Tipo de dato BigInt

En la introducción de este módulo dijimos que en JavaScript (y en otros lenguajes de programación) existe una problemática en relación con los rangos numéricos y consiste en el hecho de que ciertos valores muy grandes o muy pequeños, no pueden ser representados debido a que la información pierde precisión.

Es por eso que la especificación ES2020 agrega un nuevo tipo de dato, denominado como BigInt, que es un tipo numérico especial que provee soporte a enteros de tamaño arbitrario, de una forma relativamente sencilla. Es decir, que a diferencia del tipo de dato *Number*, que tiene un límite de precisión de 53 bits, el tipo de dato BigInt puede manejar números enteros de cualquier tamaño.

El tipo BigInt se puede crear de dos formas: o bien, podemos agregar una letra “n” al final del número entero, o llamando a la función BigInt().

```
const bigint = 1234567890123456789012345678901234567890n;
const bigintFuncion = BigInt("1234567890123456789012345678901234567890");
```

Realizar operaciones con BigInt

Al igual que el tipo *Number*, podemos realizar operaciones matemáticas con el tipo BigInt. Por ejemplo:

```
const num1 = 12345678901234567890123456789012345678901234567890n;

const num2 = 98765432109876543210987654321098765432109876543210n;

const suma = num1 + num2;

console.log("Suma:", suma);
```

Mezclando tipos de dato

En el ejemplo anterior ambos operandos son del tipo BigInt, pero ¿qué pasa si deseamos mezclar BigInt con el tipo Number? La respuesta es que en JavaScript no se pueden mezclar directamente valores de tipo BigInt y *Number* en operaciones matemáticas. Esto se debe a que son tipos de datos diferentes y tienen reglas de operación distintas, por tanto, si intentamos realizar una operación entre BigInt y *Number*, JavaScript lanzará un error. Por ejemplo:

```
const num1 = 12345678901234567890123456789012345678901234567890n;
const num2 = 50;
const suma = num1 + num2; // Uncaught TypeError: can't convert BigInt to number

console.log("Suma:", suma);
```

Para solucionar esto, podemos convertir uno de los tipos al otro antes de realizar la operación. Por ejemplo, podemos utilizar la función `BigInt()` para convertir un valor a BigInt o la función `Number()` para convertir un valor a *Number*.

```
const num1 = 12345678901234567890123456789012345678901234567890n;
const num2 = 50;
const suma = BigInt(num2) + num1; // Convertir num2 a BigInt
console.log("Suma:", suma); // Resultado: 12345678901234567890123456789012345678901234567900n;
```

Convertir BigInt a String

También podemos convertir el tipo de dato BigInt a una cadena de texto, para esto podemos utilizar el método "toString()". Por ejemplo:

```
const miNumeroBigInt = 12345678901234567890123456789012345678901234567890n;
const miNumeroString = miNumeroBigInt.toString();
console.log("Mi número BigInt como cadena de texto:", miNumeroString);
```