**Learning Objectives**

* To grasp common methods in string.

**Learning Contents**

* In JavaScript, all text data is saved as string. In JavaScript, there is not such a data type that stores a single character. Even if one character is saved, it is a string whose length is 1.
* A string can be indicated with double quotes (**"**) or a single quote (**'**), so the following two strings are both valid:
* **var firstName = 'Hello';**
* **var lastName = "World";**
* In other programming languages, single quote and double quotes will influence interpretation of strings, but there is no difference between these two syntaxes in JavaScript. A string with double quotes is totally identical with a string with single quote. However, a string starting with double quotes must be ended with double quotes, and a string starting with single quote must be ended with single quote. For example: Expression of the following string will lead to a syntax error.
* **var firstName = 'Hello World"; // Syntax error (the left and right quote must be matched with each other.)**
* In a string, **+** indicates connection. It means connect the right string with the left string. For example:
* **var text = 'hello ';**
* **text = text + 'world'; *// We can also use +=: text+='world';***
* **console.log(text); *// hello world***
* Character literal
  + In JavaScript, the data type of String contains some special character literals, also called escape character, which is used to indicate nonprinting characters or characters for other purposes. Commonly used character literals are shown as below:
  + **\n**: Linefeed
  + **\t**: Tab
  + **\b**: Backspace
  + **\r**: Carriage return
  + **\\**: Slash（\）
  + **\'**: Single quote（'）
  + **\"**: Double quotes（"）
* These character literals may appear at any position of a string, and will be parsed as a character. For example:
* **var text = 'Hello \n World'; *// A line break (\n) is added.***
* The following strings will be printed:
* **"Hello**
* **World"**
* Length of any string can be got by accessing to the attribute **length**. For example:
* **var text = 'Hello ';**
* **console.log(text.length); *// 6 (Note: There is a space behind Hello.)***
* Features of String
  + In JavaScript, string cannot be changed. In other words, once a string is created, its values cannot be modified. To modify a string whose variables have been saved, we need to destroy original string at first. Then another string containing new values should be used to fill the variable. For example:
  + **var text = 'tws ';**
  + **text = text + 'academy';**
  + In this case, the variable **text** contains the string **'tws '** initially. However, in the second line, the value of **text** is redefined as a combination of **'tws '** and **'academy'**, i.e. **'tws academy'**. In this course, we will create a new string which can contain 11 characters; then **'tws'** and **'academy'** are filled into the string. The last step is to destroy original string **'tws'** and the string **'academy'**, because these two strings are useless.
* Commonly Used String Methods
  + Take a single character out of a string
    - There are two methods: the first method is to use **charAt()** method. For example: **'cat'.charAt(1); // 'a'**; the other method is to take string as an array-like object. Every character corresponds to a numerical index. For example: **'cat'[1]; // 'a'**.> Here although a bracket is used to access the string, it cannot be deleted or added, as string cannot be changed.
  + **concat()** method - string concatenation
    - **concat()** method means one or multiple strings are concatenated with original string to form and return a new string. **concat()** method does not affect original string. For example:
    - **var hello = "Hello, ";**
    - **console.log(hello.concat("tws", " have a nice day.")); *// Hello, tws have a nice day.***
    - **console.log(hello); *// Hello,***
    - We find that, actually the role of **concat()** method is the same as the role of **+**, **+=**. Application of **+**, **+=** will get better performance experience, so it is recommended that we should use assignment operator （**+**, **+=**） to substitute **concat()** method.
  + includes() method – string search
    - includes() method is used to determine whether a string is included in another string and return true or false as the case may be. This method is case sensitive. For example:
    - **'Blue Whale'.includes('blue'); *// false (case sensitive)***
    - **'Blue Whale'.includes('Blue'); *// true***
  + substr() method – extraction of substring
    - substr() method returns characters from a designated position to another designated position in a string. In order to call this method, it is necessary to input two parameters: **starting position of character extraction** and **length of characters extracted** (optional). For example:
    - **var str = 'abcdefghij';**
    - **str.substr(0,3); *// 'abc'***
    - **str.substr(3,3); *// 'def'***
    - **str.substr(3); *// 'defghij'***
  + substring() method - Extraction of substring
    - substring() method returns a subset of a string from index starting position to index ending position or a subset from index starting position to the end of the string. This method also accepts two parameters. The first parameter is **starting position of character extraction**. Being different from substr() method, the second parameter of substring() method is **ending position of character extraction**(optional). For example:
    - **var str = 'abcdefghij';**
    - **str.substring(0,3); *// 'abc'***
    - **str.substring(3,3); *// '' ( There is no character between 3 to 3.)***
    - **str.substring(3); *// 'defghij'***
    - **str.substring(2,3); *// 'c'***

**Recommended Resources**

* JS String（<https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/String>）
* Summarization of JS String Methods（<https://www.impressivewebs.com/javascript-string-methods-reference/>）