

# JavaScript

## ■ JavaScript

### 1、JavaScript について

JavaScript (ジャバスクリプト) とは、オブジェクト指向スクリプト言語である。主にウェブブラウザなどのクライアントサイドで実装され、動的なウェブサイトの構築や、RIA などの高度なユーザインタフェースの開発に用いられる。

(Wikipedia より引用)

### 2、歴史・・・ブラウザ毎に実装されている JavaScript バージョンが微妙に異なる。

- 1.0(1996年3月。NN2.0。IE3.0)
- 1.1(1996年8月。NN3.0)
- 1.2(1997年7月。NN4.0-4.05)
- 1.3 = ECMA-262 1st edition / 2nd edition(1998年10月。NN4.06-4.7x。IE4.0。Opera5.0)
- 1.4(?。Netscape Server。Opera6.0)
- 1.5 = ECMA-262 3rd edition(2000年11月。NN6.0。FireFox1.0。IE5.5-8.0)
- 1.6(2005年11月。NN7.0-8.0。FireFox1.5。Opera7.0-9.0。Safari3.0,3.1)
- 1.7(2006年10月。FireFox2.0。Safari3.2-5.1。Chrome1.0-19.0)
- 1.8(2008年7月。FireFox3.0)
- 1.8.1(?。FireFox3.5)
- 1.9 = ECMAScript 5(FireFox4.0-11.0)

ECMAScript(エクマ スクリプト)についての詳細は、[「http://ja.wikipedia.org/wiki/ECMAScript」](http://ja.wikipedia.org/wiki/ECMAScript)を参照してください。

### 3、文法（一部）

(1) 文末

;(セミコロン)を書く。

(2) ブロックの開始/終了

「{」と「}」

例)

```
if (x == 1) {  
    // 処理  
} else {  
    // 処理2  
}
```

(3) コメント

1行コメントは「//」。複数行は「/\* \*/」

(4) for

```
for (i = 0; i <= 10; i++) {  
    // 処理  
}
```

(5) if / switch

java と同じなので説明省略 (-\_-:

(6) 比較演算子

== : 等しい

=== : 式が等しく型も等しい

!= : 等しくない

> : より大きい

>= : 以上

< : より小さい

<= : 以下

## 4、課題

### ビミョーな勤務表

2011 年 5 月分

曜日	有	欠	仕事	入社	退社	遅早	定時	普通	超過	深夜	備考
日	<input type="checkbox"/>	<input type="checkbox"/>	ムリ	10:00	12:00				2.0		
月	<input type="checkbox"/>	<input type="checkbox"/>	ムリ	15:00	23:00	5.0	2.5	0.5	4.0	1.0	

(1)以下のプログラムを作成し、ファイル名「kadai.html」で保存してください。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head>
<title>ビミョーな勤務表</title>
<meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
<meta http-equiv="Pragma" content="no-cache">
<meta http-equiv="Content-Style-Type" content="text/css">
<style type="text/css">
a
{
text-decoration: none;
}
</style>
<script language="javascript" type="text/javascript" src="test1.js"></script>
<script language="javascript" type="text/javascript" src="test2.js"></script>
<script language="javascript" type="text/javascript" src="test3.js"></script>
<script language="javascript" type="text/javascript" src="test4.js"></script>
<script language="javascript" type="text/javascript" src="test5.js"></script>
<script language="javascript" type="text/javascript" src="test6.js"></script>
<script language="javascript" type="text/javascript" src="test7.js"></script>
<script language="javascript" type="text/javascript" src="test8.js"></script>
<script language="javascript" type="text/javascript" src="test9.js"></script>
<script language="javascript" type="text/javascript" src="test10.js"></script>
<script language="javascript" type="text/javascript" src="test11.js"></script>
<script language="javascript" type="text/javascript" src="test12.js"></script>
<script language="javascript" type="text/javascript" src="test13.js"></script>
<script language="javascript" type="text/javascript" src="test14.js"></script>
<!-- JavaScriptが無効になっている時の処理 -->
<noscript>
<br><br>
<font size="3"><b>JavaScript</b>を有効してください。<br><br>
<b>Internet Explorer</b>の場合<br>
(<b>JavaScript</b>を有効にした後に<b>更新</b>ボタンを押してください。)<br><br>
<b>Netscape Communicator</b>の場合<br>
```



```
<!--欠勤-->
<td width="20" align="center">
  <input id="yasumi" name="yasumi" type="checkbox" value="" onclick="kinmu(0,'#ff0000')">
</td>
<!-- 仕事 -->
<td width="180" align="center">
  <select id="job" name="job" onchange="kinmu(0,'#ff0000')">
    <option selected="selected" value="001">△▽</option>
  </select>
</td>
<!-- 入社時間 -->
<td width="100">
  <select id="starttime" name="starttime" onchange="kinmu(0,'#ff0000')">
    <option selected="selected" value="入社時間を選択">入社時間を選択</option>
    <option value="9:00">9:00</option>
    <option value="10:00">10:00</option>
    <option value="11:00">11:00</option>
    <option value="12:00">12:00</option>
    <option value="13:00">13:00</option>
    <option value="14:00">14:00</option>
    <option value="15:00">15:00</option>
    <option value="16:00">16:00</option>
    <option value="17:00">17:00</option>
  </select>
</td>
<!-- 退社時間 -->
<td width="100">
  <select id="endtime" name="endtime" onchange="kinmu(0,'#ff0000')">
    <option selected="selected" value="退社時間を選択">退社時間を選択</option>
    <option value="9:00">9:00</option>
    <option value="10:00">10:00</option>
    <option value="11:00">11:00</option>
    <option value="12:00">12:00</option>
    <option value="13:00">13:00</option>
    <option value="14:00">14:00</option>
    <option value="15:00">15:00</option>
    <option value="16:00">16:00</option>
    <option value="17:00">17:00</option>
    <option value="18:00">18:00</option>
    <option value="19:00">19:00</option>
    <option value="20:00">20:00</option>
    <option value="21:00">21:00</option>
    <option value="22:00">22:00</option>
    <option value="23:00">23:00</option>
  </select>
</td>
```

```

</select>
</td>
<!-- 遅早 -->
<th width="50">
  <input id="tikoku" name="tikoku" type="text" value="0" size="4" maxlength="5" readonly="readonly">
</th>
<!-- 定時 -->
<th width="50">
  <input id="fixedtime" name="fixedtime" type="text" value="0" size="4" maxlength="5" readonly="readonly">
</th>
<!-- 普通 -->
<th width="50">
  <input id="commontime" name="commontime" type="text" value="0" size="4" maxlength="5"
readonly="readonly">
</th>
<!-- 超過 -->
<th width="50">
  <input id="excesstime" name="excesstime" type="text" value="0" size="4" maxlength="5" readonly="readonly">
</th>
<!-- 深夜 -->
<th width="50">
  <input id="nighttime" name="nighttime" type="text" value="0" size="4" maxlength="5" readonly="readonly">
</th>
<!-- 備考 -->
<td width="40">
  <input id="note" name="note" type="text" value="" size="20" maxlength="255">
</td>
</tr>
<!-- 2 -->
<tr>
<!-- 曜日 -->
<td width="20" bgcolor="#ffffff">月</td>
<!-- 日付 -->
<td width="20" align="center" bgcolor="#ffffff">2</td>
<!-- 有給休暇 -->
<td width="20" align="center">
  <input id="yuukyu" name="yuukyu" type="checkbox" value="" onclick="kinmu(1,'#ffffff')">
</td>
<!-- 欠勤 -->
<td width="20" align="center">
  <input id="yasumi" name="yasumi" type="checkbox" value="" onclick="kinmu(1,'#ffffff')">
</td>
<!-- 仕事 -->
<td width="180" align="center">

```

```

<select id="job" name="job" onchange="kinmu(1,'#ffffff')">
  <option selected="selected" value="001">△▽</option>
</select>
</td>
<!-- 入社時間 -->
<td width="100">
  <select id="starttime" name="starttime" onchange="kinmu(1,'#ffffff')">
    <option selected="selected" value="入社時間を選択">入社時間を選択</option>
    <option value="9:00">9:00</option>
    <option value="10:00">10:00</option>
    <option value="11:00">11:00</option>
    <option value="12:00">12:00</option>
    <option value="13:00">13:00</option>
    <option value="14:00">14:00</option>
    <option value="15:00">15:00</option>
    <option value="16:00">16:00</option>
    <option value="17:00">17:00</option>
  </select>
</td>
<!-- 退社時間 -->
<td width="100">
  <select id="endtime" name="endtime" onchange="kinmu(1,'#ffffff')">
    <option selected="selected" value="退社時間を選択">退社時間を選択</option>
    <option value="9:00">9:00</option>
    <option value="10:00">10:00</option>
    <option value="11:00">11:00</option>
    <option value="12:00">12:00</option>
    <option value="13:00">13:00</option>
    <option value="14:00">14:00</option>
    <option value="15:00">15:00</option>
    <option value="16:00">16:00</option>
    <option value="17:00">17:00</option>
    <option value="18:00">18:00</option>
    <option value="19:00">19:00</option>
    <option value="20:00">20:00</option>
    <option value="21:00">21:00</option>
    <option value="22:00">22:00</option>
    <option value="23:00">23:00</option>
  </select>
</td>
<!-- 遅早 -->
<th width="50">
  <input id="tikoku" name="tikoku" type="text" value="0" size="4" maxlength="5" readonly="readonly">
</th>

```

```
<!-- 定時 -->
<th width="50">
  <input id="fixedtime" name="fixedtime" type="text" value="0" size="4" maxlength="5" readonly="readonly">
</th>
<!-- 普通 -->
<th width="50">
  <input id="commontime" name="commontime" type="text" value="0" size="4" maxlength="5"
readonly="readonly">
</th>
<!-- 超過 -->
<th width="50">
  <input id="excesstime" name="excesstime" type="text" value="0" size="4" maxlength="5" readonly="readonly">
</th>
<!-- 深夜 -->
<th width="50">
  <input id="nighttime" name="nighttime" type="text" value="0" size="4" maxlength="5" readonly="readonly">
</th>
<!-- 備考 -->
<td width="40">
  <input id="note" name="note" type="text" value="" size="20" maxlength="255">
</td>
</tr>
</table>
</table>
</form>
</body>
</html>
```



(2)以下のプログラムを作成し、ファイル名「test1.js」で保存してください。

```
/**
 * 処理内容 更新 入社、退社チェック 入社または退社だけ
 *          設定してある場合はエラーメッセージ
 *
 * 引数 tmp1 月の日数
 *
 * 戻り値 エラーが無い場合は true
 *        エラーがある場合は false
 *
 * 備考 仕事が"仕事を選択"以外が
 *       選択されている時にエラーチェックをする
 *
 * 更新履歴
 */
function btnUpdate(tmp1) {
  var selstart = 0; //入社時刻
  var selend = 0; //退社時刻
  var job = 0; //仕事
  var i = 0;
  var msg = "";
  var msg2 = "";
  for (i=0; i <= tmp1-1; i++) {
    selstart = document.work.starttime[i].selectedIndex;
    selend = document.work.endtime[i].selectedIndex;
    job = document.work.job[i].selectedIndex;
    //仕事が選択されていない
    if (job == 0) {
      //入社時間が選択されていない
      if (selstart == 0) {
        //退社時間が選択されていない
        if (selend == 0) {
          //退社時間が選択されている
        } else {
          msg = msg+(i+1)+"日の仕事が選択されていません¥n";
          msg = msg+(i+1)+"日の入社時間が選択されていません¥n";
        }
      }
      //入社時間が選択されている
    } else {
      msg = msg+(i+1)+"日の仕事が選択されていません¥n";
      //退社時間が選択されていない
      if (selend == 0) {
        msg = msg+(i+1)+"日の退社時間が選択されていません¥n";
      }
    }
  }
}
```

```

    }
    //仕事が選択されている
  } else {
    //出社時間が選択されていない
    if (selstart == 0) {
      msg = msg+(i+1)+"日の出社時間が選択されていません¥n";
      //退社時間が選択されていない
      if (selend == 0) {
        msg = msg+(i+1)+"日の退社時間が選択されていません¥n";
      }
      //出社時間が選択されている
    } else {
      //退社時間が選択されていない
      if (selend == 0) {
        msg = msg+(i+1)+"日の退社時間が選択されていません¥n";
      }
    }
  }
}

//エラーメッセージチェック
if (msg != "") {
  alert(msg);
  return false;
}
return true;
}

```

### (3)以下のプログラムを作成し、ファイル名「test2.js」で保存してください。

```
/**
 * 処理内容 出勤退社の計算値を画面に表示する関数
 *
 * 引数 selecIndex
 *     selecColor '#ff0000' : 日曜
 *     '#0cb0e0' : 土曜
 *     '#ffffff' : 平日
 *
 * 戻り値 なし
 *
 * 更新履歴
 */
function kinmu(selecIndex,selecColor) {
    var stindex = document.work.starttime[selecIndex].selectedIndex; //出社
    var syussia = document.work.starttime[selecIndex].options[stindex].text;
    var enindex = document.work.endtime[selecIndex].selectedIndex; //退社
    var taisya = document.work.endtime[selecIndex].options[enindex].text
    var keisanSyussia = 0; //出社の計算値
    var keisanTaisya = 0; //退社の計算値
    var replacethis = RegExp(":", "g");
    var keisantikoku = 0; //遅早の計算値
    var keisanteiji = 0; //定時の計算値
    var keisanfutuu = 0; //普通の計算値
    var keisancyouka = 0; //超過の計算値

    //有給休暇または欠勤にチェックがある場合の処理
    if ((document.work.yuukyu[selecIndex].checked == true) ||
        (document.work.yasumi[selecIndex].checked == true)) {

        //遅早、定時、普通、超過、深夜をクリアする
        document.work.tikoku[selecIndex].value = "";
        document.work.fixedtime[selecIndex].value = "";
        document.work.commontime[selecIndex].value = "";
        document.work.excesstime[selecIndex].value = "";
        document.work.nighttime[selecIndex].value = "";
        //出社、退社、仕事のindex番号0に戻す
        document.work.starttime[selecIndex].selectedIndex = 0;
        document.work.endtime[selecIndex].selectedIndex = 0;
        document.work.job[selecIndex].selectedIndex = 0;
    }

    if ((syussia != "出社時間を選択") && (taisya != "退社時間を選択")) {
        //有給休暇と欠勤のチェックが無い状態で
```

```

if ((document.work.yuukyu[selecIndex].checked == false) &&
(document.work.yasumi[selecIndex].checked == false)) {

//① 出社の計算値を求める関数
syusssa = syusssa.replace(replacethis,".");

keisanSyusssa = calcsyusssa(parseFloat(syusssa));

//② 退社の計算値を求める関数
taisya = taisya.replace(replacethis,".");
keisanTaisya = calctaisya(parseFloat(taisya));

//土曜、日曜、祝日ではない時に遅刻、早退、定時、普通残業の計算を行う
if ((selecColor != '#0cb0e0') && (selecColor != '#ff0000')) {

//③ 遅早の出力する値を求める関数
keisantikoku = calctikoku(keisanSyusssa,keisanTaisya,selecIndex);

//④ 定時の出力する値を求める関数
keisanteiji = calcteiji(keisantikoku,selecIndex);

//⑤ 普通の出力する値を求める関数
keisanfutuu = calcfutuu(keisanTaisya,selecIndex);
} else {
//遅早、定時、普通をクリアする
document.work.tikoku[selecIndex].value = "";
document.work.fixedtime[selecIndex].value = "";
document.work.commontime[selecIndex].value = "";
}

//普通残業がある時に超過、深夜残業の計算を行う
if (keisanfutuu != "") {
//⑥ 超過の出力する値を求める関数
keisancyouka = calccyouka(keisanTaisya,selecIndex);

//⑦ 深夜の出力する値を求める関数
calcnight(keisanTaisya,selecIndex);
} else {
//普通残業が無くて、休日の場合は休日の超過、深夜残業の計算を行う
if ((selecColor == '#0cb0e0') || (selecColor == '#ff0000')) {
//休日の超過の計算
calckyujitucyouka(keisanSyusssa,keisanTaisya,selecIndex);

//休日の深夜の計算

```

```

calckyujitunight(keisanTaisya,selecIndex);

} else {
    //超過、深夜をクリアする
    document.work.excesstime[selecIndex].value = "";
    document.work.nighttime[selecIndex].value = "";
}
}
} else {
    //遅早、定時、普通、超過、深夜をクリアする
    document.work.tikoku[selecIndex].value = "";
    document.work.fixedtime[selecIndex].value = "";
    document.work.commontime[selecIndex].value = "";
    document.work.excesstime[selecIndex].value = "";
    document.work.nighttime[selecIndex].value = "";
    //出社、退社、仕事のindex番号0に戻す
    document.work.starttime[selecIndex].selectedIndex = 0;
    document.work.endtime[selecIndex].selectedIndex = 0;
    document.work.job[selecIndex].selectedIndex = 0;
}
} else {
    //遅早、定時、普通、超過、深夜をクリアする
    document.work.tikoku[selecIndex].value = "";
    document.work.fixedtime[selecIndex].value = "";
    document.work.commontime[selecIndex].value = "";
    document.work.excesstime[selecIndex].value = "";
    document.work.nighttime[selecIndex].value = "";
}
}
}

```

(4)以下のプログラムを作成し、ファイル名「test3.js」で保存してください。

```
/**
 * 処理内容 出社の計算値を返す関数
 *
 * 引数    startJikou  出社時刻
 *
 * 戻り値  出社の計算値
 *
 * 更新履歴
 */
function calcsyussya(startJikou) {
  var startvalue = 0; //出社の計算値
  var ichi = 0;
  var replacethis = "";

  //出社時刻が 0:00 の場合
  if (startJikou == parseFloat("0.00")) {
    startvalue = "";
  } else {
    //出社時刻が 1:00 より小さい場合
    if (startJikou <= parseFloat("1.00")) {
      startvalue = parseFloat("23.45");
    } else {
      //出社時刻が 9:00 より小さい場合
      if (startJikou <= parseFloat("9.00")) {
        startvalue = parseFloat("9.00");
      } else {
        //出社時刻が 11:45 より小さい場合 Excel 関数の CELING(出社時刻,"00:15")と同じようなことする。
        if (startJikou <= parseFloat("11.45")) {
          startvalue = startJikou; // + parseFloat("00.15");
          //X.6 の時だけ切り上げの処理を行う
          startvalue = new String(startvalue);
          ichi = startvalue.indexOf(".6");
          if (ichi > 0) {
            replacethis = RegExp(startvalue,"g");
            startvalue = startvalue.replace(replacethis,
              Math.ceil(parseFloat(startvalue)));
          }
        } else {
          //出社時刻が 13:00 より小さい場合
          if (startJikou <= parseFloat("13.00")) {
            startvalue = parseFloat("13.00");
          } else {
            //出社時刻が 23:45 より小さい場合 Excel 関数の CELING(出社時刻,"00:15")と同じようなことする。

```

```
if (startJikou <= parseFloat("23.45")) {
  startvalue = startJikou; // + parseFloat("00.15");
  //X.6 の時だけ切り上げの処理を行う
  startvalue = new String(startvalue);
  ichi = startvalue.indexOf(".6");
  if (ichi > 0) {
    replacethis = RegExp(startvalue,"g");
    startvalue = startvalue.replace(replacethis,
      Math.ceil(parseFloat(startvalue)));
  }
} else {
  startvalue = parseFloat("1:00");
}
}
}
}
}
}
return parseFloat(startvalue);
}
```

**(5)以下のプログラムを作成し、ファイル名「test4.js」で保存してください。**

```
/**
 * 処理内容  退社の計算値を返す関数
 *
 * 引数    endJikou  退社時刻
 *
 * 戻り値  退社の計算値
 *
 * 更新履歴
 */
function calctaisya(endJikou) {
  var endvalue = 0;  //退社の計算値

  //退社時刻が 0:00 の場合
  if (endJikou == parseFloat("0.00")) {
    endvalue = "";
  } else {
    //退社時刻が 1:30 より小さい場合
    if (endJikou < parseFloat("1.30")) {
      endvalue = parseFloat("1.00");
    } else {
      //退社時刻が 12:00 より小さい場合
      if (endJikou < parseFloat("12.00")) {
        //Excel 関数の FLOOR(退社時刻,"00:30")と同じようなことする。
        endvalue = parseFloat(endJikou);
      } else {
        //退社時刻が 13:30 より小さい場合
        if (endJikou < parseFloat("13.30")) {
          endvalue = parseFloat("12.00");
        } else {
          //退社時刻が 17:30 より小さい場合
          if (endJikou < parseFloat("17.30")) {
            //Excel 関数の FLOOR(退社時刻,"00:30")と同じようなことする。
            endvalue = parseFloat(endJikou);
          } else {
            //退社時刻が 18:30 より小さい場合
            if (endJikou < parseFloat("18.30")) {
              endvalue = parseFloat("17.30");
            } else {
              //Excel 関数の FLOOR(退社時刻,"00:30")と同じようなことする。
              endvalue = parseFloat(endJikou);
            }
          }
        }
      }
    }
  }
}
```



```
    }  
  }  
}  
return endvalue;  
}
```

**(5)以下のプログラムを作成し、ファイル名「test4.js」で保存してください**

```
/**
 * 処理内容 遅早の計算値を求めて画面に出力する関数
 *
 * 引数    thesyussya 出社の計算値
 *         thetaisya  退社の計算値
 *         theindex
 *
 * 戻り値  遅早の計算値
 *
 * 更新履歴
 */
function calctikoku(thesyussya,thetaisya,theindex) {
  var tikoku = "";    //遅刻の計算値
  var soutai = "";   //早退の計算値
  var outputvalue = ""; //出力文字列
  var tmpvalue;      //遅刻と早退
  var i=0;
  var j=0;
  var k=0;
  var strHour = "";  //時
  var strMinute = ""; //分
  var replacethis = "";
  var ich = 0;

  /***** 遅刻の計算 *****/

  //出社の計算値が 9:00 以上で
  if (thesyussya > parseFloat("9.00")) {
    //出社の計算値が 12:00 以下の時
    if (thesyussya <= parseFloat("12.00")) {
      tikoku = thesyussya - parseFloat("9.00");
    } //それ以外
    } else {
      tikoku = thesyussya - parseFloat("9.00") - parseFloat("1.00");
    }
  }

  /***** 早退の計算 *****/

  //退社の計算値が 17:30 以下で
  if (thetaisya < parseFloat("17.30")) {
    //退社の計算値が 9:00 以上で
    if (thetaisya > parseFloat("9.00")) {
```

```

//退社の計算値が 13:00 以下の場合
if (thetaisya <= parseFloat("13.00")) {
    soutai = parseFloat("17.30") - thetaisya - parseFloat("1.00");
//それ以外
} else {
    soutai = parseFloat("17.30") - thetaisya;
}
}
}

```

```

//画面に出力する(遅刻,早退がある場合)

```

```

//計算値 = 遅刻 + 早退

```

```

outputvalue = tikoku + soutai;

```

```

//X.6の時だけ切り上げの処理を行う

```

```

outputvalue = new String(outputvalue);
ichi = outputvalue.indexOf(".6");
if (ichi > 0) {
    replacethis = RegExp(outputvalue,"g");
    outputvalue = outputvalue.replace(replacethis,
        Math.ceil(parseFloat(outputvalue)));
}

```

```

//X.75の時だけ切り上げの処理を行う

```

```

outputvalue = new String(outputvalue);
ichi = outputvalue.indexOf(".75");
if (ichi > 0) {
    replacethis = RegExp(outputvalue,"g");
    outputvalue = outputvalue.replace(replacethis,
        Math.ceil(parseFloat(outputvalue)) + ".15");
}

```

```

//計算値が Null ではない場合

```

```

if (outputvalue != "") {
    outputvalue = new String(outputvalue);

```

```

//計算値から時間を取得する関数

```

```

strHour = getHourValue(outputvalue);

```

```

//計算値から分を取得する関数

```

```

strMinute = getMinuteValue(outputvalue);

```

```

}

```

```

//strHour が数値の場合で
if (isNaN(strHour) == false) {

    //strMinute が数値ではない(NaN)時は strMinute に"0"を代入する
    if (isNaN(strMinute) == true) {
        strMinute = 0;
    }
    strMinute = new String(strMinute);

    //strMinute が"1"または"2"の場合は strMinute に"00"を代入する
    if ((strMinute == "1") || (strMinute == "2")) {
        strMinute = "00";
    }

    //strMinute の 2 文字目が"4"の場合は
    //strMinute に 1 文字目 + (2 文字目 + 1)を代入する
    if (strMinute.charAt(1) == "4") {
        strMinute = strMinute.charAt(0) + (parseFloat(strMinute.charAt(1)) + 1);
    }

    //strMinute の 2 文字目が""の場合は
    //strMinute に 1 文字目 + "0"を代入する
    if (strMinute.charAt(1) == "") {
        strMinute = strMinute.charAt(0) + "0";
    }
}

if (strMinute != "") {
    strMinute = parseFloat(strMinute);
} else {
    strMinute = "";
}

if ((strHour != "") && (strMinute != "")) {
    tmpvalue = strHour + "." + strMinute;
} else if ((strHour != "") && (strMinute == "")) {
    tmpvalue = strHour + "." + "0";
} else if ((strHour == "") && (strMinute != "")) {
    tmpvalue = "0" + "." + strMinute;
} else {
    tmpvalue = "";
}
outputvalue = tmpvalue;

```

```
//計算値を出力の書式にする関数
if (strHour != "") {
    outputvalue = valueFormat(strHour,strMinute);
} else {
    outputvalue = "";
}

//画面に出力
document.work.tikoku[theindex].value = outputvalue;

return tmpvalue;
}
```

**(7)以下のプログラムを作成し、ファイル名「test6.js」で保存してください**

```
/**
 * 処理内容 定時の計算値を求めて画面に出力する関数
 *
 * 引数    opttikoku  遅早の計算値
 *         theindex
 *
 * 戻り値  定時の計算値
 *
 * 更新履歴
 */
function calcTeiji(opttikoku,theindex) {
    var calcvalue = ""; //定時の計算値
    var outvalue = ""; //出力文字列
    var strHour = ""; //時
    var strMinute = ""; //分

    if (opttikoku != "") {
        calcvalue = parseFloat("7.30") - parseFloat(opttikoku);
    } else {
        calcvalue = parseFloat("7.30");
    }

    calcvalue = new String(calcvalue);
    //alert(calcvalue);

    //計算値から時間を取得する関数
    strHour = getHourValue(calcvalue);

    //計算値から分を取得する関数
    strMinute = getMinuteValue(calcvalue);

    //strHour が数値の場合で
    if (isNaN(strHour) == false) {
        //strMinute が数値ではない(NaN)時は strMinute に"0"を代入する
        if (isNaN(strMinute) == true) {
            strMinute = 0;
        }
        strMinute = new String(strMinute);

        //strMinute の 2 文字目が"4"の場合は
        //strMinute に 1 文字目 + (2 文字目 + 1)を代入する
        if (strMinute.charAt(1) == "4") {
            strMinute = strMinute.charAt(0) + (parseFloat(strMinute.charAt(1)) + 1);
        }
    }
}
```



**(8)以下のプログラムを作成し、ファイル名「test7.js」で保存してください**

```
/**
 * 処理内容 普通残業の計算値を求めて画面に出力する関数
 *
 * 引数    optTaisya    退社の計算値
 *         theindex
 *
 * 戻り値  普通残業の計算値
 *
 * 更新履歴
 */
function calcfutu(optTaisya,theindex) {
    var outvalue = "";

    //退社の計算値が 18:30 より大きい場合
    if (optTaisya >= parseFloat("18.30")) {
        outvalue = "0.5";
    } else {
        //退社の計算値が 9:00 より小さい場合
        if (optTaisya < parseFloat("9:00")) {
            outvalue = "0.5";
        } else {
            outvalue = "";
        }
    }
    //画面に出力
    document.work.commontime[theindex].value = outvalue;

    return outvalue;
}
```



**(9)以下のプログラムを作成し、ファイル名「test8.js」で保存してください**

```
/**
 * 処理内容 超過残業の計算値を求めて画面に出力する関数
 *
 * 引数  opttaisya  退社の計算値
 *       theindex
 *
 * 戻り値  超過残業の計算値
 *
 * 更新履歴
 */
function calccyouka(opttaisya,theindex) {
  var outvalue = ""; //出力文字列
  var cyouka = ""; //超過
  var strHour = ""; //時
  var strMinute = ""; //分

  //退社の計算値 >= 22:00 の時
  if (opttaisya >= parseFloat("22.00")) {
    cyouka = parseFloat("22.00") - parseFloat("18.00");
  } else {
    //退社の計算値 < 9:00 の時
    if (opttaisya < parseFloat("9.00")) {
      cyouka = parseFloat("22.00") - parseFloat("18.00");
    } else {
      cyouka = opttaisya - parseFloat("18.00");
    }
  }
  cyouka = new String(cyouka);

  //計算値から時間を取得する関数
  strHour = getHourValue(cyouka);

  //計算値から分を取得する関数
  strMinute = getMinuteValue(cyouka);

  //strHour が数値の場合で
  if (isNaN(strHour) == false) {
    //strMinute が数値ではない(NaN)時は strMinute に"0"を代入する
    if (isNaN(strMinute) == true) {
      strMinute = "0";
    }
  }
  strMinute = new String(strMinute);
}
```



(10)以下のプログラムを作成し、ファイル名「test9.js」で保存してください。

```
/**
 * 処理内容 超過残業(休日)の計算値を求めて画面に出力する関数
 *
 * 引数  optsyussya  出社の計算値
 *       opttaisya   退社の計算値
 *       theindex
 *
 * 戻り値 なし
 *
 * 更新履歴
 */
function calckyujitucyouka(optsyussya,opttaisya,theindex) {
  var outvalue = ""; //出力文字列
  var cyouka = ""; //超過(休日)
  var strHour = ""; //時
  var strMinute = ""; //分
  var ichi = "";
  var ichi2 = "";

  //出社の計算値がX.3の時に退社の計算値に".6"を足す
  optsyussya = new String(optsyussya);
  opttaisya = new String(opttaisya);

  //退社の計算値 < 9:00 で
  if (opttaisya < parseFloat("9.00")) {
    //出社の計算値 < 13:00 の時
    if (optsyussya < parseFloat("13.00")) {
      cyouka = parseFloat("22.00") - optsyussya - parseFloat("1.00");
    } else {
      cyouka = parseFloat("22.00") - optsyussya;
    }
  } else {
    //出社の計算値 < 13:00 で
    if (optsyussya < parseFloat("13.00")) {
      //退社の計算値 < 13:00 の時
      if (opttaisya < parseFloat("13.00")) {
        cyouka = opttaisya - optsyussya;
      } else {
        cyouka = opttaisya - optsyussya - parseFloat("1.00");
      }
    } else {
      cyouka = opttaisya - optsyussya;
    }
  }
}
```



```
    strMinute = "";
}

if (strMinute != "") {
    strMinute = parseFloat(strMinute);
} else {
    strMinute = "";
}

//計算値を出力の書式にする関数
outvalue = valueFormat(strHour,strMinute);

//画面に出力
document.work.excesstime[theindex].value = outvalue;
}
```

(11)以下のプログラムを作成し、ファイル名「test10.js」で保存してください。

```
/**
 * 処理内容 深夜残業の計算値を求めて画面に出力する関数
 *
 * 引数  opttaisya  退社の計算値
 *       theindex
 *
 * 戻り値 なし
 *
 * 更新履歴
 */
function calcnight(opttaisya,theindex) {
  var outvalue = ""; //出力文字列
  var night = ""; //深夜
  var strHour = ""; //時
  var strMinute = ""; //分

  //退社の計算値 >= 22:00 の時
  if (opttaisya >= parseFloat("22.00")) {
    night = opttaisya - parseFloat("22.00");
  } else {
    //退社の計算値 < 9:00 の時
    if (opttaisya < parseFloat("9.00")) {
      night = (opttaisya - parseFloat("1.00")) + (24 - parseFloat("22.00"));
    } else {
      night = "";
    }
  }
}

night = new String(night);

//計算値から時間を取得する関数
strHour = getHourValue(night);

//計算値から分を取得する関数
strMinute = getMinuteValue(night);

if (strHour != 0) {
  //strHour が数値の場合で
  if (isNaN(strHour) == false) {
    //strMinute が数値ではない(NaN)時はstrMinute に"0"を代入する
    if (isNaN(strMinute) == true) {
      strMinute = "0";
    }
  }
}
```

```
strMinute = new String(strMinute);

//strMinuteの2文字目が""の時は
//strMinuteに1文字目 + "0"を代入する
if (strMinute.charAt(1) == "") {
    strMinute = strMinute.charAt(0) + "0";
}
}
}

if (strMinute != "") {
    strMinute = parseFloat(strMinute);
} else {
    strMinute = "";
}

//計算値を出力の書式にする関数
outvalue = valueFormat(strHour,strMinute);

//画面に出力
document.work.nighttime[theindex].value = outvalue;
}
```

(12)以下のプログラムを作成し、ファイル名「test11.js」で保存してください。

```
/**
 * 処理内容 深夜残業(休日)の計算値を求めて画面に出力する関数
 *
 * 引数    opttaisya    退社の計算値
 *         theindex
 *
 * 戻り値  なし
 *
 * 更新履歴
 */
function calckyujitunight(opttaisya,theindex) {
    var outvalue = ""; //出力文字列
    var night = "";    //深夜(休日)
    var strHour = "";  //時
    var strMinute = ""; //分

    //退社の計算値 < 9:00の時
    if (opttaisya < parseFloat("9.00")) {
        night = (opttaisya - parseFloat("1.00")) + (24 - parseFloat("22.00"));
    } else {
        //退社の計算値 > 22:00の時
        if (opttaisya > parseFloat("22.00")) {
            night = opttaisya - parseFloat("22.00");
        } else {
            night = "";
        }
    }
}

night = new String(night);

//計算値から時間を取得する関数
strHour = getHourValue(night);

//計算値から分を取得する関数
strMinute = getMinuteValue(night);

if (strHour != 0) {
    if (isNaN(strHour) == false) {
        //strMinuteが数値ではない(NaN)時はstrMinuteに"0"を代入する
        if (isNaN(strMinute) == true) {
            strMinute = "0";
        }
        strMinute = new String(strMinute);
    }
}
```



```
//strMinuteの2文字目が""の時は
//strMinuteに1文字目 + "0"を代入する
if (strMinute.charAt(1) == "") {
    strMinute = strMinute.charAt(0) + "0";
}
}
}

if (strMinute != "") {
    strMinute = parseFloat(strMinute);
} else {
    strMinute = "";
}

//計算値を出力の書式にする関数
outvalue = valueFormat(strHour,strMinute);

//画面に出力
document.work.nighttime[theindex].value = outvalue;
}
```

(13)以下のプログラムを作成し、ファイル名「test12.js」で保存してください。

```
/**
 * 処理内容 計算値を出力の書式にする関数
 *
 * 引数  optHour  時
 *      optMinute 分
 *
 * 戻り値 出力の書式の値
 *
 * 更新履歴
 */
function valueFormat(optHour,optMinute) {
  var modoriti = 0; //戻り値
  var strchk = "";

  strchk = optHour + "." + optMinute;
  //strchkの値が"0.0"の時
  if (strchk == "0.0") {
    modoriti = "";
  // "0.0"以外
  } else {
    switch (optMinute) {
      //Minute(計算値)が0の時
      case 0:
        modoriti = optHour + ".0";
        break;
      //Minute(計算値)が15の時
      case 15:
        modoriti = optHour + ".25";
        break;
      //Minute(計算値)が30の時
      case 30:
        modoriti = optHour + ".5";
        break;
      //Minute(計算値)が45の時
      case 45:
        modoriti = optHour + ".75";
        break;
      //それ以外は
      default:
        modoriti = "";
        break;
    }
  }
}
```

```
return modoriti;  
}
```

(14)以下のプログラムを作成し、ファイル名「test13.js」で保存してください。

```
/**
 * 処理内容 計算値から時間を取得する関数
 *
 * 引数    optcalcvalue    計算値
 *
 * 戻り値  時
 *
 * 更新履歴
 */
function getHourValue(optcalcvalue) {
  var tmpHour = ""; //時
  var i = 0;

  //時
  for (i=0;i<optcalcvalue.length;i++) {
    if (optcalcvalue.charAt(i) != ".") {
      tmpHour = tmpHour + optcalcvalue.charAt(i);
    } else {
      //ループを抜ける
      break;
    }
  }
  return tmpHour;
}
```

(15)以下のプログラムを作成し、ファイル名「test14.js」で保存してください。

```
/**
 * 処理内容 計算値から分を取得する関数
 *
 * 引数    optcalcvalue    計算値
 *
 * 戻り値  分
 *
 * 更新履歴
 */
function getMinuteValue(optcalcvalue) {
  var tmpMinute = ""; //分
  var i = 0;
  var j = 0;
  var k = 0;

  //分
  for (i=0;i<optcalcvalue.length;i++) {
    if (optcalcvalue.charAt(i) == ".") {
      //ループを抜ける
      break;
    }
    j++;
  }

  for (i=j+1;i<optcalcvalue.length;i++) {
    if (k != 2) {
      tmpMinute = tmpMinute + optcalcvalue.charAt(i);
    } else {
      //ループを抜ける
      break;
    }
    k++;
  }
  return tmpMinute;
}
```

#### <参考情報>

とほほのWWW入門

<http://www.tohoho-web.com/www.htm>