DAY: -20 (5 JULY 2024)

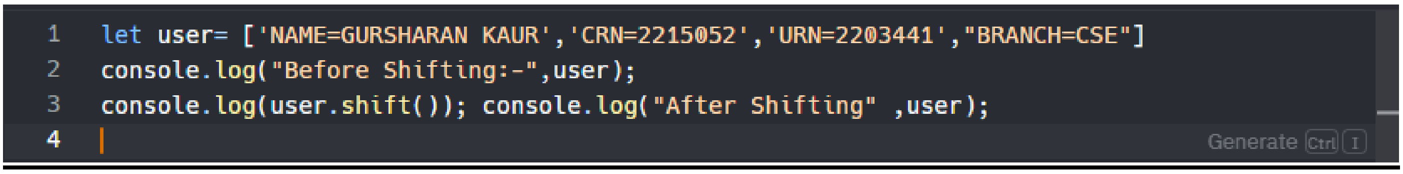
Another properties and methods of array: -

Shifting Elements: - Shifting is equivalent to popping, but working on the first element instead of thelast.

JavaScript Array shift (): -

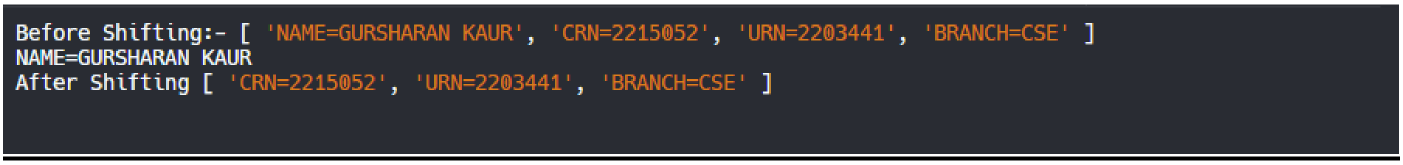
The shift () method removes the first array element and "shifts" all other elementsto a lower index.
The shift () method returns the value that was "shifted out".

EXAMPLE: -



```
let user= ['NAME=GURSHARAN KAUR','CRN=2215052','URN=2203441',"BRANCH=CSE"]
console.log("Before Shifting:-",user);
console.log(user.shift()); console.log("After Shifting" ,user);
```

OUTPUT: -

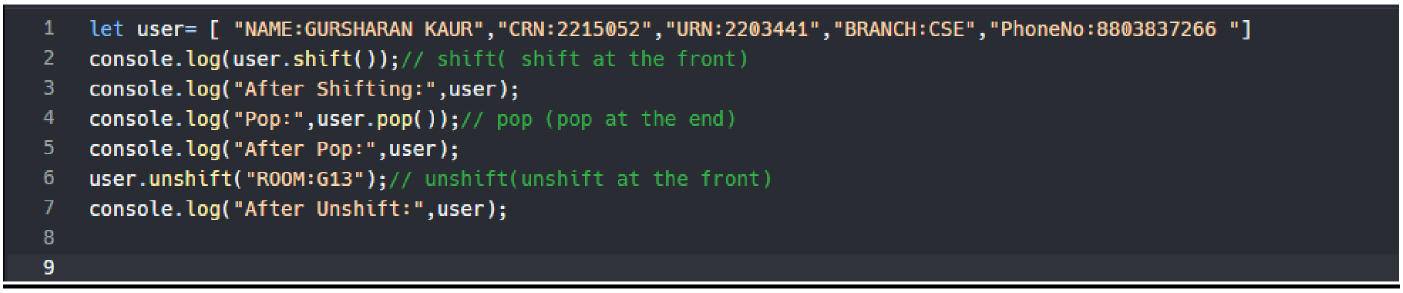


```
Before Shifting:- [ 'NAME=GURSHARAN KAUR', 'CRN=2215052', 'URN=2203441', 'BRANCH=CSE' ]
NAME=GURSHARAN KAUR
After Shifting [ 'CRN=2215052', 'URN=2203441', 'BRANCH=CSE' ]
```

JavaScript Array unshift (): -

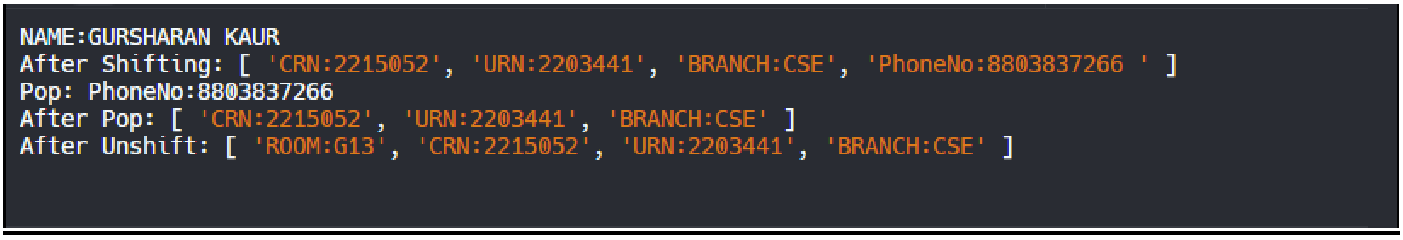
The unshift () method adds a new element to an array (at the beginning), and"unshifts" older elements.
The unshift () method returns the new array length.

EXAMPLE: -



```
let user= [ "NAME:GURSHARAN KAUR","CRN:2215052","URN:2203441","BRANCH:CSE","PhoneNo:8803837266 "]
console.log(user.shift());// shift( shift at the front)
console.log("After Shifting:",user);
console.log("Pop:",user.pop());// pop (pop at the end)
console.log("After Pop:",user);
user.unshift("ROOM:G13");// unshift(unshift at the front)
console.log("After Unshift:",user);
```

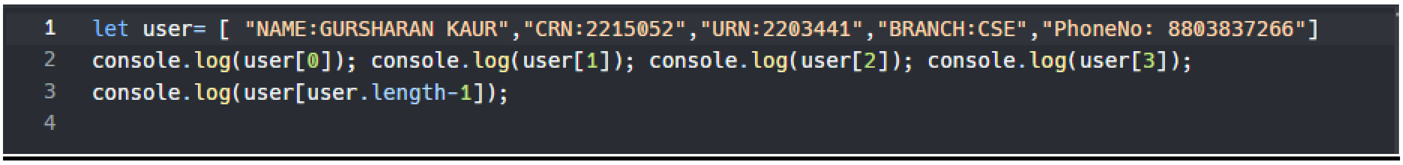
OUTPUT: -



```
NAME:GURSHARAN KAUR
After Shifting: [ 'CRN:2215052', 'URN:2203441', 'BRANCH:CSE', 'PhoneNo:8803837266 ' ]
Pop: PhoneNo:8803837266
After Pop: [ 'CRN:2215052', 'URN:2203441', 'BRANCH:CSE' ]
After Unshift: [ 'ROOM:G13', 'CRN:2215052', 'URN:2203441', 'BRANCH:CSE' ]
```

Changing Elements: - Array elements are accessed using their **index number**

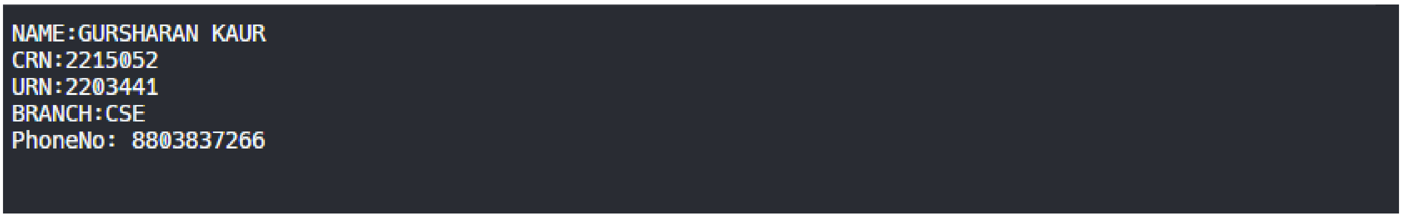
EXAMPLE: -



```javascript
let user= [ "NAME:GURSHARAN KAUR","CRN:2215052","URN:2203441","BRANCH:CSE","PhoneNo: 8803837266"]
console.log(user[0]); console.log(user[1]); console.log(user[2]); console.log(user[3]);
console.log(user[user.length-1]);

```

OUTPUT: -



```
NAME:GURSHARAN KAUR
CRN:2215052
URN:2203441
BRANCH:CSE
PhoneNo: 8803837266
```

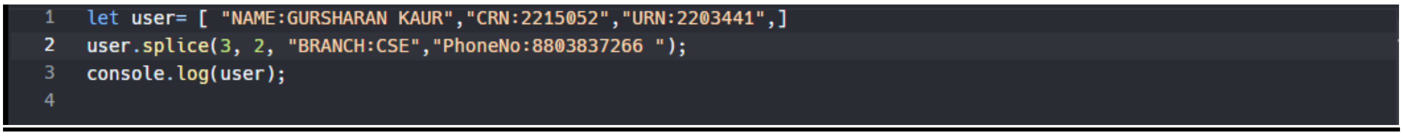
Splicing and Slicing Arrays: -

- The **splice ()** method adds new item 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.

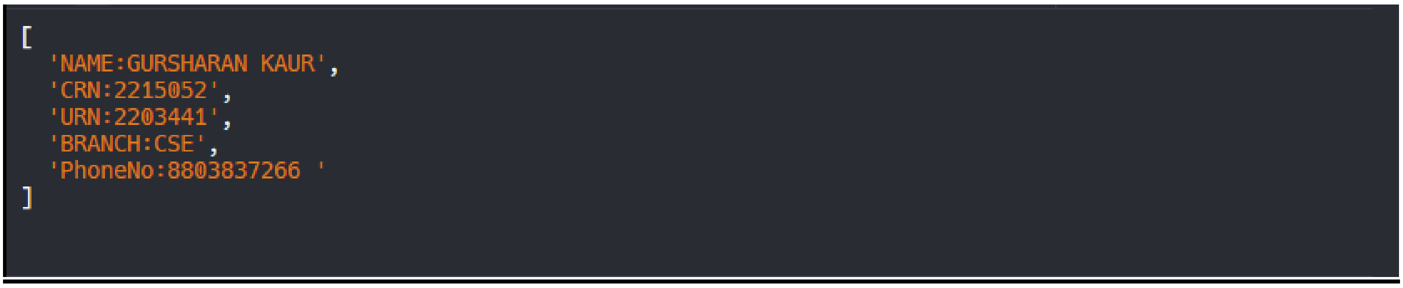
EXAMPLE: -



```javascript
let user= [ "NAME:GURSHARAN KAUR","CRN:2215052","URN:2203441",]
user.splice(3, 2, "BRANCH:CSE","PhoneNo:8803837266 ");
console.log(user);

```

OUTPUT: -

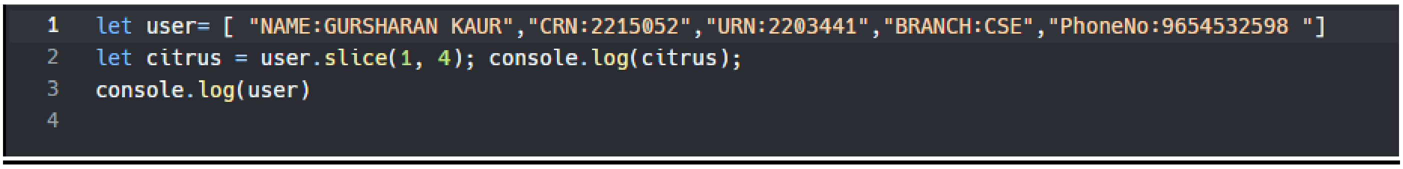


```
[
  'NAME:GURSHARAN KAUR',
  'CRN:2215052',
  'URN:2203441',
  'BRANCH:CSE',
  'PhoneNo:8803837266 '
]
```

JavaScript Array slice (): -

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

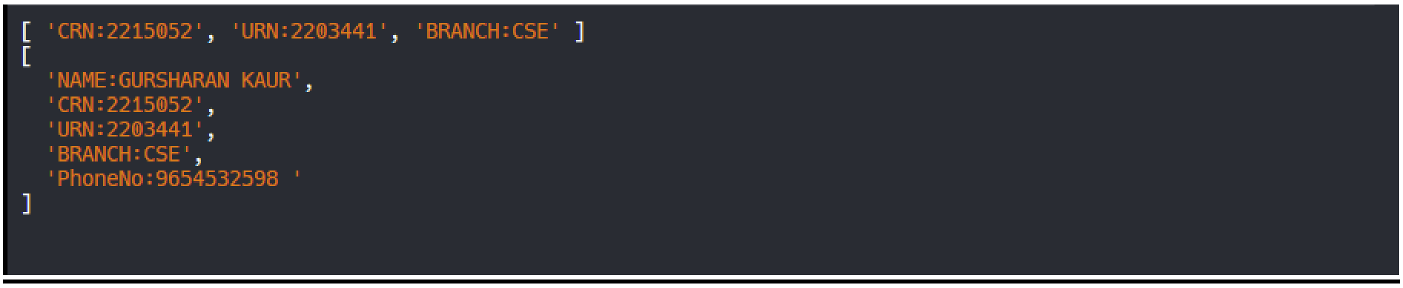
EXAMPLE: -



```
let user= [ "NAME:GURSHARAN KAUR","CRN:2215052","URN:2203441","BRANCH:CSE","PhoneNo:9654532598 "]
let citrus = user.slice(1, 4); console.log(citrus);
console.log(user)

```

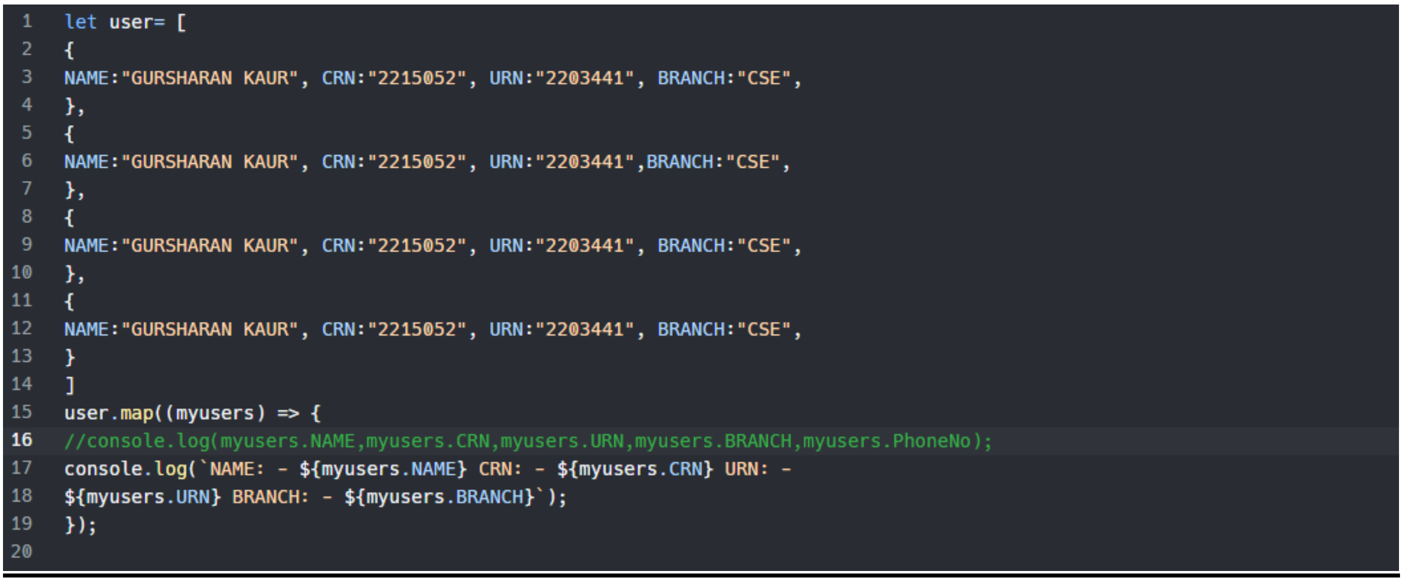
OUTPUT: -



```
[ 'CRN:2215052', 'URN:2203441', 'BRANCH:CSE' ]
[
  'NAME:GURSHARAN KAUR',
  'CRN:2215052',
  'URN:2203441',
  'BRANCH:CSE',
  'PhoneNo:9654532598 '
]
```

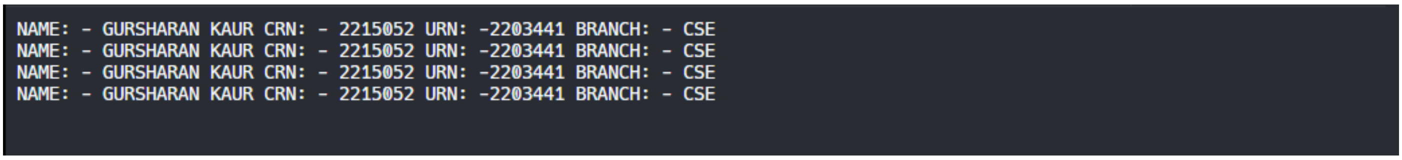
JavaScript Array map (): -

EXAMPLE: -



```
let user= [
{
NAME:"GURSHARAN KAUR", CRN:"2215052", URN:"2203441", BRANCH:"CSE",
},
{
NAME:"GURSHARAN KAUR", CRN:"2215052", URN:"2203441",BRANCH:"CSE",
},
{
NAME:"GURSHARAN KAUR", CRN:"2215052", URN:"2203441", BRANCH:"CSE",
},
{
NAME:"GURSHARAN KAUR", CRN:"2215052", URN:"2203441", BRANCH:"CSE",
}
]
user.map((myusers) => {
//console.log(myusers.NAME,myusers.CRN,myusers.URN,myusers.BRANCH,myusers.PhoneNo);
console.log(`NAME: - ${myusers.NAME} CRN: - ${myusers.CRN} URN: -
${myusers.URN} BRANCH: - ${myusers.BRANCH}`);
});
```

OUTPUT: -



```
NAME: - GURSHARAN KAUR CRN: - 2215052 URN: -2203441 BRANCH: - CSE
NAME: - GURSHARAN KAUR CRN: - 2215052 URN: -2203441 BRANCH: - CSE
NAME: - GURSHARAN KAUR CRN: - 2215052 URN: -2203441 BRANCH: - CSE
NAME: - GURSHARAN KAUR CRN: - 2215052 URN: -2203441 BRANCH: - CSE
```

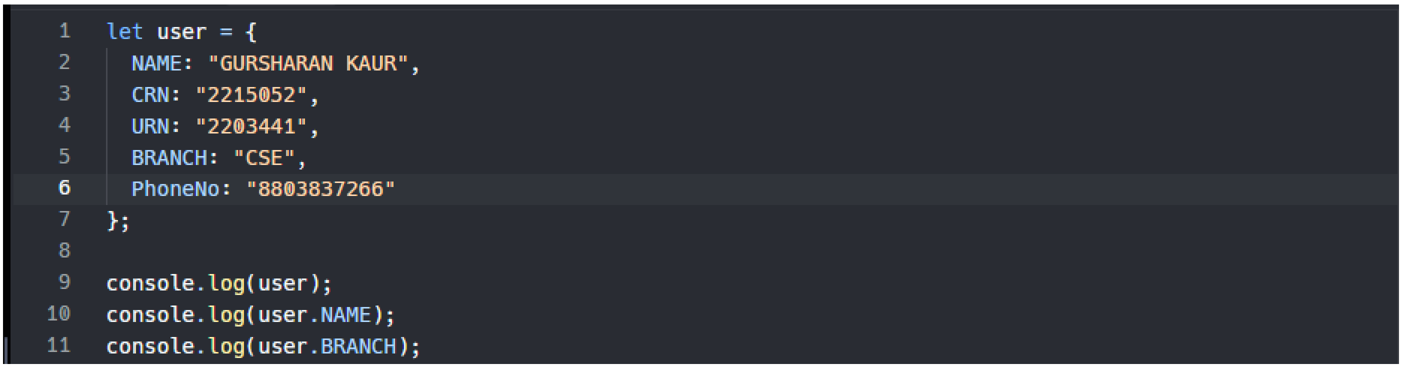
Description

- 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 {}.

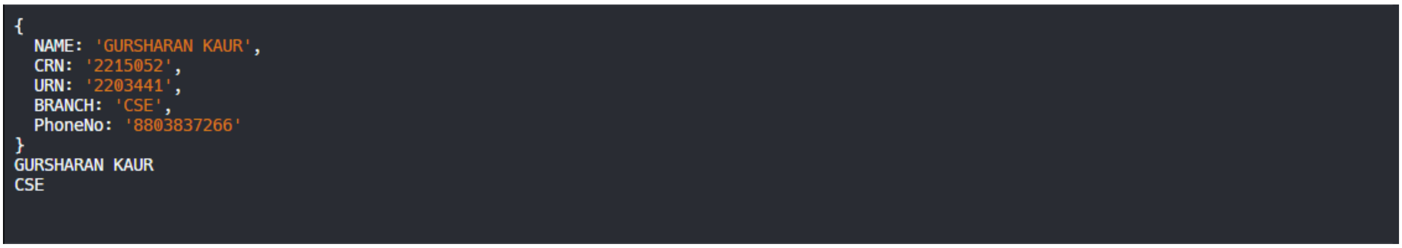
EXAMPLE: -



```
let user = {
  NAME: "GURSHARAN KAUR",
  CRN: "2215052",
  URN: "2203441",
  BRANCH: "CSE",
  PhoneNo: "8803837266"
};

console.log(user);
console.log(user.NAME);
console.log(user.BRANCH);
```

OUTPUT: -



```
{
  NAME: 'GURSHARAN KAUR',
  CRN: '2215052',
  URN: '2203441',
  BRANCH: 'CSE',
  PhoneNo: '8803837266'
}
GURSHARAN KAUR
CSE
```