

1.Math Object マセマティック・オブジェクト【PR58】

数学的な処理を行うもの。¹

ただし、準備されているものには、三角関数、指数対数、平方根など初等的な関数しかない。

文法：

Math.プロパティ/メソッド（数値）

プロパティ：

E、LN2、LN10、LOG2E、LOG10E、PI、SQRT1_2、SQRT2

メソッド：

abs,acos,asin,atan,ceil,cos,exp,floor,log,max,min,pow,random,round,sin,sqrt,tan

Math オブジェクトは、普通のプログラミング言語のように「型」をもたないため、その精度には問題がある。したがって、科学計算、技術計算、経済などの計算での使用には十分注意が必要となる。

こうした問題を超えるために、「数値オブジェクト (Number)」が作られた。²

■プログラム 1：「sin の値を 0° から 180° まで 10° ごとに求める」

三角関数 sin を表示するプログラム。【PR61】

計算精度は、ブラウザや OS によって違いがあるので、注意が必要。

```
<HTML>
<HEAD>
<TITLE>sin</TITLE>
</HEAD>
<BODY>
<SCRIPT type="text/Javascript">
  for (x=0; x<=180; x=x+10){
    s=Math.sin(x*3.1415927/180);
    document.write(x, ", ",s,"<BR>");
  }
</SCRIPT>
</BODY>
</HTML>
```

¹ 開始の文字「M」は**大文字**でなければならない。

² Number オブジェクトの使用は、IE4.0、NN3.0 以降。また、形式指定に関するメソッドは、IE5.5、NN6.0 以降。

◆ラジアン表記（弧度法）

JavaScript では、角度に関する表記は「度数」を使わず、「ラジアン表記」を使う。
ラジアンとは、円の中心角とその弧に関する比。

180 度 ; π rad

360 度 ; 2π rad

となる。

したがって、角度を扱う \sin 、 \cos などの三角関数では、日常的に使う度数表記をラジアン表記に変換する必要がある。

$(\text{角度} \times \pi) \div 180$

■プログラム 2 : 「sin カーブを図で示す」

スペースを使って、右に送り、sin カーブを描く。

```
<HTML>
<HEAD>
<TITLE>sin カーブ</TITLE>
<SCRIPT type="text/Javascript">
  // スペースを入れる関数
  function spc(n){
    s="";
    for (i=0; i<n; i++)
      s=s+" ";
    return s;
  }
</SCRIPT>
</HEAD>
<BODY>
<PRE>  // <PRE>は、改行や空白をソースのままに表示するタグ
<SCRIPT LANGUAGE="Javascript">
  document.write("¥t-1.....+.....0.....+.....1<BR><BR>"); //目盛り
  for (x=0; x<=360; x=x+20){
    ts=10*Math.sin(x*3.1415927/180)+10;
    document.write(x,"¥t",spc(ts),"*<BR>");
  }
</SCRIPT>
</PRE>
```

```
</BODY>
```

```
</HTML>
```

(解説)

「`\t`」は、「エスケープシーケンス」と呼ばれる類で、「タブ」を意味している。³

1-2. 繰り上げ、繰り下げ、四捨五入に関するもの

日常的に、使われる `Math` オブジェクトの中で最も使われるのが、これらのメソッドもしくはプロパティです。

>> 繰り上げ【PR63】

繰り上げられます。

つまり、最も近い整数に上げられます。

`ceil()`

```
num=Math.ceil(23.4);
```

この結果は、「24」になります。

パラメータが、「負の数」の場合、繰り上げられるので、「正」の方向に数が切り上げられます。

```
num=Math.ceil(-23.4);
```

この結果は、「-23」になります。

>> 繰り下げ【PR63】

繰り下げられます。最も近い整数に下げられます。

`floor()`

```
num=Math.floor(23.4);
```

この結果は、「23」になります。

パラメータが、「負の数」の場合、繰り下げられるので、「負」の方向に数が切り下げられます。

```
num=Math.floor(-23.4);
```

この結果は、「-24」になります。

³ エスケープシーケンスについては、【PR22】の「特殊文字」の項目を参照

>> 四捨五入 【PR63】

与えられたパラメータ（値）の小数点以下第 1 位の位を四捨五入します。

round()

```
num=Math.round(12.45);
```

結果は、「12」になります。

（小数点以下、第 2 位やそれ以下は影響しないことに注意！）

1-3. 数の大小の比較

2つの数のうちで、大小を比較します。

この比較では、数値の他に変数でも比較することができる。

>> 大きい値を返す 【PR68】

与えられた 2つの値のうち、「大きい」値の方を返す。

max(num1,num2)

```
num=Math.max(12,45);
```

この結果は、「45」が与えられる。

>> 小さい値を返す 【PR68】

与えられた 2つの値のうち、「小さい」値の方を返す。

min(num1,num2)

```
num=Math.min(12,45);
```

この結果は、「12」が与えられる。

1-4. 乱数 【PR70】

「乱数」とは、できるだけたらしめな数である。
コンピュータや、数学の世界では、さまざまところで、この「乱数」が使われている。
特に、コンピュータでは、何かをコンピュータが決定する時に使われる。したがって、ゲームなどで広く応用されている。

乱数を発生させるメソッドは、

Math.random()

このメソッドは、「0」から「1」までの間の数を返します。

() 内は数値を入れ必要がない。

したがって、得られる結果は、すべて小数点以下の並んだ数字です。

```
num=Math.random();
```

例 : 0.2368452

0.89511230

よって、実際に使う場合は、「10 倍」もしくは、「100 倍」などをして、不必要なくらいを切り捨てて使う。