

```

<!--
/*****
/* 太陽・月の出没時刻の計算 */
/*
/* 参考: 「日の出・日の入りの計算」 長沢工著 地人書館 */
/*
/* 2010.12.31 made for MAC H/P by K. Kanatsu */
*****/

/* ##### 太陽・月 共通 ##### */

function t(y,m,d){
/*-----*/
/* 時刻変数Tを求める */
/* (太陽位置の略算式p.71, 月位置の略算式p.124の変数) */
/* k : 2000年1月1日力学時正午からの経過日数 */
/* input:y年m月d.h日(JST) */
/* output:時刻変数T */
/*-----*/
var k;
var dd; /* 日付の整数部 */
var hh; /* 日付の小数部 = 時間 */
y -= 2000; /* 2000年が起点 */
dd = Math.floor( d );
hh = d - dd ;
if ( m < 3 ) { /* 1,2月の時は、 */
y -= 1; /* 前年の */
m += 12; /* 13,14月として計算 */
}
k = 365 * y + 30 * m + dd - 33.5 - 9/24
+ Math.floor( ( 3 * ( m+ 1 ) ) / 5 ) + Math.floor( y / 4 );
return ( k + hh + 64 / 86400 ) / 365.25;
}

function rad(deg){
/*-----*/
/* 角度変換(「度」から「ラジアン」へ) */
/* (360度以上の値は360度以下にして計算) */
/*-----*/
return ( deg % 360 ) * Math.PI / 180;
}

function riseandset(year,month,day,lon,lat){
/* 太陽、月の出入り時刻の表示 */
dsp_sun(year,month,day+0.25,lon,lat);
dsp_moon(year,month,day+0.5,lon,lat);
}

/* ##### 太陽 ##### */

function lambda_sun(t){
/*-----*/
/* 太陽位置の略算式(p.71) */
/* lambda : 太陽の視黄経 */
/*-----*/
return 280.4603 + 360.00769 * t
+ ( 1.9146 - 0.00005 * t ) * Math.sin( rad(357.538 + 359.991 * t ) )
+ 0.0200 * Math.sin( rad( 355.05 + 719.981 * t ) )
+ 0.0048 * Math.sin( rad( 234.95 + 19.341 * t ) )
+ 0.0020 * Math.sin( rad( 247.1 + 329.64 * t ) )
+ 0.0018 * Math.sin( rad( 297.8 + 4452.67 * t ) )
+ 0.0018 * Math.sin( rad( 251.3 + 0.20 * t ) )
+ 0.0015 * Math.sin( rad( 343.2 + 450.37 * t ) )
+ 0.0013 * Math.sin( rad( 81.4 + 225.18 * t ) )
+ 0.0008 * Math.sin( rad( 132.5 + 659.29 * t ) )
+ 0.0007 * Math.sin( rad( 153.3 + 90.38 * t ) )
+ 0.0007 * Math.sin( rad( 206.8 + 30.35 * t ) )
+ 0.0006 * Math.sin( rad( 29.8 + 337.18 * t ) )
+ 0.0005 * Math.sin( rad( 207.4 + 1.50 * t ) )
+ 0.0005 * Math.sin( rad( 291.2 + 22.81 * t ) )
+ 0.0004 * Math.sin( rad( 234.9 + 315.56 * t ) )
+ 0.0004 * Math.sin( rad( 157.3 + 299.30 * t ) )
+ 0.0004 * Math.sin( rad( 21.1 + 720.02 * t ) )
+ 0.0003 * Math.sin( rad( 352.5 + 1079.97 * t ) )
+ 0.0003 * Math.sin( rad( 329.7 + 44.43 * t ) );
}

function q(t){
/*-----*/
/* 太陽位置の略算式(p.71) */
/*-----*/

```

```

/* q : 太陽の距離を求めるための変数 */
/*-----*/
return ( 0.007256 - 0.000002 * t ) * Math.sin( rad( 267.54 + 359.991 * t ) )
+ 0.000091 * Math.sin( rad( 265.1 + 719.98 * t ) )
+ 0.000030 * Math.sin( rad( 90.0 ) )
+ 0.000013 * Math.sin( rad( 27.8 + 4452.67 * t ) )
+ 0.000007 * Math.sin( rad( 254 + 450.4 * t ) )
+ 0.000007 * Math.sin( rad( 156 + 329.6 * t ) );
}

function sun(y, m, d, lon, lat, dk, rs) {
/*-----*/
/* input: y=year, m=month, d=day(day+hour), lon=経度, lat=緯度, */
/* dk=0(出沒) or -18(薄明) rs=0(出)1(没) */
/* output: hour (出沒時刻) */
/*-----*/
do {
/*-----*/
/* 1. 時刻変数Tの計算 p.71 */
/*-----*/
var tt; /* 指定日時の経過日数 */
tt = t(y, m, d);

/*-----*/
/* 2. 太陽の視黄経 λs、距離rの計算 p.73 */
/*-----*/
var lambda_s;
lambda_s = lambda_sun( tt );
lambda_s %= 360;
var r;
r = Math.pow( 10 , q( tt ) );

/*-----*/
/* 3. 太陽の赤経 α、赤緯 δ の計算 p.74 */
/*-----*/
var epsilon; /* 黄道傾角 */
epsilon = 23.439291 - 0.000130042 * tt;

var tan_alpha;
var sin_delta;
tan_alpha = Math.tan( rad(lambda_s) ) * Math.cos( rad(epsilon) );
sin_delta = Math.sin( rad(lambda_s) ) * Math.sin( rad(epsilon) );

var alpha; /* 赤経 radian */
var delta; /* 赤緯 radian */
alpha = Math.atan(tan_alpha);
delta = Math.asin(sin_delta);

var alpha_deg; /* 赤経 degree */
var delta_deg; /* 赤緯 degree */
alpha_deg = 180 * alpha / Math.PI;
if ( alpha_deg < 0 ) alpha_deg += 180; /* 赤経の値の補正 */
if ( lambda_s >= 180 ) alpha_deg += 180; /* 象限判定 */

delta_deg = 180 * delta / Math.PI;

/*-----*/
/* 4. 恒星時 θ の計算 p.76 */
/*-----*/
var theta;
var hh= d - Math.floor( d );
theta = 325.4606 + 360.007700536 * tt
+ 0.00000003879 * tt * tt + 360 * hh + lon;
theta %= 360;

/*-----*/
/* 5. 出沒高度 k の計算 p.37 */
/*-----*/
var k; /* 太陽の出沒高度 */
/* dk(高度) → 0:出沒, -18:薄明 */
if ( dk == 0 ) k = -0.266994 / r - 0.585556 + 0.0024428 / r + dk;
else k = 0.0024428 / r + dk;

/*-----*/
/* 6. 出沒高度k に対応する時角tk への換算 p.38 */
/*-----*/
cos_tk = ( Math.sin( rad( k ) ) - Math.sin( delta ) * Math.sin( rad( lat ) ) )
/ ( Math.cos( delta ) * Math.cos( rad( lat ) ) );
var tk;
tk = Math.acos( cos_tk );
if ( rs == 0 ) /* 「出」のとき */

```

riseandset2.js

```
    tk_deg = 180 * tk / Math.PI * -1 ;    /* マイナス値にする */
else
    tk_deg = 180 * tk / Math.PI;

/*=====*/
/* 7. 恒星時theta から太陽の時角t を計算する p.39 */
/* t1 = theta - alpha */
/*=====*/
var t1;
t1 = theta - alpha_deg;

/*=====*/
/* 8. 仮定時刻d に対する補正值delta_d の計算 p.40 */
/*=====*/
var delta_d;
delta_d = ( tk_deg - t1 ) / 360;

/* --- delta_d の修正 -----*/
/* step 1 */
if( delta_d > 1 ) delta_d -= 1;
if( delta_d < -1 ) delta_d += 1;
/* step 2 */
if( delta_d > 0.9 ) delta_d -= 1;
if( delta_d < -0.9 ) delta_d += 1;
/* ----- */

d += delta_d;

} while ( Math.abs( delta_d ) > 0.00005 ); /* 収束の判定 */

return d - Math.floor( d );          /* 時刻を返す */

}

function jikoku(hh,mm) {
/* 時刻の編集 hh,mm → hh:mm */
var hhmm;
if ( hh < 10 ){
    hhmm = "&nbsp;" + String(hh);
} else {
    hhmm = String(hh);
}

hhmm += ":";

if ( mm < 10 ){
    hhmm += "0" + String(mm);
} else {
    hhmm += String(mm);
}

return hhmm
}

function dsp_sun(y,m,d,lon,lat){
/* 日の出・日の入り、薄明時刻の表示 */
var h;
var hh;
var mm;

document.write("<font size=¥1¥ color=¥#ccffff¥>");

/* 薄明開始 */
h = sun(y,m,d,lon,lat,-18,0) * 24;
hh = Math.floor( h );
mm = Math.floor( ( h - hh ) * 60 );
document.write( " 薄明始" + jikoku(hh,mm) + " " );

/* 薄明終了 */
document.write("<font size=¥1¥ color=¥#ccffff¥>");
h = sun(y,m,d,lon,lat,-18,1) * 24;
hh = Math.floor( h );
mm = Math.floor( ( h - hh ) * 60 );
document.write( "薄明終" + jikoku(hh,mm) + "<br>" );

/* 日の出 */
document.write("<font size=¥1¥ color=¥#ffccff¥>");
h = sun(y,m,d,lon,lat,0,0) * 24;
hh = Math.floor( h );
mm = Math.floor( ( h - hh ) * 60 );
document.write( " 日出" + jikoku(hh,mm) );
```

```

/* 日の入り */
h = sun(y, m, d, lon, lat, 0, 1) * 24;
hh = Math.floor( h );
mm = Math.floor( ( h - hh ) * 60 );
document.write( " 日入" + jikoku(hh, mm) + "<br>" );

}

/* ##### 月 ##### */

function lambda_moon(t) {
/*-----*/
/* 月位置の略算式(p.124) */
/* lambda : 月の黄経 */
/*-----*/
var am = 0.0040 * Math.sin( rad( 119.5 + 1.33 * t ) )
+ 0.0020 * Math.sin( rad( 55.0 + 19.34 * t ) )
+ 0.0006 * Math.sin( rad( 71 + 0.2 * t ) )
+ 0.0006 * Math.sin( rad( 54 + 19.3 * t ) );

var lambda_w;
lambda_w = 218.3161 + 4812.67881 * t
+ 6.2887 * Math.sin( rad( 134.961 + 4771.9886 * t + am ) )
+ 1.2740 * Math.sin( rad( 100.738 + 4133.3536 * t ) )
+ 0.6583 * Math.sin( rad( 235.700 + 8905.3422 * t ) )
+ 0.2136 * Math.sin( rad( 269.926 + 9543.9773 * t ) )
+ 0.1856 * Math.sin( rad( 177.525 + 359.9905 * t ) )
+ 0.1143 * Math.sin( rad( 6.546 + 9664.0404 * t ) )
+ 0.0588 * Math.sin( rad( 214.22 + 638.635 * t ) )
+ 0.0572 * Math.sin( rad( 103.21 + 3773.363 * t ) )
+ 0.0533 * Math.sin( rad( 10.66 + 13677.331 * t ) )
+ 0.0459 * Math.sin( rad( 238.18 + 8545.352 * t ) )
+ 0.0410 * Math.sin( rad( 137.43 + 4411.998 * t ) )
+ 0.0348 * Math.sin( rad( 117.84 + 4452.671 * t ) )
+ 0.0305 * Math.sin( rad( 312.49 + 5131.979 * t ) )
+ 0.0153 * Math.sin( rad( 130.84 + 758.698 * t ) )
+ 0.0125 * Math.sin( rad( 141.51 + 14436.029 * t ) )
+ 0.0110 * Math.sin( rad( 231.59 + 4892.052 * t ) )
+ 0.0107 * Math.sin( rad( 336.44 + 13038.696 * t ) )
+ 0.0100 * Math.sin( rad( 44.89 + 14315.966 * t ) )
+ 0.0085 * Math.sin( rad( 201.5 + 8266.71 * t ) )
+ 0.0079 * Math.sin( rad( 278.2 + 4493.34 * t ) )
+ 0.0068 * Math.sin( rad( 53.2 + 9265.33 * t ) )
+ 0.0052 * Math.sin( rad( 197.2 + 319.32 * t ) )
+ 0.0050 * Math.sin( rad( 295.4 + 4812.66 * t ) )
+ 0.0048 * Math.sin( rad( 235.0 + 19.34 * t ) )
+ 0.0040 * Math.sin( rad( 13.2 + 13317.34 * t ) )
+ 0.0040 * Math.sin( rad( 145.6 + 18449.32 * t ) )
+ 0.0040 * Math.sin( rad( 119.5 + 1.33 * t ) )
+ 0.0039 * Math.sin( rad( 111.3 + 17810.68 * t ) )
+ 0.0037 * Math.sin( rad( 349.1 + 5410.62 * t ) )
+ 0.0027 * Math.sin( rad( 272.5 + 9183.99 * t ) )
+ 0.0026 * Math.sin( rad( 107.2 + 13797.39 * t ) )
+ 0.0024 * Math.sin( rad( 211.9 + 988.63 * t ) )
+ 0.0024 * Math.sin( rad( 252.8 + 9224.66 * t ) )
+ 0.0022 * Math.sin( rad( 240.6 + 8185.36 * t ) )
+ 0.0021 * Math.sin( rad( 87.5 + 9903.97 * t ) )
+ 0.0021 * Math.sin( rad( 175.1 + 719.98 * t ) )
+ 0.0021 * Math.sin( rad( 105.6 + 3413.37 * t ) )
+ 0.0020 * Math.sin( rad( 55.0 + 19.34 * t ) )
+ 0.0018 * Math.sin( rad( 4.1 + 4013.29 * t ) )
+ 0.0016 * Math.sin( rad( 242.2 + 18569.38 * t ) )
+ 0.0012 * Math.sin( rad( 339.0 + 12678.71 * t ) )
+ 0.0011 * Math.sin( rad( 276.5 + 19208.02 * t ) )
+ 0.0009 * Math.sin( rad( 218 + 8586.0 * t ) )
+ 0.0008 * Math.sin( rad( 188 + 14037.3 * t ) )
+ 0.0008 * Math.sin( rad( 204 + 7906.7 * t ) )
+ 0.0007 * Math.sin( rad( 140 + 4052.0 * t ) )
+ 0.0007 * Math.sin( rad( 275 + 4853.3 * t ) )
+ 0.0007 * Math.sin( rad( 216 + 278.6 * t ) )
+ 0.0006 * Math.sin( rad( 128 + 1118.7 * t ) )
+ 0.0005 * Math.sin( rad( 247 + 22582.7 * t ) )
+ 0.0005 * Math.sin( rad( 181 + 19088.0 * t ) )
+ 0.0005 * Math.sin( rad( 114 + 17450.7 * t ) )
+ 0.0005 * Math.sin( rad( 332 + 5091.3 * t ) )
+ 0.0004 * Math.sin( rad( 313 + 398.7 * t ) )
+ 0.0004 * Math.sin( rad( 278 + 120.1 * t ) )
+ 0.0004 * Math.sin( rad( 71 + 9584.7 * t ) )
+ 0.0004 * Math.sin( rad( 20 + 720.0 * t ) )
+ 0.0003 * Math.sin( rad( 83 + 3814.0 * t ) )
+ 0.0003 * Math.sin( rad( 66 + 3494.7 * t ) )

```

```

riseandset2.js

+ 0.0003 * Math.sin( rad( 147      + 18089.3   * t ) )
+ 0.0003 * Math.sin( rad( 311      + 5492.0    * t ) )
+ 0.0003 * Math.sin( rad( 161      + 40.7      * t ) )
+ 0.0003 * Math.sin( rad( 280      + 23221.3  * t ) );

while( lambda_w < 0 ){
  lambda_w += 360;
}

return lambda_w;
}

function beta(t){
  /*-----*/
  /* 月位置の略算式(p.124) */
  /* beta : 月の黄緯      */
  /*-----*/
  var bm = 0.0267 * Math.sin( rad( 234.95 + 19.341 * t ) )
    + 0.0043 * Math.sin( rad( 322.1  + 19.36  * t ) )
    + 0.0040 * Math.sin( rad( 119.5  + 1.33  * t ) )
    + 0.0026 * Math.sin( rad( 55.0   + 19.34  * t ) )
    + 0.0005 * Math.sin( rad( 307    + 19.4   * t ) );

  return 5.1282 * Math.sin( rad( 93.273 + 4832.0202 * t + bm ) )
    + 0.2806 * Math.sin( rad( 228.235 + 9604.0088 * t ) )
    + 0.2777 * Math.sin( rad( 138.311 + 60.0316 * t ) )
    + 0.1732 * Math.sin( rad( 142.427 + 4073.3220 * t ) )
    + 0.0554 * Math.sin( rad( 194.01  + 8965.374  * t ) )
    + 0.0463 * Math.sin( rad( 172.55  + 698.667  * t ) )
    + 0.0326 * Math.sin( rad( 328.96  + 13737.362 * t ) )
    + 0.0172 * Math.sin( rad( 3.18    + 14375.997 * t ) )
    + 0.0093 * Math.sin( rad( 277.4   + 8845.31  * t ) )
    + 0.0088 * Math.sin( rad( 176.7   + 4711.96  * t ) )
    + 0.0082 * Math.sin( rad( 144.9   + 3713.33  * t ) )
    + 0.0043 * Math.sin( rad( 307.6   + 5470.66  * t ) )
    + 0.0042 * Math.sin( rad( 103.9   + 18509.35 * t ) )
    + 0.0034 * Math.sin( rad( 319.9   + 4433.31  * t ) )
    + 0.0025 * Math.sin( rad( 196.5   + 8605.38  * t ) )
    + 0.0022 * Math.sin( rad( 331.4   + 13377.37 * t ) )
    + 0.0021 * Math.sin( rad( 170.1   + 1058.66  * t ) )
    + 0.0019 * Math.sin( rad( 230.7   + 9244.02  * t ) )
    + 0.0018 * Math.sin( rad( 243.3   + 8206.68  * t ) )
    + 0.0018 * Math.sin( rad( 270.8   + 5192.01  * t ) )
    + 0.0017 * Math.sin( rad( 99.8    + 14496.06 * t ) )
    + 0.0016 * Math.sin( rad( 135.7   + 420.02  * t ) )
    + 0.0015 * Math.sin( rad( 211.1   + 9284.69  * t ) )
    + 0.0015 * Math.sin( rad( 45.8    + 9964.00  * t ) )
    + 0.0014 * Math.sin( rad( 219.2   + 299.96  * t ) )
    + 0.0013 * Math.sin( rad( 95.8    + 4472.03  * t ) )
    + 0.0013 * Math.sin( rad( 155.4   + 379.35  * t ) )
    + 0.0012 * Math.sin( rad( 38.4    + 4812.68  * t ) )
    + 0.0012 * Math.sin( rad( 148.2   + 4851.36  * t ) )
    + 0.0011 * Math.sin( rad( 138.3   + 19147.99 * t ) )
    + 0.0010 * Math.sin( rad( 18.0    + 12978.66 * t ) )
    + 0.0008 * Math.sin( rad( 70      + 17870.7  * t ) )
    + 0.0008 * Math.sin( rad( 326     + 9724.1  * t ) )
    + 0.0007 * Math.sin( rad( 294     + 13098.7  * t ) )
    + 0.0006 * Math.sin( rad( 224     + 5590.7  * t ) )
    + 0.0006 * Math.sin( rad( 52      + 13617.3  * t ) )
    + 0.0005 * Math.sin( rad( 280     + 8485.3  * t ) )
    + 0.0005 * Math.sin( rad( 239     + 4193.4  * t ) )
    + 0.0004 * Math.sin( rad( 311     + 9483.9  * t ) )
    + 0.0004 * Math.sin( rad( 238     + 23281.3 * t ) )
    + 0.0004 * Math.sin( rad( 81      + 10242.6 * t ) )
    + 0.0004 * Math.sin( rad( 13      + 9325.4  * t ) )
    + 0.0004 * Math.sin( rad( 147     + 14097.4 * t ) )
    + 0.0003 * Math.sin( rad( 205     + 22642.7 * t ) )
    + 0.0003 * Math.sin( rad( 107     + 18149.4 * t ) )
    + 0.0003 * Math.sin( rad( 146     + 3353.3  * t ) )
    + 0.0003 * Math.sin( rad( 234     + 19268.0 * t ) );
}

function pi(t){
  /*-----*/
  /* 月位置の略算式(p.124) */
  /* pi : 月の視差      */
  /*-----*/
  return 0.9507 * Math.sin( rad( 90 ) )
    + 0.0518 * Math.sin( rad( 224.98 + 4771.989 * t ) )
}

```

riseandset2.js

```
+ 0.0095 * Math.sin( rad( 190.7 + 4133.35 * t ) )
+ 0.0078 * Math.sin( rad( 325.7 + 8905.34 * t ) )
+ 0.0028 * Math.sin( rad( 0.0 + 9543.98 * t ) )
+ 0.0009 * Math.sin( rad( 100.0 + 13677.3 * t ) )
+ 0.0005 * Math.sin( rad( 329 + 8545.4 * t ) )
+ 0.0004 * Math.sin( rad( 194 + 3773.4 * t ) )
+ 0.0003 * Math.sin( rad( 227 + 4412.0 * t ) );
}
```

```
var sw=0; /* check first */
```

```
function moon(y,m,d,lon,lat,rs){
  /******
  /* input: y=year,m=month,d=day(day+hour),lon=経度,lat=緯度, /*
  /* rs=0(出)1(没) /*
  /* output: hour (出沒時刻) /*
  /******
  do{
    var d_start;
    if ( sw == 0 ){
      sw = 1;
      d_start = Math.floor( d ); /* save initial day */
    }
    /*=====*/
    /* 時刻変数Tの計算 p.126 */
    /*=====*/
    var tt; /* 指定日時の経過日数 */
    tt = t(y,m,d);

    /*=====*/
    /* 月の黄経λm,黄緯βmの計算 p.127 */
    /*=====*/
    var lambda_m;
    lambda_m = lambda_moon( tt );
    lambda_m %= 360;
    var beta_m;
    beta_m = beta( tt );
    beta_m %= 360;

    /*=====*/
    /* 月の赤経α、赤緯δの計算 p.131 */
    /*=====*/
    var epsilon; /* 黄道傾角 */
    epsilon = 23.439291 - 0.000130042 * tt;

    /* 座標変換(p.131) */
    var u;
    var v;
    var w;

    u = Math.cos( rad( beta_m ) ) * Math.cos( rad( lambda_m ) );
    v = -1 * Math.sin( rad( beta_m ) ) * Math.sin( rad( epsilon ) )
      + Math.cos( rad( beta_m ) ) * Math.sin( rad( lambda_m ) ) * Math.cos( rad( epsilon ) );
    w = Math.sin( rad( beta_m ) ) * Math.cos( rad( epsilon ) )
      + Math.cos( rad( beta_m ) ) * Math.sin( rad( lambda_m ) ) * Math.sin( rad( epsilon ) );

    var tan_alpha;
    var tan_delta;
    tan_alpha = v / u;
    tan_delta = w / ( Math.sqrt( Math.pow(u,2) + Math.pow(v,2) ) );

    var alpha; /* 赤経 radian */
    var delta; /* 赤緯 radian */
    alpha = Math.atan(tan_alpha);
    delta = Math.atan(tan_delta);

    var alpha_deg; /* 赤経 degree */
    var delta_deg; /* 赤緯 degree */
    alpha_deg = 180 * alpha / Math.PI;

    if ( u < 0 ) alpha_deg += 180; /* p.131 */

    if ( alpha_deg < 0 ) alpha_deg += 360;

    delta_deg = 180 * delta / Math.PI;

    /*=====*/
    /* 恒星時θの計算 p.76 */
    /*=====*/
    var theta;
    var hh= d - Math.floor( d );
```

riseandset2.js

```
theta = 325.4606 + 360.007700536 * tt
      + 0.00000003879 * Math.pow(tt,2) + 360 * hh + lon;
theta %= 360;

/*=====*/
/* 出没高度 k の計算 */
/*=====*/
var k; /* 出没高度 */
k = -0.585556 + pi( tt ); /* p.118 (5.1) pi : 月の視差 p.131 */

/*-----*/
/* 出没高度k に対応する時角tk への換算 */
/*-----*/
cos_tk = ( Math.sin( rad( k ) ) - Math.sin( delta ) * Math.sin( rad( lat ) ) )
          / ( Math.cos( delta ) * Math.cos( rad( lat ) ) );
var tk;
tk = Math.acos( cos_tk );
if ( rs == 0 ) /* 「出」の時 */
  tk_deg = 180 * tk / Math.PI * -1; /* 時角をマイナス値にする */
else
  tk_deg = 180 * tk / Math.PI;

/*-----*/
/* 恒星時theta から月の時角t を計算する */
/* t1 = theta - alpha */
/*-----*/
var t1;
t1 = theta - alpha_deg;

/*-----*/
/* 仮定時刻d に対する補正值delta_d の計算 p.121 */
/*-----*/
var delta_d;
var tk_w;

tk_w = tk_deg - t1;
if ( tk_w > 180 ) tk_w -= 360; /* 値の調整 p.122 */
if ( tk_w < -180 ) tk_w += 360; /* 値の調整 p.122 */

delta_d = tk_w / 347.8;

/* --- delta_d の修正 -----*/
/* step 1 */
if( delta_d > 1 ) delta_d -= 1;
if( delta_d < -1 ) delta_d += 1;
/* step 2 */
if( delta_d > 0.9 ) delta_d -= 1;
if( delta_d < -0.9 ) delta_d += 1;
/* ----- */

d += delta_d;

if ( d_start != Math.floor( d ) ){ /* 日付が変わったら */
  return -1; /* 計算終了(この日は月の出、入り無し) */
}

} while ( Math.abs( delta_d ) > 0.00005 ); /* 収束の判定 */

return d - Math.floor( d ); /* 時刻を返す */
}

function dsp_moon(y, m, d, lon, lat){
  /* 月の出、月の入りの表示 */
  var h;
  var hh;
  var mm;

  /* 月の出 */
  document.write("<font size=¥1¥" color=¥"#ff9900¥">");
  sw = 0;
  h = moon(y, m, d, lon, lat, 0) * 24;
  if ( h < 0 ){
    document.write( " 月出 — " );
  }else{
    hh = Math.floor( h );
    mm = Math.floor( ( h - hh ) * 60 );
    document.write( " 月出" + jikoku(hh, mm) );
  }

  /* 月の入り */
}
```

riseandset2.js

```
sw = 0;
h = moon(y, m, d, lon, lat, l) * 24;
if ( h < 0 ){
  document.write( " 月入  —<br>" );
}else{
  hh = Math.floor( h );
  mm = Math.floor( ( h - hh ) * 60 );
  document.write( " 月入" + jikoku(hh, mm) + "<br>" );
}
}

/* ##### end of function ##### */

/* % start main program % */

var lon=133.064;      /* 経度(松江) */
var lat=35.489;      /* 緯度(松江) */

var now = new Date(); /* get today */

/* 現在の年月日のセット */
year = now.getFullYear();
month = now.getMonth() + 1;
day = now.getDate();

document.write( "<font size=14 color=#ffff00>" );
document.write( "松江<br>" );

riseandset(year, month, day, lon, lat);

/* % end main program % */

/-->
```