Actividad Modulo 3 – Number y operaciones

1. **Número a un nuevo número:** Declara una variable **num** con un valor numérico y luego crea una nueva variable llamada **newNum** utilizando la función **Number()** para convertir **num** a un nuevo número. Muestra en consola **newNum**.
2. **Rangos numéricos - Number.MAX_VALUE:** Muestra en consola el valor máximo que puede ser representado en JavaScript utilizando **Number.MAX_VALUE**.
3. **Rangos numéricos - Number.MIN_VALUE:** Muestra en consola el valor mínimo positivo que puede ser representado en JavaScript utilizando **Number.MIN_VALUE**.
4. **Rangos numéricos - Number.MAX_SAFE_INTEGER:** Muestra en consola el valor máximo seguro que puede ser representado con precisión en JavaScript utilizando **Number.MAX_SAFE_INTEGER**.
5. **Rangos numéricos - Number.MIN_SAFE_INTEGER:** Muestra en consola el valor mínimo seguro que puede ser representado con precisión en JavaScript utilizando **Number.MIN_SAFE_INTEGER**.
6. **Comprobación numérica - Number.isFinite(number):** Declara una variable **num** y utiliza el método **Number.isFinite()** para comprobar si **num** es un número finito. Muestra el resultado en consola.
7. **Comprobación numérica - Number.isInteger(number):** Declara una variable **num** y utiliza el método **Number.isInteger()** para comprobar si **num** es un número entero. Muestra el resultado en consola.
8. **Comprobación numérica - Number.isSafeInteger(number):** Declara una variable **num** y utiliza el método **Number.isSafeInteger()** para comprobar si **num** es un número seguro para representar con precisión en JavaScript. Muestra el resultado en consola.
9. **Representación numérica - .toExponential(digits):** Declara una variable **num** y utiliza el método **.toExponential()** para mostrar su representación en notación exponencial con un número específico de dígitos decimales. Muestra el resultado en consola.
10. **Representación numérica - .toFixed(digits):** Declara una variable **num** y utiliza el método **.toFixed()** para mostrar su representación con un número específico de dígitos decimales. Muestra el resultado en consola.
11. **Representación numérica - .toPrecision(size):** Declara una variable **num** y utiliza el método **.toPrecision()** para mostrar su representación con una longitud total específica. Muestra el resultado en consola.

12. **Convertir - Number.parseInt(text):** Declara una variable **text** con un valor numérico en formato de cadena y utiliza **Number.parseInt()** para convertirlo a un número entero. Muestra el resultado en consola.
13. **Convertir - Number.parseInt(text, radix):** Declara una variable **text** con un valor numérico en formato de cadena y utiliza **Number.parseInt()** con una base específica para convertirlo a un número entero. Muestra el resultado en consola.
14. **Convertir - Number.parseFloat(text):** Declara una variable **text** con un valor numérico en formato de cadena y utiliza **Number.parseFloat()** para convertirlo a un número de punto flotante. Muestra el resultado en consola.
15. **Convertir - Number.parseFloat(text, radix):** Declara una variable **text** con un valor numérico en formato de cadena y utiliza **Number.parseFloat()** con una base específica para convertirlo a un número de punto flotante. Muestra el resultado en consola.
16. **Convertir - .toString:** Declara una variable **num** y utiliza el método **.toString()** para convertirlo a una cadena. Muestra el resultado en consola.
17. **Método Math.abs(x):** Declara una variable **num** con un valor negativo y utiliza **Math.abs()** para obtener su valor absoluto. Muestra el resultado en consola.
18. **Método Math.sign(x):** Declara una variable **num** con un valor y utiliza **Math.sign()** para obtener su signo. Muestra el resultado en consola.
19. **Método Math.exp(x):** Declara una variable **num** y utiliza **Math.exp()** para calcular la exponenciación de **num**. Muestra el resultado en consola.
20. **Método Math.expm1(x):** Declara una variable **num** y utiliza **Math.expm1()** para calcular **e^x - 1**. Muestra el resultado en consola.
21. **Método Math.max(a, b, c...):** Utiliza **Math.max()** para encontrar el valor máximo entre varios números. Muestra el resultado en consola.
22. **Método Math.min(a, b, c...):** Utiliza **Math.min()** para encontrar el valor mínimo entre varios números. Muestra el resultado en consola.
23. **Método Math.pow(base, exp):** Utiliza **Math.pow()** para calcular la potencia de un número. Muestra el resultado en consola.
24. **Método Math.sqrt(x):** Utiliza **Math.sqrt()** para calcular la raíz cuadrada de un número. Muestra el resultado en consola.
25. **Método Math.cbrt(x):** Utiliza **Math.cbrt()** para calcular la raíz cúbica de un número. Muestra el resultado en consola.
26. **Método Math.imul(a, b):** Utiliza **Math.imul()** para calcular la multiplicación de dos números como un entero de 32 bits. Muestra el resultado en consola.
27. **Método Math.clz32(x):** Utiliza **Math.clz32()** para contar los ceros principales de un número en su representación de 32 bits. Muestra el resultado en consola.

28. **Método Math.random():** Utiliza **Math.random()** para generar un número pseudoaleatorio entre 0 (inclusive) y 1 (exclusive). Muestra el resultado en consola.
29. **Método de redondeo - Math.round(x):** Utiliza **Math.round()** para redondear un número al entero más cercano. Muestra el resultado en consola.
30. **Método de redondeo - Math.ceil(x):** Utiliza **Math.ceil()** para redondear hacia arriba un número al entero más cercano. Muestra el resultado en consola.
31. **Método de redondeo - Math.floor(x):** Utiliza **Math.floor()** para redondear hacia abajo un número al entero más cercano. Muestra el resultado en consola.
32. **Método de redondeo - Math.fround(x):** Utiliza **Math.fround()** para convertir un número a su representación de punto flotante de 32 bits más cercana. Muestra el resultado en consola.
33. **Método de redondeo - Math.trunc(x):** Utiliza **Math.trunc()** para truncar la parte decimal de un número. Muestra el resultado en consola.
34. **Método trigonométrico - Math.sin(x):** Utiliza **Math.sin()** para calcular el seno de un ángulo en radianes. Muestra el resultado en consola.
35. **Método trigonométrico - Math.cos(x):** Utiliza **Math.cos()** para calcular el coseno de un ángulo en radianes. Muestra el resultado en consola.
36. **Método trigonométrico - Math.tan(x):** Utiliza **Math.tan()** para calcular la tangente de un ángulo en radianes. Muestra el resultado en consola.
37. **Método trigonométrico - Math.hypot(a, b...):** Utiliza **Math.hypot()** para calcular la longitud de la hipotenusa de un triángulo a partir de sus lados. Muestra el resultado en consola.
38. **Operadores aritméticos con .toFixed(digits), Number.parseInt(text):** Declara dos variables numéricas y realiza una operación aritmética que involucre **.toFixed()** y **Number.parseInt()**. Muestra el resultado en consola.
39. **Operadores aritméticos con Number.parseFloat(text), .toPrecision(size):** Declara dos variables numéricas en formato de cadena y realiza una operación aritmética que involucre **Number.parseFloat()** y **.toPrecision()**. Muestra el resultado en consola.
40. **Operadores aritméticos con Math.random(), Math.abs(x):** Utiliza **Math.random()** para generar un número y realiza una operación aritmética que involucre **Math.abs()**. Muestra el resultado en consola.
41. **Operadores aritméticos con Math.random(), .toString:** Utiliza **Math.random()** para generar un número y realiza una operación aritmética que involucre **.toString()**. Muestra el resultado en consola.
42. **Operadores aritméticos con Math.random(), Math.exp(x):** Utiliza **Math.random()** para generar un número y realiza una operación aritmética que involucre **Math.exp()**. Muestra el resultado en consola.

43. **Operadores aritméticos con Math.sqrt(x), Math.clz32(x):** Utiliza **Math.sqrt()** y **Math.clz32()** para realizar una operación aritmética. Muestra el resultado en consola.
44. **Operadores de asignación con Asignación:** Declara una variable **x** con un valor y asigna ese valor a otra variable **y**. Muestra el valor de **y** en consola.
45. **Operadores de asignación con a += b:** Declara dos variables numéricas y utiliza el operador **+=** para sumar el valor de la segunda variable a la primera. Muestra el resultado en consola.
46. **Operadores de asignación con Suma y asignación:** Declara dos variables numéricas y utiliza el operador **+=** para incrementar el valor de la primera variable sumándole el valor de la segunda. Muestra el resultado en consola.
47. **Operadores de asignación con Resta y asignación:** Declara dos variables numéricas y utiliza el operador **-=** para decrementar el valor de la primera variable restando el valor de la segunda. Muestra el resultado en consola.
48. **Operadores de asignación con Multiplicación y asignación:** Declara dos variables numéricas y utiliza el operador ***=** para multiplicar el valor de la primera variable por el valor de la segunda. Muestra el resultado en consola.
49. **Operadores de asignación con División y asignación:** Declara dos variables numéricas y utiliza el operador **/=** para dividir el valor de la primera variable por el valor de la segunda. Muestra el resultado en consola.
50. **Operadores de asignación con Módulo y asignación:** Declara dos variables numéricas y utiliza el operador **%=** para calcular el residuo de la división de la primera variable por la segunda. Muestra el resultado en consola.
51. **Operadores de asignación con Exponenciación y asignación:** Declara una variable numérica y utiliza el operador ****=** para elevarla a una potencia específica. Muestra el resultado en consola.
52. **Operadores unarios Incremento:** Declara una variable numérica y utiliza el operador **++** para incrementar su valor en 1. Muestra el resultado en consola.
53. **Operadores unarios Decremento:** Declara una variable numérica y utiliza el operador **--** para decrementar su valor en 1. Muestra el resultado en consola.
54. **Operadores unarios Incremento previo:** Declara una variable num con un valor numérico. Utiliza el operador unario de incremento previo (++num) para aumentar el valor de num en 1 antes de asignarlo a la variable result. Muestra el nuevo valor de result en la consola.
55. **Operadores unarios Decremento previo:** Declara una variable num con un valor numérico. Utiliza el operador unario de decremento previo (--num) para disminuir el valor de num en 1 antes de asignarlo a la variable result. Muestra el nuevo valor de result en la consola.
56. **Operadores unarios Resta unaria**: Declara una variable num con un valor numérico. Utiliza el operador unario de resta (-num) para obtener el opuesto

numérico de num y asigna el resultado a la variable result. Muestra el valor resultante en la consola.