INFO 2302 Web Technologies

Filtering, searching, and sorting

Dr. Marini Othman
Department of Information Systems
Kulliyyah of Information and Communication Technology
International Islamic University Malaysia

LEADING THE WAY
KHALĪFAH • AMĀNAH • IQRA' • RAḤMATAN LIL-ĀLAMĪN

If your pages contain many data,
there are 3 techniques to help your users
find the content they are looking for:-

filtering, searching, and sorting

Filtering

Allows you to reduce to a set of values.

Also allows to create a subset of data that fulfills a certain criteria.

Searching

Search is like filtering but shows only results that match a search term.

Sorting

Sorting lets you reorder a set of items on the page based on criteria (for example, alphabetically).


INFO 2302 Web Technologies
Filtering
الجامعة الإسلامية العالمية ماليزيا
INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA
Garden of Knowledge and Virtue
LEADING THE WAY
KHALĪFAH • AMĀNAH • IQRA' • RAḤMATAN LIL-ĀLAMĪN
RCE
GREATER GOMBAK
ACKNOWLEDGED BY
UNITED NATIONS UNIVERSITY

The filter() method

The filter() method of Array instances creates a shallow copy of a portion of a given array, filtered down to just the elements from the given array that pass the test implemented by the provided function.

Simple filtering example

```
const cats = ["Leopard", "Serval", "Jaguar", "Tiger", "Caracal", "Lion"];

const filtered = cats.filter((cat) => cat.startsWith("L"));
console.log(filtered);
// [ "Leopard", "Lion" ]


const words = ['spray', 'elite', 'exuberant', 'destruction', 'present'];
const result = words.filter((word) => word.length > 6);

console.log(result);
// ["exuberant", "destruction", "present"]
```

Simple filtering example (continued)

```
const array = [-3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13];

function isPrime(num) {
  for (let i = 2; num > i; i++) {
    if (num % i === 0) {
      return false;
    }
  }
  return num > 1;
}

console.log(array.filter(isPrime));
// [2, 3, 5, 7, 11, 13]
```


INFO 2302 Web Technologies
Searching
الجامعة الإسلامية العالمية ماليزيا
INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA
يونيورسيتي إسلام انتارابغسا ملسيا
Garden of Knowledge and Virtue
LEADING THE WAY
KHALĪFAH • AMĀNAH • IQRA' • RAḤMATAN LIL-ĀLAMĪN
GREATER GOMBAK
REGIONAL CENTRE OF EXPERTISE ON EDUCATION FOR SUSTAINABLE DEVELOPMENT
ACKNOWLEDGED BY
UNITED NATIONS UNIVERSITY

The search() method

The search() method of String values executes a search for a match between a regular expression and this string, returning the index of the first match in the string.

*Refer https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions/Cheatsheet for the MDN Regular expressions cheatsheet

Simple searching example using regular expressions

```
const str = "hey JudE";
const re = /[A-Z]/;
const reDot = /[.]/;

console.log(str.search(re));
// returns 4, which is the index of the first capital letter "J"

console.log(str.search(reDot));
// returns -1 cannot find '.' dot punctuation
```

Simple contact search example

```
<!DOCTYPE html>
<html lang="en-US">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width">
    <title>Simple contact search example</title>
  </head>
  <body>

  <label for="search">Search by contact name: </label>
  <input id="search" type="text">
  <button>Search</button>

  <p></p>

    <script>
    const contacts = ['Chris:2232322',
                      'Sarah:3453456',
                      'Bill:7654322',
                      'Mary:9998769',
                      'Dianne:9384975'];
 const para = document.querySelector('p');
    const input = document.querySelector('input');
    const btn = document.querySelector('button');

    btn.addEventListener('click', () => {
      const searchName = input.value.toLowerCase();
      input.value = '';
      input.focus();
      para.textContent = '';
      for (let contact of contacts) {
        let splitContact = contact.split(':');
        if (splitContact[0].toLowerCase() === searchName) {
          para.textContent = splitContact[0] + '\'s number is ' +
splitContact[1] + '.';
          break;
        }
      }
      if (para.textContent === '') {
       para.textContent = 'Contact not found.';
     }
    });

    </script>
  </body>
</html>
```


INFO 2302 Web Technologies
Sorting
الجامعة الإسلامية العالمية ماليزيا
INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA
Garden of Knowledge and Virtue
LEADING THE WAY
KHALĪFAH • AMĀNAH • IQRA' • RAḤMATAN LIL-ĀLAMĪN
GREATER GOMBAK
ACKNOWLEDGED BY
UNITED NATIONS UNIVERSITY

The sort() method

The sort() method sorts the elements of an array in place and returns the array. The default sort order is according to string Unicode code points.

```
//sorting strings
var fruit = ['cherries', 'apples', 'bananas'];
fruit.sort();
// ['apples', 'bananas', 'cherries']

//sorting numbers
var scores = [1, 10, 21, 2];
scores.sort();
// [1, 10, 2, 21]
// Note that 10 comes before 2,
// because '10' comes before '2' in Unicode code point order.
```

The sort() method (continued)

```
//sorting a combination of number and letters
var things = ['word', 'Word', '1 Word', '2 Words'];
things.sort();
// ['1 Word', '2 Words', 'Word', 'word']
// In Unicode, numbers come before upper case letters,
// which come before lower case letters.
```

Changing order using the compare function

The sort () method only ever compares two values at a time (you will see these referred to as *a* and *b),* and it determines whether value *a* should appear before or after value *b.*

If you want to change the order of the sort, you write a compare function. A compare function should return a number. That number indicates which of the items should come first.

If compare function returns,

- <0, indicates that it should show *a* before *b*
- 0, indicates that the items should remain in the same order
- >0, indicates that it should show *b* before *a*

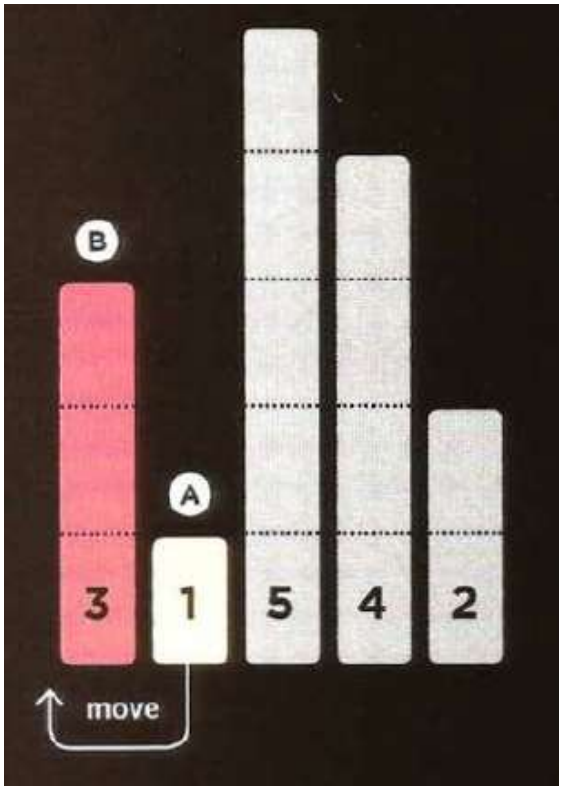
Example, consider an array with the values 3, 1, 5, 4, 2.

a should go before *b*

1 – 3 = -2

a – b = <0

```
var prices = [3, 1, 5, 4, 2]; //numbers stored in an array
prices.sort(function() {      //two values are compared
      return a - b;           //decides which goes first
} ) ;
```



***It is up to the browser to sort in different order. This example shows the order used by Safari. Other browsers sort items in different order.**

```
//Using a compare function to sort in ascending order
var prices = [1, 2, 125, 2, 19, 14];
prices.sort(function(a,b) {
   return a – b;
}

//Using a compare function to sort in descending order
var prices = [1, 2, 125, 2, 19, 14];
prices.sort(function(a,b) {
   return b – a;
}
```

References

MDN Web Docs. Looping codes. https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Building_blocks/Looping_code#exiting_loops_with_break

Duckett. JavaScript and Jquery: interactive front-end web development- Chapter 12: Filtering, searching and sorting https://www.javascriptbook.com/code/c12/