|  |  |
| --- | --- |
| **­­­Reverse String**  **Given a string, return a new *string* with the reversed order of characters**  **--- Examples**  **reverse('apple') === 'leppa'**  **reverse('hello') === 'olleh'**  **reverse('Greetings!') === '!sgniteerG'**  1. method chaining  *function* reverse(*str*) {  return str.split('').reverse().join('');  }  2. for loop    *function* reverse(*str*) {  *let* reversed = '';  for (*let* character of str) { // or (let i = str.length - 1; i >= 0; i--)  reversed = character + reversed;  }  return reversed;  }  3. helper function  *function* reverse(*str*) {  return str.split('').reduce((*rev*, *char*) *=>* char + rev, '');  } | **Palindrome**  **Given a string, return true if the string is a palindrome or false if it is not. Palindromes are strings that form the same word if it is reversed. \*Do\* include spaces and punctuation in determining if the string is a palindrome.**  **--- Examples:**  **palindrome("abba") === true**  **palindrome("abcdefg") === false**  1. reverse str and compare  function palindrome(str) {  let reversed = str.split('').reverse().join('');  return reversed == str;  }  2. every() method  function palindrome(str) {  return str.split('').every((char, i) => {  return char == str[str.length – i – 1];  });  } |