

Training Day20 report

5 July 2024

Another properties and methods of array:

Shifting Elements

Shifting is equivalent to popping, but working on the first element instead of the last.

JavaScript Array shift():

The shift() method removes the first array element and "shifts" all other elements to a lower index.

The shift() method returns the value that was "shifted out".

```
let user= [ "NAME=Anoop Kumar",CRN=2215017,URN=2203404,"BRANCH=CSE",PhoneNo=9914608994]
console.log(user.shift());
console.log("Original list:" ,user):
```

Output:

```
PS C:\Users\dell\Documents\Javascript.js> node
"c:\Users\dell\Documents\Javascript.js\array1.js"
```

NAME=Anoop Kumar

Original list: ['CRN=2215017', 'URN=2203404', 'BRANCH=CSE', 'PhoneNo=9914608994']

JavaScript Array unshift()

The unshift() method adds a new element to an array (at the beginning), and "unshifts" older elements.

The unshift() method returns the new array length.



```
let user= [ "NAME: Anoop Kumar","CRN:2215017","URN:2203404","BRANCH:
CSE","PhoneNo:9914608994"]
console.log(user.shift());// shift( shift at the front)
console.log("Original list:" ,user);
console.log("Pop:",user.pop());// pop (pop at the end)
user.unshift("ROOM: G13");// unshift(unshift at the front)
console.log(user);
```

Output:

PS C:\Users\dell\Documents\Javascript.js> node
"c:\Users\dell\Documents\Javascript.js\array1.js"

NAME: Anoop Kumar

Original list: ['CRN:2215017', 'URN:2203404', 'BRANCH:
CSE','PhoneNo:9914608994']

Pop: PhoneNo:9914608994

['ROOM: G13', 'CRN:2215017', 'URN:2203404', 'BRANCH: CSE']

Changing Elements

Array elements are accessed using their **index number**.

```
let user= [ "NAME:Anoop
Kumar","CRN:2215017","URN:2203404","BRANCH:CSE","PhoneNo:9914608994"]
//console.log(user.shift());// shift( shift at the front)
//console.log("Original list:" ,user);
//console.log("Pop:",user.pop());// pop (pop at the end)
//user.unshift("ROOM:G13");// unshift(unshift at the front)
//console.log(user);
//user.push("LAB:DS LAB");// push (push at the end)
console.log(user);
console.log(user[0]);
console.log(user[1]);
console.log(user[2]);
console.log(user[3]);
console.log(user[user.length-1]);
```

Output:

```
PS C:\Users\dell\Documents\Javascript.js> node array1.js
```

Debugger attached.

```
[  
  'NAME:Anoop Kumar',  
  
  'CRN:2215017',  
  'URN:2203404',  
  'BRANCH:CSE',  
  'PhoneNo:9914608994'  
]  
NAME:Anoop Kumar  
CRN:2215017  
URN:2203404  
BRANCH:CSE  
PhoneNo:9914608994
```

Splicing and Slicing Arrays

The splice() method adds new items to an array.

The slice() method slices out a piece of an array.

JavaScript Array splice()

The splice() method can be used to add new items to an array.

```
let user= [ "NAME: Anoop Kumar  
", "CRN:2215017", "URN:2203404", "BRANCH:CSE", "PhoneNo:9914608994"]  
user.splice(2, 2, "Lemon", "Kiwi");  
console.log(user);
```

Output:

```
PS C:\Users\dell\Documents\Javascript.js> node  
"c:\Users\dell\Documents\Javascript.js\array1.js" [  
  'NAME:Anoop Kumar',  
  'CRN:2215017',  
  'Lemon',
```

```
'Kiwi',  
'PhoneNo:9914608994'  
]
```

JavaScript Array slice()

The slice() method slices out a piece of an array into a new array.

```
let user= [ "NAME: Anoop Kumar  
", "CRN:2215017", "URN:2203404", "BRANCH:CSE", "PhoneNo:9914608994"]  
let citrus = user.slice(1, 4);  
console.log(citrus);  
console.log(user)
```

Output:

```
PS C:\Users\dell\Documents\Javascript.js> node  
"c:\Users\dell\Downloads\array1.js"  
[ 'CRN:2215017', 'URN:2203404', 'BRANCH:CSE' ]  
[  
  'NAME:Anoop Kumar',  
  'CRN:2215017',  
  'URN:2203404',  
  'BRANCH:CSE',  
  'PhoneNo:9914608994'  
]
```

JavaScript Array map():

```
let user= [  
  {  
    NAME:" Anoop Kumar ",  
    CRN:"2215017",  
    URN:"2203404",  
    BRANCH:"CSE",  
    PhoneNo:"9914608994"  
  },  
  {  
    NAME:" Anoop Kumar ",  
    CRN:"2215017",  
    URN:"2203404",  
    BRANCH:"CSE",
```

```

        PhoneNo: "9914608994"
      },
      {
        NAME: " Anoop Kumar ",
        CRN: "2215017",
        URN: "2203404",
        BRANCH: "CSE",
        PhoneNo: "9914608994"
      },
      {
        NAME: " Anoop Kumar ",
        CRN: "2215017",
        URN: "2203404",
        BRANCH: "CSE",
        PhoneNo: "9914608994"
      }
    ]
  }
}

user.map((myusers) => {
  //console.log(myusers.NAME,myusers.CRN,myusers.URN,myusers.BRANCH,myusers.PhoneNo);
  console.log(`My name is ${myusers.NAME} CRN is ${myusers.CRN} and URN is ${myusers.URN} My BRANCH is ${myusers.BRANCH} Phone number is ${myusers.PhoneNo}`);
});

```

Output:

PS C:\Users\dell\Documents\Javascript.js> node array1.js

Debugger attached.

My name is Anoop Kumar CRN is 2215017 and URN is 2203404 My BRANCH is CSE Phone number is 9914608994

My name is Anoop Kumar CRN is 2215017 and URN is 2203404 My BRANCH is CSE Phone number is 9914608994

My name is Anoop Kumar CRN is 2215017 and URN is 2203404 My BRANCH is CSE Phone number is 9914608994

My name is Anoop Kumar CRN is 2215017 and URN is 2203404 My BRANCH is CSE Phone number is 9914608994

Waiting for the debugger to disconnect...

Description

map() creates a new array from calling a function for every array element.

map() does not execute the function for empty elements.

map() does not change the original array.

Javascript Objects literal:

An object literal is a list of property names:values insides the curly braces{ }.

```
{  
  NAME:"Anoop Kumar",  
  CRN:"2215017",  
  URN:"2203404",  
  BRANCH:"CSE",  
  PhoneNo:"9914608994"  
};  
console.log(user);  
console.log(user.NAME);  
console.log(user.BRANCH);
```