

Embora os operadores padrão +, -, *, /, %e ** trabalhem com BigInt, é importante entender que você não pode misturar operandos de Digite BIGINT com operando de números regulares. Isso pode parecer confuso em Primeiro, mas há uma boa razão para isso. Se um tipo numérico fosse mais Geral que o outro, seria fácil definir aritmética em misto operando para simplesmente retornar um valor do tipo mais geral. Mas também não Tipo é mais geral que o outro: BigInt pode representar extraordinariamente valores grandes, tornando -o mais geral que os números regulares. Mas BigInt só pode representar números inteiros, fazendo o tipo de número JavaScript comum mais geral. Não há como contornar esse problema, então JavaScript contaminam -o simplesmente não permitindo operandos mistos para a aritmética operadores.

Operadores de comparação, por outro lado, trabalham com tipos numéricos mistos (mas consulte §3.9.1 para obter mais informações sobre a diferença entre == e ===):

```
1 < 2n // => true
```

```
2 > 1n // => true
```

```
0 == 0n // => true
```

```
0 === 0n // => false: o === verifica o tipo de igualdade como bem
```

Os operadores bitwise (descritos em §4.8.3) geralmente funcionam com BigInt operando. Nenhuma das funções do objeto de matemática aceita bigint operando, no entanto.

3.2.6 Datas e horários

JavaScript define uma classe de data simples para representar e manipular os números que representam datas e horários. JavaScript Datas são objetos, mas também têm uma representação numérica como um