

# ななちゃんのIT教室

## もぐらたたきゲームを作ろうの巻

by nara.yasuhiro@gmail.com

プログラミングをまったく知らなかったななちゃんが  
もぐらたたきゲームプログラムを作れるようになるまでのお話

第 0.3 版 2017 年 5 月 7 日



フリー素材  
<http://freeillustration.net>

もくじ  
第1回 イベント？

## 第1回 イベント？

なな：これって、朝日新聞の「ののちゃんのDO科学」のパクリ？

先生：パロディって言ってちょうだい。家政婦のミタ（「家政婦は見た」のパロディ）、クレヨンしんちゃんのダズニーランド（「ディズニーランド」のパロディ）みたいなものよ。

なな：もぐらたたきゲームを作りたいんだけど、マウスをクリックしたら JavaScript プログラムが動くようにするにはどうすれば良いの。



先生：クリック対象の html 部品に、「onclick=go()」みたいな項をつけ加えるの。こういうのを、コンピュータ技術者は「マウスがクリックされたという『イベント』が発生したら、『go()』という『イベントハンドラ』を実行する」と言うの。

なな：「イベント会場」なんかの「イベント」？

先生：そうそう。event は、英語で、「出来事、事件、行事」の意味なの。「イベント会場」の event は「行事」、JavaScript の event は「出来事」ね。

なな：英語国民だったら勉強しなくても分かるから得みたいね。

先生：英語国民だと、日常のことばとごっちゃになるという悩みもあるらしいよ。「マウスの左ボタンをクリック」というのが「ネズミの左へソをカチャ！する」みたいに聞こえるのかも。



なな：そーなんだあ。

先生：それでは、まず、練習ということで、「お人好しのもぐら」ゲームのプログラムを作りましょう。

なな：「お人好しのもぐら」？

先生：画面に、ふたつ、「穴」の絵があるけど、マウスカーソルをその上に移動すると、もぐらさんが顔を出し、マウスカーソルを穴の絵の外に移動すると、穴の絵に戻るの。そして、もぐらさんが顔を出しているところで、マウスの左ボタンをクリックすると、「あたり！」の絵になるの。

なな：簡単に「あたり！」になるから「お人好し」なのね。



先生：それでは、「お人好しのもぐら」のプログラムを見てみましょう。

mole1.html

```

<h1>もぐらたたき(お人好しのもぐら版)</h1>

<br>
得点<input type=text id=out value=0><br>
<script>
  var outp = document.getElementById('out');

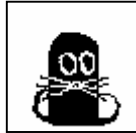
  function up(p) {
    p.src = "moleface.png";
  }

  function dn(p) {
    p.src = "molehole.png";
  }

  function hit(p) {
    p.src = "molehit.png";
    outp.value++;
  }
</script>

```

注意: 「onMouseOver」などは、「onmouseover」と書いても大丈夫です



moleface.png



molehole.png



molehit.png

第一段階のプログラム

「」は、穴の絵を表示し、マウスカーソルが乗ったら up、はずれたら dn、クリックされたら hit というイベントハンドラを呼び出す、という意味です。(this) は、up、dn、hit に、入れ替え用の絵（顔、穴、当たり!）を表示する位置を教えるためのものなのよ。

なな: たったこれだけのプログラムで、もぐらたたきの雰囲気を感じられるわね。

先生: つぎは、もぐら 1 匹の、完全版もぐらたたきのプログラムです。

注意: 次のプログラムでは、第一段階のプログラムで省略していた

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>

```

と、

```

  </body>
</html>

```

をつけています。

先生: 「onLoad=changed(this)」は、画像表示が完了したというイベント発生で changed ハンドラを呼び出す。  
 changed ハンドラは、1 ~ 5 秒 (ランダム変動) 後 (タイムアウトイベント) に、updn ハンドラを呼び出し。  
 updn ハンドラは、穴の絵 なら 顔の絵 に、穴の絵でなければ穴の絵に、絵の差し替えを行う。  
 つまり、「絵を取り換え」→ <「絵の表示完了」→ 「1 ~ 5 秒後に絵を取り換え」> を繰り返す。  
 それから、それとは別に、1 秒ごとに残り時間表示を更新し、ゼロになったら停止。30 秒からスタート。

第二段階の  
プログラム

C:\Users\User\Desktop\ X +

← → ↻ | file:///C:/Users/User/Desktop/SFC副教材2/もぐらた

得点  残り時間

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    得点<input type="text" id="out" value="0">
    残り時間<input type="text" id="tleft" value="30"><br>
    <table border="1"><tr><td id="area">
      
    </td></tr></table>
    <script>
var outp = document.getElementById("out");
var tleftp = document.getElementById("tleft");
var tleft = 30;

function updn(p) {
  if ((tleft <= 0)&&(p.src.match(/molehole.png/)));
  else if (p.src.match(/molehole.png/)) p.src = "moleface.png";
  else
    p.src = "molehole.png";
}

function hit(p) {
  if (p.src.match(/moleface.png/)) {
    p.src = "molehit.png";
    outp.value++; }
}

function changed(p) {
  setTimeout(updn,Math.floor(Math.random()*4000+1000),p);
}

function tdown() {
  tleftp.value = --tleft;
  if (tleft <= 0) clearInterval(tid);
}

outp.value = 30;
var tid = setInterval (tdown,1000);
    </script>
  </body>
</html>
    
```

**外枠表示のための table**

画像の表示完了というイベント発生時に changed ハンドラを呼び出す

残り時間がなくなったら 画像更新はしない

穴の絵 なら 顔の絵 に、穴の絵でなければ 穴の絵 に 差し替え

1 ~ 5 秒後に updn(p) を呼び出す

タイマーイベントのハンドラ

残り時間表示を 1 減らす。それを表示。

残り時間がゼロになったら タイマーを停止

1 秒(1000 ミリ秒)ごとに tdown を呼び出す

なな: 「画像の表示完了」 や、「タイマーで設定した時間が経過した」というのも、イベントの一種ということね。

先生: そうね。そして、さいごに、第三段階、最終版のプログラム。  
ちょっと高等テクニックだけど、<img> タグを 5 × 5 個  
(25 個) 複写しています。

第三段階  
(最終) の  
プログラム

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <h1>もぐらたたき</h1>
    得点<input type="text" id="out" value="0">
    残り時間<input type="text" id="tleft" value="30"><br>
    <table border="1"><tr><td id="area">
      
    </td></tr></table>

  <script>
    var outp    = document.getElementById("out");
    var tleftp = document.getElementById("tleft");
    var tleft  = 30;

    function updn(p) {
      if ((tleft <= 0)&&(p.src.match(/molehole.png/)));
      else if (p.src.match(/molehole.png/)) p.src = "moleface.png";
      else p.src = "molehole.png";
    }

    function hit(p) {
      if (p.src.match(/moleface.png/)) {
        p.src = "molehit.png";
        outp.value++;
      }
    }

    function changed(p) {
      setTimeout(updn,Math.floor(Math.random()*4000+1000),p);
    }

    function tdown() {
      tleftp.value = --tleft;
      if (tleft <= 0) clearInterval(tid);
    }

    var areap = document.getElementById("area");
    var cvs = areap.children[0]; areap.innerHTML="";
    for (var j=0; j<5; j++) {
      for(k=0; k<5; k++) { areap.appendChild(cvs.cloneNode(true));}
      areap.appendChild(document.createElement("br"));
    }

    outp.value = 30;
    var tid = setInterval(tdown,1000);
  </script>
</body>
</html>

```



もぐら  
ボックスの  
ひな型  
(<img>タグ)



もぐらボックス  
(<img>タグ)  
1 つを  
コピー後クリア、  
あらためて  
5 × 5 個を  
貼り付けている

cloneNode は  
タグ の複製  
(クローン)を作る

appendChild は  
タグ を貼り付ける

<br> タグを作る