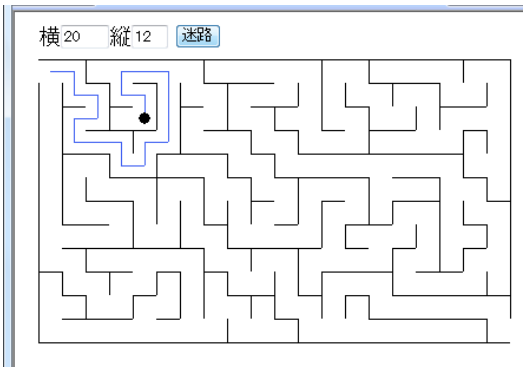
M[][]と同じ 2 次元配列 G[][]を用意し、そこに直線を描いたかどうかの情報（描いた線の id 名）を与えます。描いていないところは 0 を置きます。たとえば左に進場合はその位置(i,j)の左のマスの、G[i][j-1]が 0 なら水平線を描き、0 でないなら、その id 名の水平線を隠します。

・maze2.html

```
var M,G;            // 迷路配列
var oldY,oldX,H,V; // 移動の前の位置、水平線と垂直線の id 番号
function move(event)
{
    var obj,mouse,i,j;
    i=parseInt((oldY+10)/20);j=parseInt((oldX+10)/20);
    if (event.keyCode==115 && (M[i][j-1]&2)==0){
        oldX-=20;//左
        if (G[i][j-1]==0){ //線を引く時は移動先の配列
            G[i][j-1]="h"+H;
            obj=document.getElementById("h"+H);H++;
            obj.style.top=oldY;obj.style.left=oldX;
        }
        else {
            obj=document.getElementById(G[i][j]);
            obj.style.visibility="hidden";
            G[i][j]=0;   // 消すときは今いる位置の配列
```

```
        }
    }
    if (event.keyCode==100 && (M[i][j]&2)==0){//右
        if (G[i][j+1]==0){
            G[i][j+1]="h"+H;
            obj=document.getElementById("h"+H);H++;
            obj.style.top=oldY;obj.style.left=oldX;
        }
        else {
            obj=document.getElementById(G[i][j]);
            obj.style.visibility="hidden";
            G[i][j]=0;
        }
        oldX+=20;
    }
    if (event.keyCode==101 && (M[i][j]&1)==0){//上
        oldY-=20;
        if (G[i-1][j]==0){
            G[i-1][j]="v"+V;
            obj=document.getElementById("v"+V);V++;
            obj.style.top=oldY;obj.style.left=oldX;
        }
        else {
            obj=document.getElementById(G[i][j]);
            obj.style.visibility="hidden";
            G[i][j]=0;
        }
    }
    if (event.keyCode==120 && (M[i+1][j]&1)==0){//下
        if (G[i+1][j]==0){
            G[i+1][j]="v"+V;
            obj=document.getElementById("v"+V);V++;
            obj.style.top=oldY;obj.style.left=oldX;
        }
        else {
            obj=document.getElementById(G[i][j]);
```

```
                obj.style.visibility="hidden";
                G[i][j]=0;
            }
            oldY+=20;
        }
        mouse=document.getElementById("mouse");
        mouse.style.top=oldY-5;mouse.style.left=oldX-5;
}
function meiro()
{
        .
        .
        M=new Array(NY+2);
        G=new Array(NY+2);
        for (i=0;i<NY+2;i++){
            M[i]=new Array(NX+2);
            G[i]=new Array(NX+2);
        }
        for (i=0;i<NY+2;i++){
            for (j=0;j<NX+2;j++){
                if (i==0 || j==0 || i==NY+1 || j==NX+1)
                    M[i][j]=15;
                else
                    M[i][j]=3;
                G[i][j]=0;
            }
        }
        G[1][1]=1;     //始点用特殊処理
        .
        .
}
```


横 20
縦 12
迷路