JavaScript
Arrays

Naveen Pete

Arrays

- Let us group data in a list

```
var friend1 = "Satish";
var friend2 = "Bala";
var friend3 = "Prakash";
var friend4 = "Guru";
```

- Rather than have 4 different variables, we can define 1 variable with 4 names within it

```
var friends = ["Satish", "Bala", "Prakash", "Guru"];
```

- Square brackets [] denote an array
- Elements within an array are separated using comma (,)
- Array elements are indexed starting at 0
- We use the indices to retrieve data

Arrays

- Array elements are indexed starting at 0
- We use the indices to retrieve data

```
console.log(friends[0]);

console.log(friends[1] + "  " + friends[3]);
```

- We can also update an array using the index

```
friends[0] = 'Umesh';
```

- We can add new data to an existing array

```
friends[4] = 'Suresh';
```

Arrays

- We can initialize an empty array in following ways

```
var friends = [];
var colors = new Array();
```

- An array can hold any type of data

```
var items = [1, “Hari”, true, null];
```

- Arrays have a length property

```
var numbers = [12, 35, 60, 72, 50];
console.log(numbers.length);
```

Array Methods

- push() and pop()
  - Use push() to add an element to the end of an array
  - Use pop() to remove the last element in an array. Also, it returns the removed element

```
var colors = [“red”, “green”, “blue”];

colors.push(“orange”);
colors.push(“violet”);

colors.pop();
```

Array Methods

- shift() and unshift()
  - Use unshift() to add an element to the front of an array
  - Use shift() to remove the first element in an array. Also, it returns the removed element

```
var colors = [“red”, “green”, “blue”];

colors.unshift(“orange”);
colors.unshift(“violet”);

colors.shift();
```

Array Methods

- indexOf()
  - Use indexOf() to find the index of an item in an array

```
var friends = [“Satish”, “Bala”, “Prakash”, “Guru”];

friends.indexOf(‘Prakash’);

friends.indexOf(‘Vinay’);
```

- slice()
  - Use slice() to copy parts of an array
  - It does not alter the original array

```
var nums = [1, 2, 3, ‘a’, ‘b’, 4, 5];

var letters = nums.slice(3, 5);
console.log(letters);
```

Array Methods

- indexOf()
  - Use indexOf() to find the index of an item in an array

```
var friends = [“Satish”, “Bala”, “Prakash”, “Guru”];

friends.indexOf(‘Prakash’);

friends.indexOf(‘Vinay’);
```

- slice()
  - Use slice() to copy parts of an array
  - It does not alter the original array
  - To copy an entire array, do not pass any parameter to slice() method

```
var nums = [1, 2, 3, ‘a’, ‘b’, 4, 5];

var letters = nums.slice(3, 5);
console.log(letters);
```

Array Iteration

- for loop
  - To loop over an array using a for loop, we need to make use of array's length property

```
var friends = [“Satish”, “Bala”, “Prakash”, “Guru”];

for(var i = 0; i < friends.length; i++) {
  console.log(friends[i]);
}
```

- forEach
  - An easier built-in way to iterating through an array

```
var friends = [“Satish”, “Bala”, “Prakash”, “Guru”];

friends.forEach(function(friend) {
  console.log(friend);
});
```

Q & A

- Thank you!