JavaScript

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl no | Function Name/Operator | Description | Example |  |
| 1 | Exponentiation operator | The exponentiation operator (\*\*) will return the first operand’s power of the second operand. | a \*\* b is equivalent to ab, which is equivalent to Math.pow(a, b) | console.log(10 \*\* 2); // 100 |
| 2 | Length | The length property returns the length of a string: | let text = "ABCDEFGHIJ";  let length = text.length; |  |
| 3 | at(),charAt(),[ ] | This method returns the character at a specified index (position) in a string | let text = "HELLO WORLD";  let char = text.charAt(0); |  |
| 4 | charCodeAt() | The charCodeAt() method returns the code of the character at a specified index in a string | let text = "HELLO WORLD";  let char = text.charCodeAt(0); |  |
|  | Slice() | slice() extracts a part of a string and returns the extracted part in a new string | let text = "Apple, Banana, Kiwi";  let part = text.slice(7, 13); | Slice out a portion of a string from position 7 to position 13: |
|  | substring() | substring() is similar to slice() | let str = "Apple, Banana, Kiwi";  let part = str.substring(7, 13); |  |
|  | Substring() | The difference is that start and end values less than 0 are treated as 0 in substring(). | let str = "Apple, Banana, Kiwi";  let part = str.substring(7, 13); |  |
|  | substr() | The difference is that the second parameter specifies the length of the extracted part. | let str = "Apple, Banana, Kiwi";  let part = str.substr(7, 6); |  |