

全部课程 (/courses/) / 网页版2048 (/courses/62) / 网页版2048

在线实验，请到PC端体验

## 网页版 2048

### 一、实验介绍

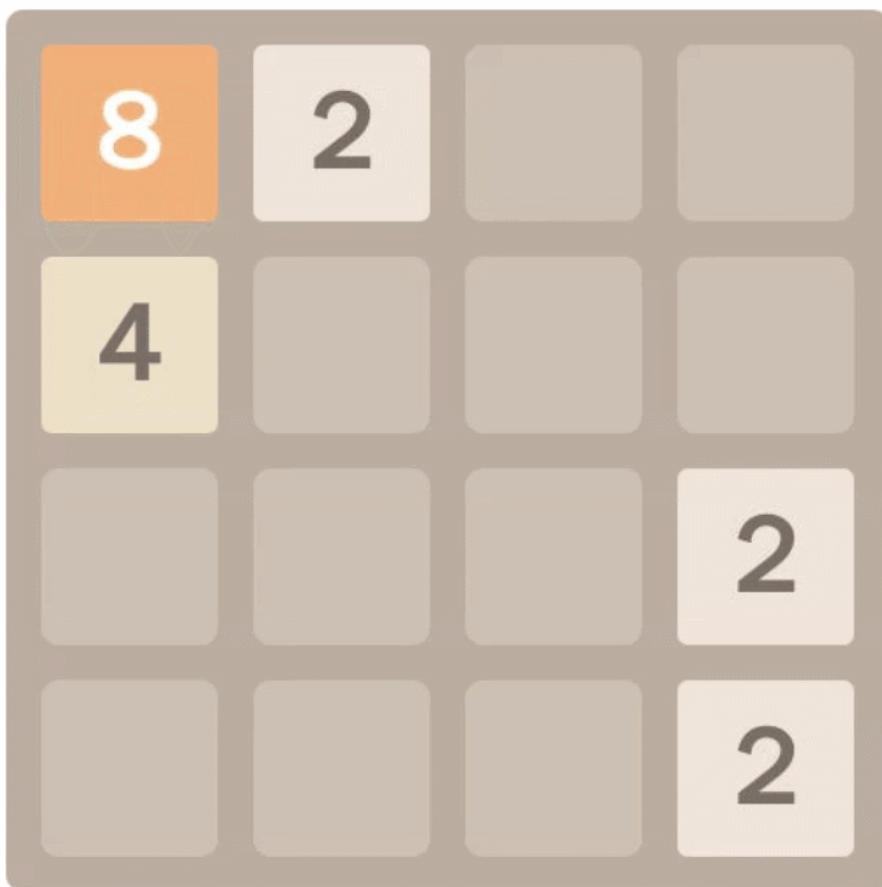
#### 1.1 实验内容

《2048》是一款风靡一时的，非常简单上手的小游戏，如果你还没玩过，可以在手机上下载一个体验一下。本教程带领大家使用 HTML+CSS+JS+jQuery 实现一个网页版的《2048》。最终的效果图如下：

# ShiYanLou-2048

New Game

score: 8



#### 1.2 实验知识点

- 学习 web 应用程序的开发流程
- 如何让应用在移动端自适应处理以应对各种大小的屏幕，布局和初始化

- 使用 JavaScript 及 jQuery 编写游戏的逻辑、实现方块移动、判定游戏结果等。

## 1.3 实验环境

1. Terminal: Linux 命令行终端, 打开后会进入 Bash 环境, 可以使用 Linux 命令
2. Firefox: 浏览器, 可以在需要前端界面的课程里, 只需要打开环境里写的 HTML/JS 页面即可
3. GVim: 非常好用的编辑器, 最简单的用法可以参考课程 Vim编辑器 (<http://www.shiyanlou.com/courses/2>) (或者使用 Sublime Text 编辑器)

实验楼的 Sublime Text 编辑器位于: 应用程序菜单->开发 下。

## 1.4 适合人群

本课程难度中等, 属于中级级别课程, 适合具有前端基础 (HTML+CSS+JavaScript+jQuery) 的用户, 可以加深巩固你的前端技能。

## 1.5 代码获取

你可以通过下面命令将代码下载到实验楼环境中, 作为参照对比进行学习。

```
$ wget https://github.com/shiyanlou/2048_code
```

## 二、实验原理

抛开页面样式, 2048 游戏可以抽象为一个二维数组, 初始状态下, 需要随机生成两个数字, 随机生成的数字只能是2或4。接下来的实现中, 我们会让2和4出现概率相同, 当然, 你要可以让4出现的概率小一些。

```
0 0 0 0
0 0 0 0
2 0 0 0
0 0 2 0
```

接下来, 当我按下一个方向键, 数字会向这个方向移动, 并且会合并相邻的两个相同数字的单元。然后再随机生成一个数字, 比如, 这里我们按上->左->上, 界面可能变为:

```
上 2 0 2 0 左 4 0 0 0 上 8 0 2 0
0 0 0 0    0 0 0 0    0 0 0 0
0 0 0 0    0 0 2 0    0 0 0 0
4 0 0 0    4 0 0 0    0 4 0 0
```

这样, 我们不断的移动、合并数字, 直到数字不能再合并或者有2048出现, 游戏结束。你可以看出, 游戏的核心就是操作这个二维数组。

## 三、项目文件结构

```
2048
|__ index.html
|__ main.js
|__ support.js
|__ showanimation.js
|__ style.css
```

## 四、实验步骤

### 4.1 页面布局

相信你已经观察到, 2048 游戏是建立在 16 个方格上的, 下面我们使用 div + css 绘制出这 4X4 的单元格:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, height=device-height, initial-scale=1.0, minimum-scale=
1.0, maximum-scale=1.0, user-scalable=no"/>
  <title>ShiYanLou-2048</title>
  <link rel="stylesheet" href="style.css"/>
  <script type="text/javascript" src="http://labfile.oss.aliyuncs.com/jquery/2.1.3/jquery.min.js"></script>
  <script type="text/javascript" src="main.js"></script>
  <script type="text/javascript" src="showanimation.js"></script>
  <script type="text/javascript" src="support.js"></script>
</head>
<body>
  <header>
    <h1>ShiYanLou-2048</h1>
    <a href="javascript:new_game();" id="new_game_button">New Game</a>
    <p>score: <span id="score">0</span></p>
  </header>
  <div id="grid_container">
    <div class="grid_cell" id="grid_cell_0_0"></div>
    <div class="grid_cell" id="grid_cell_0_1"></div>
    <div class="grid_cell" id="grid_cell_0_2"></div>
    <div class="grid_cell" id="grid_cell_0_3"></div>

    <div class="grid_cell" id="grid_cell_1_0"></div>
    <div class="grid_cell" id="grid_cell_1_1"></div>
    <div class="grid_cell" id="grid_cell_1_2"></div>
    <div class="grid_cell" id="grid_cell_1_3"></div>

    <div class="grid_cell" id="grid_cell_2_0"></div>
    <div class="grid_cell" id="grid_cell_2_1"></div>
    <div class="grid_cell" id="grid_cell_2_2"></div>
    <div class="grid_cell" id="grid_cell_2_3"></div>

    <div class="grid_cell" id="grid_cell_3_0"></div>
    <div class="grid_cell" id="grid_cell_3_1"></div>
    <div class="grid_cell" id="grid_cell_3_2"></div>
    <div class="grid_cell" id="grid_cell_3_3"></div>
  </div>
</body>
</html>
```

然后需要给页面和每个格子加上样式:

```
header {
  display: block;
  margin: 0 auto;
  width: 100%;
  text-align: center;
}

header h1 {
  font-family: Arial;
  font-size: 60px;
  font-weight: bold;
  margin: 0 auto;
}

header #new_game_button {
  display: block;
  margin: 0px auto;
  width: 100px;
  padding: 10px 10px;
  background-color: #8f7a66;
  font-family: Arial;
  color: white;
  border-radius: 10px;
  text-decoration: none;
}

header #new_game_button:hover {
  background-color: #9f8b7a;
}

header p {
  font-family: Arial;
  font-size: 25px;
  margin: 5px auto;
}

#grid_container {
  width: 460px;
  height: 460px;
  padding: 20px;
  margin: 0px auto;
  background-color: #bbada0;
  border-radius: 10px;
  position: relative;
}

.grid_cell {
  width: 100px;
  height: 100px;
  border-radius: 6px;
  background-color: #ccc0b3;
  position: absolute;
}

.number_cell {
  border-radius: 6px;
  font-family: Arial;
  font-weight: bold;
  font-size: 60px;
  line-height: 100px;
  text-align: center;
  position: absolute;
}
```

动手实践是学习 IT 技术最有效的方式!

开始实验

完成这步后，使用 Firefox 浏览器打开 index.html 文件，我们应该能看到下面的效果了：

# ShiYanLou-2048

New Game

score: 0



动手实践是学习 IT 技术最有效的方式!

开始实验

## 4.2 初始化棋局

现在移动端设备越来越流行，所以我们需要做自适应处理以应对各种大小的屏幕，游戏开始的时候，我们需要在棋盘上随机生成两个数字，这里通过 JavaScript 代码来实现，在 `main.js` 中

```

var board = new Array(); //每个格子的数字
var score = 0; //分数
var has_conflicted = new Array(); //解决连续消除的标记
var startx = 0; //移动端触摸屏幕时开始点的x坐标
var starty = 0; //移动端触摸屏幕时开始点的y坐标
var endx = 0; //移动端触摸屏幕时结束点的x坐标
var endy = 0; //移动端触摸屏幕时结束点的y坐标
var success_string = 'Success';
var gameover_string = 'GameOver';

//HTML文档加载完成后，初始化棋局
$(document).ready(function() {
    //做自适应处理
    prepare_for_mobile();
    new_game();
});

//开始新游戏
function new_game() {
    //初始化棋盘
    init();
    //在随机两个格子生成数字
    generate_one_number();
    generate_one_number();
}

//初始化
function init() {
    for (var i = 0; i < 4; i++) {
        for (var j = 0; j < 4; j++) {
            var grid_cell = $('#grid_cell_' + i + '_' + j);
            grid_cell.css('top', get_pos_top(i, j));
            grid_cell.css('left', get_pos_left(i, j));
        }
    }
    for (var i = 0; i < 4; i++) {
        board[i] = new Array();
        has_conflicted[i] = new Array();
        for (var j = 0; j < 4; j++) {
            board[i][j] = 0;
            has_conflicted[i][j] = false;
        }
    }
    update_board_view();
    score = 0;
    update_score(score);
}

//更新棋局
function update_board_view() {
    $('.number_cell').remove();
    for (var i = 0; i < 4; i++) {
        for (var j = 0; j < 4; j++) {
            $('#grid_container').append('<div class="number_cell" id="number_cell_' + i + '_' + j + '"></div>');
            var number_cell = $('#number_cell_' + i + '_' + j);
            if (board[i][j] == 0) {
                number_cell.css('width', '0px');
                number_cell.css('height', '0px');
                number_cell.css('top', get_pos_top(i, j) + cell_side_length / 2);
                number_cell.css('left', get_pos_left(i, j) + cell_side_length / 2);
            } else {
                number_cell.css('width', cell_side_length);
                number_cell.css('height', cell_side_length);
                number_cell.css('top', get_pos_top(i, j));
                number_cell.css('left', get_pos_left(i, j));
                number_cell.css('background-color', get_number_background_color(board[i][j]));
                number_cell.css('color', get_number_color(board[i][j]));
                number_cell.text(board[i][j]);
            }
            has_conflicted[i][j] = false;
        }
    }
    $('.number_cell').css('line-height', cell_side_length + 'px');
    $('.number_cell').css('font-size', 0.6 * cell_side_length + 'px');
}

//随机在一个格子生成数字

```

动手实践是学习 IT 技术最有效的方式!

开始实验

```
function generate_one_number() {
  if (nospace(board)) {
    return false;
  }
  //随机一个位置
  var randx = parseInt(Math.floor(Math.random() * 4));
  var randy = parseInt(Math.floor(Math.random() * 4));
  var time = 0;
  while (time < 50) {
    if (board[randx][randy] == 0) {
      break;
    }
    randx = parseInt(Math.floor(Math.random() * 4));
    randy = parseInt(Math.floor(Math.random() * 4));
    time++;
  }
  if (time == 50) {
    for (var i = 0; i < 4; i++) {
      for (var j = 0; j < 4; j++) {
        if (board[i][j] == 0) {
          randx = i;
          randy = j;
        }
      }
    }
  }
  //随机一个数字
  var rand_number = Math.random() < 0.5 ? 2 : 4;
  //在随机位置显示随机数字 动手实践是学习 IT 技术最有效的方式! 开始实验
  board[randx][randy] = rand_number;
  show_number_with_animation(randx, randy, rand_number);
  return true;
}

//自适应处理
function prepare_for_mobile() {
  if (document_width > 500) {
    grid_container_width = 500;
    cell_side_length = 100;
    cell_space = 20;
  }
  $('#grid_container').css('width', grid_container_width - 2 * cell_space);
  $('#grid_container').css('height', grid_container_width - 2 * cell_space);
  $('#grid_container').css('padding', cell_space);
  $('#grid_container').css('border-radius', 0.02 * grid_container_width);
  $('.grid_cell').css('width', cell_side_length);
  $('.grid_cell').css('height', cell_side_length);
  $('.grid_cell').css('border-radius', 0.02 * grid_container_width);
}
```

在 support.js 中:

```

document_width = window.screen.availWidth; //屏幕宽度
grid_container_width = 0.92 * document_width; //棋盘宽度
cell_side_length = 0.18 * document_width; //格子的大小
cell_space = 0.04 * document_width; //格子之间的间隔

//获得相应格子距离棋盘顶部的距离
function get_pos_top(i, j) {
    return cell_space + i * (cell_space + cell_side_length);
}

//获得相应格子距离棋盘左边的距离
function get_pos_left(i, j) {
    return cell_space + j * (cell_space + cell_side_length);
}

//获得相应数字的背景色
function get_number_background_color(number) {
    switch (number) {
        case 2: return '#eee4da'; break;
        case 4: return '#ede0c8'; break;
        case 8: return '#f2b179'; break;
        case 16: return '#f59563'; break;
        case 32: return '#f67c5f'; break;
        case 64: return '#f65e3b'; break;
        case 128: return '#edcf72'; break;
        case 256: return '#edcc61'; break;
        case 512: return '#9c0'; break;
        case 1024: return '#359e95'; break;
        case 2048: return '#09c'; break;
        case 4096: return '#a6c'; break;
        case 8192: return '#93c'; break;
    }
    return 'black';
}

//获得相应数字的颜色
function get_number_color(number) {
    if (number <= 4)
        return '#776e65';
    return 'white';
}

//判断棋盘上是否还有空格子
function nospace(board) {
    for (var i = 0; i < 4; i++) {
        for (var j = 0; j < 4; j++) {
            if (board[i][j] == 0) {
                return false;
            }
        }
    }
    return true;
}

```

动手实践是学习IT技术最有效的方式!

开始实验

在 showanimation.js 中:

```

//动画显示数字格子
function show_number_with_animation(i, j, rand_number) {
    var number_cell = $('#number_cell_' + i + '_' + j);
    number_cell.css('background-color', get_number_background_color(rand_number));
    number_cell.css('color', get_number_color(rand_number));
    number_cell.text(rand_number);
    number_cell.animate({
        width: cell_side_length,
        height: cell_side_length,
        top: get_pos_top(i, j),
        left: get_pos_left(i, j)
    }, 50);
}

//更新分数
function update_score(score) {
    $('#score').text(score);
}

```



完成这步后, 能看到下面的效果:

# ShiYanLou-2048

New Game

score: 0



开始实验

## 4.3 移动数字块

上面我们完成了布局和初始化, 最后我们就要实现让数字块能移动, 并能够消除, 直至游戏成功或失败, 在 `main.js` 中加入

```
//监听键盘的上下左右移动
$(document).keydown(function(event) {
    if ($('#score').text() == success_string) {
        new_game();
        return;
    }
    switch (event.keyCode) {
        case 37: //left
            event.preventDefault();
            if (move_left()) {
                setTimeout('generate_one_number()', 210);
                setTimeout('is_gameover()', 300);
            }
            break;
        case 38: //up
            event.preventDefault();
            if (move_up()) {
                setTimeout('generate_one_number()', 210);
                setTimeout('is_gameover()', 300);
            }
            break;
        case 39: //right
            event.preventDefault();
            if (move_right()) {
                setTimeout('generate_one_number()', 210);
                setTimeout('is_gameover()', 300);
            }
            break;
        case 40: //down
            event.preventDefault();
            if (move_down()) {
                setTimeout('generate_one_number()', 210);
                setTimeout('is_gameover()', 300);
            }
            break;
        default:
            break;
    }
});

//监听移动设备的触摸开始
document.addEventListener('touchstart', function(event) {
    startx = event.touches[0].pageX;
    starty = event.touches[0].pageY;
});

//监听移动设备的触摸移动
document.addEventListener('touchmove', function(event) {
    event.preventDefault();
});

//监听移动设备的触摸结束
document.addEventListener('touchend', function(event) {
    endx = event.changedTouches[0].pageX;
    endy = event.changedTouches[0].pageY;

    var deltax = endx - startx;
    var deltax = endy - starty;
    if (Math.abs(deltax) < 0.3 * document_width && Math.abs(deltax) < 0.3 * document_width) {
        return;
    }
    if ($('#score').text() == success_string) {
        new_game();
        return;
    }
    //x
    if (Math.abs(deltax) >= Math.abs(deltax)) {
        if (deltax > 0) {
            //move right
            if (move_right()) {
                setTimeout('generate_one_number()', 210);
                setTimeout('is_gameover()', 300);
            }
        } else {
            //move left
            if (move_left()) {
                setTimeout('generate_one_number()', 210);
            }
        }
    }
});
```

动手实践是学习 IT 技术最有效的方式!

开始实验

```

        setTimeout('is_gameover()', 300);
    }
} else { //y
    if (deltay > 0) {
        //move down
        if (move_down()) {
            setTimeout('generate_one_number()', 210);
            setTimeout('is_gameover()', 300);
        }
    } else {
        //move up
        if (move_up()) {
            setTimeout('generate_one_number()', 210);
            setTimeout('is_gameover()', 300);
        }
    }
}
});

//向左移动
function move_left() {
    if (!can_move_left(board)) {
        return false;
    }
    //move left
    for (var i = 0; i < 4; i++) {
        for (var j = 1; j < 4; j++) {
            if (board[i][j] != 0) {
                for (var k = 0; k < j; k++) {
                    if (board[i][k] == 0 && no_block_horizontal(i, k, j, board)) {
                        show_move_animation(i, j, i, k);
                        board[i][k] = board[i][j];
                        board[i][j] = 0;
                        break;
                    } else if (board[i][k] == board[i][j] && no_block_horizontal(i, k, j, board) && !has_conflicted[i][k]) {
                        show_move_animation(i, j, i, k);
                        board[i][k] += board[i][j];
                        board[i][j] = 0;
                        //add score
                        score += board[i][k];
                        update_score(score);
                        has_conflicted[i][k] = true;
                        break;
                    }
                }
            }
        }
    }
    setTimeout('update_board_view()', 200);
    return true;
}

//向右移动
function move_right() {
    if (!can_move_right(board)) {
        return false;
    }
    //move right
    for (var i = 0; i < 4; i++) {
        for (var j = 2; j >= 0; j--) {
            if (board[i][j] != 0) {
                for (var k = 3; k > j; k--) {
                    if (board[i][k] == 0 && no_block_horizontal(i, j, k, board)) {
                        show_move_animation(i, j, i, k);
                        board[i][k] = board[i][j];
                        board[i][j] = 0;
                        break;
                    } else if (board[i][k] == board[i][j] && no_block_horizontal(i, j, k, board) && !has_conflicted[i][k]) {
                        show_move_animation(i, j, i, k);
                        board[i][k] += board[i][j];
                        board[i][j] = 0;
                        //add score
                        score += board[i][k];
                        update_score(score);
                        has_conflicted[i][k] = true;
                        break;
                    }
                }
            }
        }
    }
}

```

动手实践是学习 IT 技术最有效的方式!

开始实验

```

    }
  }
}
setTimeout('update_board_view()', 200);
return true;
}

//向上移动
function move_up() {
  if (!can_move_up(board)) {
    return false;
  }
  //move up
  for (var j = 0; j < 4; j++) {
    for (var i = 1; i < 4; i++) {
      if (board[i][j] != 0) {
        for (var k = 0; k < i; k++) {
          if (board[k][j] == 0 && no_block_vertical(j, k, i, board)) {
            show_move_animation(i, j, k, j);
            board[k][j] = board[i][j];
            board[i][j] = 0;
            break;
          } else if (board[k][j] == board[i][j] && no_block_vertical(j, k, i, board) && !has_conflicted[k][j]) {
            show_move_animation(i, j, k, j);
            board[k][j] += board[i][j];
            board[i][j] = 0;
            //add score
            score += board[k][j];
            update_score(score);
            has_conflicted[k][j] = true;
            break;
          }
        }
      }
    }
  }
}
setTimeout('update_board_view()', 200);
return true;
}

//向下移动
function move_down() {
  if (!can_move_down(board)) {
    return false;
  }
  //move down
  for (var j = 0; j < 4; j++) {
    for (var i = 2; i >= 0; i--) {
      if (board[i][j] != 0) {
        for (var k = 3; k > i; k--) {
          if (board[k][j] == 0 && no_block_vertical(j, i, k, board)) {
            show_move_animation(i, j, k, j);
            board[k][j] = board[i][j];
            board[i][j] = 0;
            break;
          } else if (board[k][j] == board[i][j] && no_block_vertical(j, i, k, board) && !has_conflicted[k][j]) {
            show_move_animation(i, j, k, j);
            board[k][j] += board[i][j];
            board[i][j] = 0;
            //add score
            score += board[k][j];
            update_score(score);
            has_conflicted[k][j] = true;
            break;
          }
        }
      }
    }
  }
}
setTimeout('update_board_view()', 200);
return true;
}

//判断游戏成功或失败
function is_gameover() {
  for (var i = 0; i < 4; i++) {
    for (var j = 0; j < 4; j++) {
      if (board[i][j] == 2048) {
        update_score(success_string);
      }
    }
  }
}

```

动手实践是学习 IT 技术最有效的方式!

开始实验

```
        return;
    }
}
if (nospace(board) && nomove(board)) {
    gameover();
}
}

//游戏结束时更新游戏失败的文字
function gameover() {
    update_score(gameover_string);
}
```

在 support.js 中加入

动手实践是学习 IT 技术最有效的方式!

开始实验

```

//判断是否能向左移动
function can_move_left(board) {
  for (var i = 0; i < 4; i++) {
    for (var j = 1; j < 4; j++) {
      if (board[i][j] != 0) {
        if (board[i][j - 1] == 0 || board[i][j] == board[i][j - 1]) {
          return true;
        }
      }
    }
  }
  return false;
}

//判断是否能向右移动
function can_move_right(board) {
  for (var i = 0; i < 4; i++) {
    for (var j = 2; j >= 0; j--) {
      if (board[i][j] != 0) {
        if (board[i][j + 1] == 0 || board[i][j] == board[i][j + 1]) {
          return true;
        }
      }
    }
  }
  return false;
}

//判断是否能向上移动      动手实践是学习 IT 技术最有效的方式!      开始实验
function can_move_up(board) {
  for (var j = 0; j < 4; j++) {
    for (var i = 1; i < 4; i++) {
      if (board[i][j] != 0) {
        if (board[i - 1][j] == 0 || board[i - 1][j] == board[i][j]) {
          return true;
        }
      }
    }
  }
  return false;
}

//判断是否能向下移动
function can_move_down(board) {
  for (var j = 0; j < 4; j++) {
    for (var i = 2; i >= 0; i--) {
      if (board[i][j] != 0) {
        if (board[i + 1][j] == 0 || board[i + 1][j] == board[i][j]) {
          return true;
        }
      }
    }
  }
  return false;
}

//判断水平方向上时候是否有空格子
function no_block_horizontal(row, col1, col2, board) {
  for (var i = col1 + 1; i < col2; i++) {
    if (board[row][i] != 0) {
      return false;
    }
  }
  return true;
}

//判断垂直方向上时候是否有空格子
function no_block_vertical(col, row1, row2, board) {
  for (var i = row1 + 1; i < row2; i++) {
    if (board[i][col] != 0) {
      return false;
    }
  }
  return true;
}

//判断是否还能移动
function nomove(board) {

```

```
    if (can_move_down(board) || can_move_up(board) || can_move_right(board) || can_move_left(board)) {
        return false;
    }
    return true;
}
```

在 showanimation.js 中加入

```
//格子移动时有动画效果
function show_move_animation(fromx, fromy, tox, toy) {
    var number_cell = $('#number_cell_' + fromx + '_' + fromy);
    number_cell.animate({
        top: get_pos_top(tox, toy),
        left: get_pos_left(tox, toy)
    }, 200);
}
```

到此，我们的网页版 2048 就完成了。

## 五、实验总结

本课程中，通过使用 HTML+CSS+JavaScript+jQuery 我们实现了一个网页版的 2048 游戏，并学会如何让它兼容移动设备。相信通过本课程的学习，你可以加深对前对技术的理解和提高综合运用能力。

## 六、课后习题

动手实践是学习 IT 技术最有效的方式！

[开始实验](#)

1. 试着将 2048 改成 4096。
2. 当单元格里的数字超过 3 位时，会溢出单元格，试着修复它。

### 课程教师



**Christian**

共发布过17门课程

Web前端资深培训讲师，大型电子商务软件产品经理、交互设计师、用户体验师

[查看老师的所有课程 > \(/teacher/20407\)](#)

### 前置课程

- [jQuery基础入门 \(/courses/51\)](#)
- [HTML基础入门 \(/courses/19\)](#)
- [Javascript基础（新版） \(/courses/21\)](#)
- [CSS速成教程 \(/courses/53\)](#)

### 进阶课程

- [网页版扫雷 \(/courses/144\)](#)
- [网页版-别踩白块游戏 \(/courses/306\)](#)



## 动手做实验，轻松学IT



公司

<http://weibo.com/shiyanlou2013>



合作

[关于我们 \(/aboutus\)](#)

[我要投稿 \(/contribute\)](#)