



Operadores e aritmética

A aritmética na computação é bastante parecida com a matemática tradicional, mas existem algumas diferenças.

Alguns cuidados



Como vimos no tópico passado, o primeiro cuidado que devemos tomar antes de fazer contas na programação é que nossos números sejam de fato números e não suas representações textuais, vamos olhar outros pontos de atenção ao trabalharmos com números:

- Saber se os números são inteiros ou de ponto-flutuante (*float*/decimal)
- Afinal, a divisão de 10 por 2 em inteiro ou *float* será 5, no entanto, a divisão de 11 por 2 em inteiro é 5 e em *float* 5.5
- O separador decimal, diferentemente do usado no Brasil, que é a **vírgula**, na programação usamos o padrão dos EUA - o **ponto**.
- Isso inclui as entradas do usuário, ele também deve usar ponto, ou teremos que normalizar o valor transformando pontos em vírgulas.
- Temos que respeitar a precedência das operações da mesma forma que faríamos na matemática. Por exemplo:
- $n1 + n2 / 2.0$ quer dizer: divida $n2$ por 2 e some $n1$, porque divisão precede soma na matemática, então ela será aplicada primeiro;
- Para ajustar a precedência usamos parênteses nas operações que queremos fazer primeiro: $(n1 + n2) / 2.0$, significa que a soma deve ser feita antes;
- Você pode usar quantos parênteses precisar, só cuidado que, diferentemente da matemática do colégio não usamos colchetes ou chaves, esses dois estão reservados na programação para outros usos

Operações básicas

No javascript temos disponíveis as 4 operações básicas com os seguintes operadores:

- `+` : Soma
- `-` : Subtração
- `*` : Multiplicação
- `/` : Divisão

Ainda temos o operador de resto, que indica o resto de uma divisão:

- `%`: Resto da divisão

Operações não tão básicas

Se precisar de mais operações, potência, raiz quadrada, funções trigonométricas e constantes matemáticas como PI ou EULER você pode recorrer à classe Math.

Não vamos descrever todos os métodos da classe, mas alguns dos mais úteis:

Valor absoluto:

- `Math.abs(num)`: Retorna o módulo, ou valor absoluto, de um número.

Arredondamentos:

- `Math.ceil(num)`: Retorna o menor inteiro que é maior ou igual a um número;
- `Math.floor(num)`: Retorna o maior inteiro que é menor ou igual a um número;
- `Math.round(num)`: Retorna o valor arredondado, para o valor inteiro mais próximo.

Valores máximos e mínimos:

- `Math.max(num1, ., numN)`: Retorna o maior dentre os parâmetros recebidos;
- `Math.min(num1, ., numN)`: Retorna o menor dentre os parâmetros recebidos.

Potência e Raiz:

- `Math.pow(base, exp)`: Retorna a base elevada à potência do expoente;
- `Math.sqrt(num)`: Retorna a raiz quadrada positiva de um número.

Geração de números pseudoaleatório:

- `Math.random()`: Retorna um número pseudoaleatório entre 0 e 1.

Funções trigonométricas e PI:

- `Math.PI`: Contém o valor aproximado do PI;
- `Math.sin(num)`: Retorna o seno de um número;
- `Math.cos(num)`: Retorna o cosseno de um número;
- `Math.tan(num)`: Retorna a tangente de um número.

Consulte a [documentação no MDN](#) para ter acesso completo a todos os métodos existentes dentro dessa classe.

Referências e Materiais Complementares

- [Expressões e operadores](#)
- [Math](#)

Próximo Tópico