Propiedades y métodos del objeto Math

El objeto Math nos permite realizar tareas matemáticas, tiene una gran variedad de propiedades y métodos, a continuación se mostrarán unos cuantos:

Constante PI: Retorna una aproximación de la constante pi (flotante).

```
>> Math.PI
<- 3.141592653589793
```

hypot(): Retorna un número entero correspondiente a la longitud de la hipotenusa de un triángulo rectángulo, recibiendo como argumento dos números enteros que corresponden la longitud de los catetos del dicho triángulo.

```
>> Math.hypot(3,4)
<- 5
```

log2(): Retorna un número que al ser exponente de 2, de como resultado el número que recibe como argumento.

```
>> Math.log2(64)
<- 6
```

Está también log10, funciona igual sólo que con base 10 en vez de 2.

floor(): Retorna el menor entero más próximo al argumento, su argumento debe un valor numérico.

```
>> Math.floor(0.99)
<- 0
>> Math.floor(1.29)
<- 1
```

ceil(): Retorna el mayor entero más próximo al argumento, su argumento debe un valor numérico.

```
>> Math.ceil(0.2)
<- 1
>> Math.ceil(0.99)
<- 1
```

random(): Retorna un número aleatorio entre 0 y 1.

```
>> Math.random()
<- 0.2215748118788119
>> Math.random()
<- 0.8051673060752702
```

Algoritmo para generar un número entero aleatorio dado un rango

Utilizando métodos del objeto Math, podemos crear una función que nos retorne un número entero aleatorio, especificando en los parámetros el rango.

```
>> function aleatorio( num_min, num_max){
    return Math.floor(Math.random()*((num_max+1)-num_min)+num_min)
   }
```

```
>> aleatorio(2,5)
<- 5
>> aleatorio(2,5)
<- 3
>> aleatorio(2,5)
<- 2
```

Constante de Euler: Retorna una aproximación del número de Euler.

```
» Math.E
← 2.718281828459045
```

Formas de recorrer un arreglo:

```
arreglo = [18, 1, "hola"];

» for(var i=0; i<arreglo.length; i++){
    console.log(arreglo[i]);
  }
  18
  1
  hola

» for(var i in arreglo){
    console.log(arreglo[i]);
  }
  18
  1
  hola

» arreglo.forEach( x =>{
    console.log(x)})
  18
  1
  hola
```

Por cierto, en un mismo bloque de código, si creamos una variable con let no podremos declararla otra vez con var, se produciría un error, pero si podremos hacerlo si cuando creamos esa variable con let, está dentro de otro ámbito. A continuación un par de ejemplos: