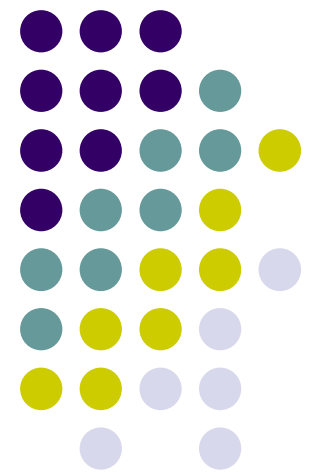


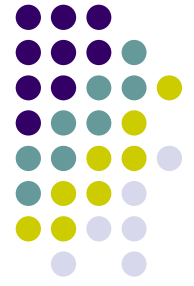
# 第5回 JavaScriptゼミ

---

## 3 - 4 Mathオブジェクト

発表者: 鈴木朋央 発表日: 06/11/10





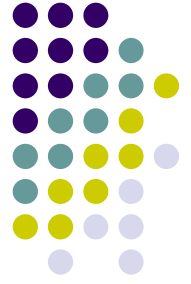
# Mathオブジェクトとは

- 高度な演算 (三角関数や対数計算) を行うことができるメソッドを持つ
- DateやString同様、JavaScriptの組み込みオブジェクトである



しかし…

- インスタンスを生成することなしにメソッドやプロパティにアクセス可能



# クラスメソッド

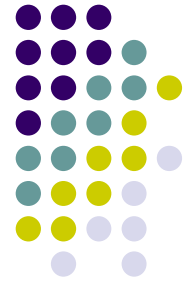
new演算子でインスタンスを生成する必要はない

Math.メソッド名()

使用例: 小数部を四捨五入した値を戻すround()メソッド

```
var myNum = 4.56;
```

```
var myResult = Math.round(myNum);
```



# クラス変数

インスタンスに直接関連付けられていないで定数のように使用できる変数のこと

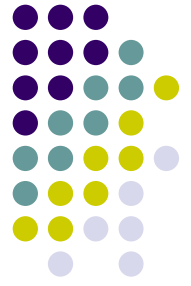
これにアクセスする場合もインスタンスを生成する必要はない

Math.変数名

使用例: 円周率を表す「PI」を使用して外周を求める

```
var radius = 5;    半径
```

```
var area = 2 * Math.PI * radius;    外周
```

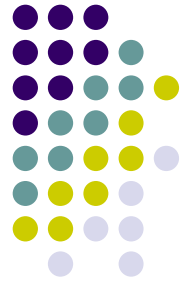


# Mathオブジェクトのプロパティ

次のようなプロパティがクラス変数として用意されている

プロパティ名	値
E	自然対数の底
LN2	2の自然対数値
LN10	10の自然対数値
LG2E	2を底とするeの対数値
LOG10E	10を底とするeの対数値
PI	円周率
SQRT1_2	$\frac{1}{2}$ の平方根
SQRT2	2の平方根

# 一般的な算術演算のためのメソッド



メソッド: `abs()`

構文: `Math.abs(number)`

説明: 引数として渡された数値の絶対値を戻す

```
var num = Math.abs(-1)    * numは1
```

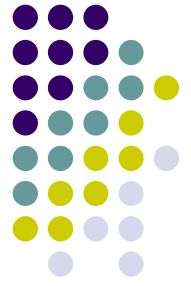
メソッド: `exp()`

構文: `Math.exp(number)`

説明:  $e$ の`number`乗の数値を戻す

```
var num = Math.exp(2);    * numは7.839
```

# 一般的な算術演算のためのメソッド



メソッド: `ceil()`

構文: `Math.ceil(number)`

説明: 引数を下回らない最小の整数を戻す。負の値も適応可能。

```
var num = Math.ceil(11.1)    * numは12
```

```
var num = Math.ceil(-14.5)  * numは-14
```

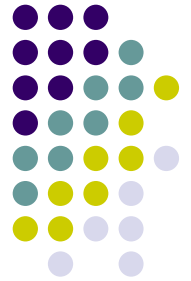
メソッド: `floor()`

構文: `Math.floor(number)`

説明: 引数の値を超えない最大の整数を戻す

```
var num = Math.floor(33.3); * numは33
```

# 一般的な算術演算のためのメソッド



メソッド: `log()`

構文: `Math.log(number)`

説明: `number`の自然対数を戻す

```
var num = Math.log(Math.E)    * numは1
```

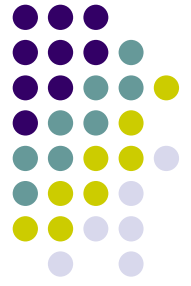
メソッド: `max()`

構文: `Math.max(number1, number2, number3, ...)`

説明: 引数の中で一番大きい数値を戻す

```
var num = Math.max(10, 2, 6); * numは10
```

# 一般的な算術演算のためのメソッド



メソッド: `min()`

構文: `Math.min(number1, number2, number3, ...)`

説明: 引数の中で一番小さい数値を戻す

```
var num = Math.min(10, 2, 101) * numは2
```

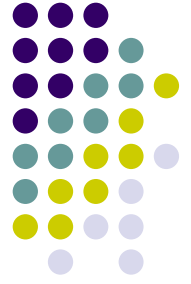
メソッド: `pow()`

構文: `Math.pow(base, exponent)`

説明: 引数baseの引数exponent乗の値を戻す

```
var num = Math.pow(2, 4); * numは16
```

# 一般的な算術演算のためのメソッド



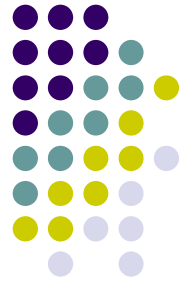
メソッド: `round()`

構文: `Math.round(number)`

説明: 引数の値を小数部分以下第1位を四捨五入した値を戻す

```
var num = Math.round(10.55)    * numは11
```

# 一般的な算術演算のためのメソッド



メソッド: `random()`

構文: `Math.random()`

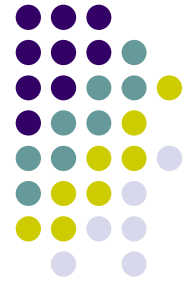
説明: 0 から 1 の間の乱数を戻す

```
var num = Math.ceil( Math.random() * 10); * num=1 ~ 10
```

\* 補足

`random()`メソッドは1以下の乱数を生成するため、ランダムな整数を使用したい場合は、結果に整数値をかければよい

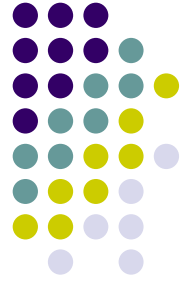
かつ小数点以下を切りたい場合は、`ceil()`メソッドや`floor`メソッドを併用することにより実現可能



# random()メソッドの使用例

例: 4つのイメージをランダムに選択して表示

```
<script language="JavaScript" type="text/javascript">  
var imgArray = new Array("img1.jpg", "img2.jpg", "img3.jpg",  
"img4.jpg");  
var imageFile = imgArray[Math.floor(Math.random() * 4)];  
document.write("<img src= \"", "images/" + imageFile, "\">");  
</script>
```



# 三角関数関連のメソッド

Mathオブジェクトに用意されている三角関数関連のメソッド

メソッド	説明
<code>sin()</code>	サインを戻す
<code>cos()</code>	コサインを戻す
<code>tan()</code>	タンジェントを戻す
<code>asin()</code>	アークサイン(サインの逆関数)を戻す
<code>acos()</code>	アークコサイン(コサインの逆関数を)戻す
<code>atan()</code>	アークタンジェント(タンジェントの逆関数)を戻す



# 使用法

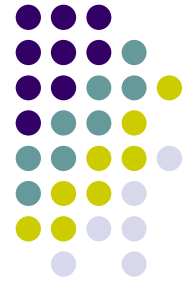
三角関数のメソッドの引数にはラジアン値で角度を渡す

$$\text{ラジアン値} = \text{円周率} * \text{角度} / 180$$

使用例: 90度の角度のサイン値

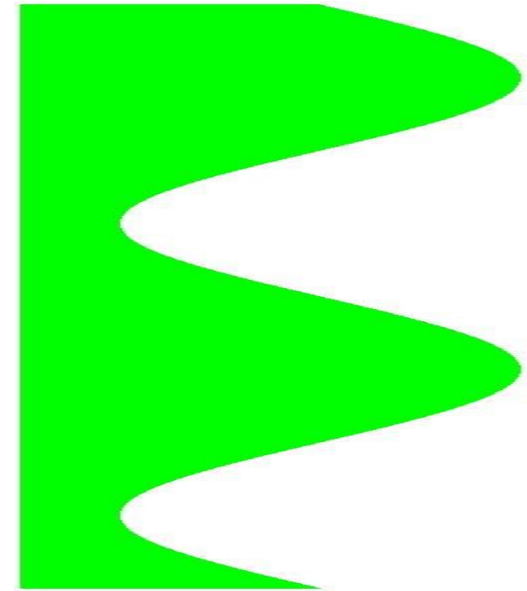
```
var degrees = 90;
```

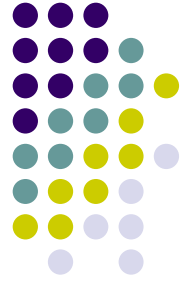
```
var radians = Math.sin( Math.PI * degrees / 180);
```



# 使用例 ~ サインカーブの描画

```
<script language="JavaScript" type="text/javascript">
var sinValue;
var offset = 150;
for(var degrees = 0; degrees <= 720; degrees++){
    // ラジアン値に変換
    sinValue = Math.sin(Math.PI * degrees / 180);
    // イメージファイルを横に引き伸ばして描画
    document.write("<img src='images/greenDot.gif' height=1 “,”width=“,
                    offset + Math.round(100 * sinValue), “>”);
    document.write("<br>");
}
</script>
```





# 練習問題

じゃんけんゲームを作ってください。

イメージを使おうが、テキストだけだろうが何でもOKです。  
ビジュアルは任せます。

ただし、以下の2点だけは実現してください。

相手が勝ったとき・・・

気を良くしたのか、同じ手を出す確率UP！

相手が負けたとき・・・

よほど悔しいのか、同じ手を出す確率DOWN！