

```
//MATH.RANDOM:
//A função built-
in Math.random() tem a missão de trazer um número aleatório dentro de um intervalo de números, por padrão a função random traz um número aleatório entre 0 e 1, porém com casas decimais, isso significa que ela pode trazer: 0.234235 ou 0.80342355 por exemplo, qualquer número real que esteja entre 0 e 1.
//Porém, essa função aceita mais parâmetros onde podemos gerar números aleatórios dentro de um intervalo pré-determinado por nós, vejamos alguns exemplos:

//USANDO RANDOM SEM PASSAR PARÂMETROS:
console.log('\n1)', Math.random()); //Note que vamos gerar valores aleatórios entre 0 e 1 com ponto flutuante...
console.log('1)', Math.random());
console.log('1)', Math.random());
console.log('1)', Math.random());

//GERANDO NÚMEROS REAIS ALEATÓRIOS DE 0 ATÉ UM VALOR MÁXIMO:
console.log('\n2)', Math.random() * 10); //Note que só temos que escolher um número e fazer ele multiplicado por random, gerando valores reais de 0 á 9, visto que o último número do range será desconsiderado...
console.log('2)', Math.random() * 10);
console.log('2)', Math.random() * 10);
console.log('2)', Math.random() * 10);

//USANDO FLOOR PARA GERAR NÚMEROS ALEATÓRIOS INTEIROS:
console.log('\n3)', Math.floor(Math.random())); //Geralmente usamos a função Math.floor para arredondar os números para baixo retirando o ponto flutuante, como random nunca chega ao número 1 se usarmos Math.floor numa função Math.random sem parâmetros, acabaremos por sempre gerando o valor "0", não importa quantas vezes tentemos...
console.log('3)', Math.floor(Math.random()));
console.log('3)', Math.floor(Math.random()));
console.log('3)', Math.floor(Math.random()));

//USANDO RANDOM E FLOOR PARA GERAR NÚMEROS INTEIROS DE 0 ATÉ UM DETERMINADO VALOR MÁXIMO:
```

```
console.log('\n4)', Math.floor(Math.random() * 10)); //Perceba que agora geramos números aleatórios entre 0 e 9, visto que o último número do range será desconsiderado...
console.log('4)', Math.floor(Math.random() * 10));
console.log('4)', Math.floor(Math.random() * 10));
console.log('4)', Math.floor(Math.random() * 10));
```

//GERANDO NÚMEROS RANDÔMICOS ENTRE UM VALOR MÍNIMO PRÉ-DETERMINADO E UM VALOR MÁXIMO PRÉ-DETERMINADO:

```
console.log('\n5)', Math.floor(Math.random() * ((10 + 1) - 5) + 5)); //Quando queremos trabalhar com valor mínimo e máximo temos que fazer o valor máximo + 1 subtraído pelo valor mínimo, pois a função random sempre desconsidera o último número do range, por isso temos que acrescentar 1 dígito, esse valor será multiplicado pelos valores randômicos gerados na função random e depois serão somados ao número mínimo no final do campo de parâmetros. Escolhemos para testar valores randômicos de 5 á 10, então tivemos que fazer 10 + 1, e depois subtraí-
```

lo pelo valor mínimo "5", que vai dar 6. O "6" será multiplicado pelos valores randômicos de 0 a 1, gerando valores reais de 1 á 5, mas como queremos valores de 5 á 10 teremos que multiplicar os valores randômicos por 5, assim o resultado sempre será de 5 á 10...

```
console.log('5)', Math.floor(Math.random() * (11 - 5) + 5)); //Aqui já colocamos o valor 11 direto...
console.log('5)', Math.floor(Math.random() * (11 - 5) + 5));
console.log('5)', Math.floor(Math.random() * (11 - 5) + 5));
```

//GERANDO NÚMEROS BINÁRIOS RANDÔMICOS:

```
console.log('\n6)', Math.floor(Math.random() * 2)); //Basta usar o número 2, que o range será sempre multiplicado por 1...
console.log('6)', Math.floor(Math.random() * 2));
console.log('6)', Math.floor(Math.random() * 2));
console.log('6)', Math.floor(Math.random() * 2));
```

//GERANDO NÚMEROS DE 1 BYTE RANDÔMICOS:

```
console.log('\n7)', Math.floor(Math.random() * 256)); //Basta usar o número 256, o range já irá desconsiderar o último valor...
console.log('7)', Math.floor(Math.random() * 256));
console.log('7)', Math.floor(Math.random() * 256));
console.log('7)', Math.floor(Math.random() * 256));
```

RESULTADO NO CONSOLE...

```
[Running] node "c:\Users\User\Documents\JAVASCRIPT\ARQUIVOS_DAS_AULAS\045-Math.random__Gerando_Numeros_Aleatorios.js"
```

```
1) 0.8051250727288284  
1) 0.34958922647481416  
1) 0.262002328616054  
1) 0.08801293973409496
```

```
2) 4.934202636549198  
2) 2.5896336630755745  
2) 6.659577593228305  
2) 6.851903532507304
```

```
3) 0  
3) 0  
3) 0  
3) 0
```

```
4) 0  
4) 7  
4) 8  
4) 7
```

```
5) 6  
5) 7  
5) 6  
5) 6
```

```
6) 0  
6) 1  
6) 0  
6) 1
```

```
7) 94  
7) 205  
7) 200
```

7) 204

[Done] exited with code=0 in 0.127 seconds