16/09/2020

JAVASCRIPT SET METHODS

1.Set add() method

- The JavaScript Set add() method is used to add an element to Set object with a specified value.
- Each element must have a unique value.

  For example

```
<!DOCTYPE html>
<html>
<body>

<script>
var set = new Set();
set.add("jQuery");
set.add("javascript");
set.add("html");
for (elements of set) {
 document.write(elements+"<br>");
}
</script>
</body>
</html>
```

2.The Set clear() method:

- The JavaScript Set clear() method is used to remove all the elements from Set object.

```
<!DOCTYPE html>
<html>
<body>

<script>
var set = new Set();
set.add("jQuery");
set.add("javascript");
set.add("html");
document.write(set.size);
set.clear();
document.write("<br>");
document.write(set.size);
</script>
</body>
</html>
```

3.The Set delete() method

- The JavaScript Set delete() method is used to remove all the elements from Set object.

```
<!DOCTYPE html>

<html>

<body>

<script>

var set = new Set();

set.add("jQuery");

set.add("javascript");
set.add("html");

document.write(set.size);

set.delete("javascript");

document.write("<br>");

document.writeln(set.size);

</script>

</body>

</html>
```

4.The Set has() method

- The JavaScript Set has() method indicates whether the Set object contains the specified value. It returns true if the specified value is present, otherwise false.

```
<!DOCTYPE html>

<html>

<body>

<script>

var set = new Set();

set.add("javascript");

set.add("AngularJS");

set.add("html");
```

```
document.writeln(set.has("Bootstrap"));
</script>
</body>
</html>
```

5.The Set values() method

- The JavaScript Set values() method returns an object of new Set iterator. This object contains the value for each element.

  ```
  <!DOCTYPE html>
  <html>
  <body>
  <script>
  var set = new Set();
  set.add("jQuery");
  set.add("AngularJS");
  set.add("javascript");
  var itr=set.values();
  document.write(itr.next().value);
  document.write("<br>");
  document.write(itr.next().value);
  ```

```
document.write("<br>");

document.write(itr.next().value);

 </script>

</body>

</html>
```

JAVASCRIPT MAP METHODS

- The JavaScript Map method is used to map keys to values. It stores each element as key-value pair. It operates the elements such as search, update and delete on the basis of specified key.
- A map object cannot contain the duplicate keys.
- A map object can contain the duplicate values.
- The key and value can be of any type.
- A map object iterates its elements in insertion order.

1.The Map clear() method

- The JavaScript map clear() method is used to remove all the elements from Map object.

```
<!DOCTYPE html>

<html>

<body>

<script>

var map=new Map();

map.set(1,"jQuery");//key,value pair
map.set(2,"javascript");
map.set(3,"html");
document.write(map.size);
map.clear();
document.write("<br>");
document.write(map.size);
</script>
</body>
</html>
```

2. Map delete() method

- The JavaScript map delete() method is used to remove all the elements from Map object.

```
<!DOCTYPE html>
<html>
<body>
<script>
var map=new Map();
map.set(1,"jQuery");
map.set(2,"javascript");
map.set(3,"html");
```

```
document.write(map.size);
map.delete(2);
document.write("<br>");
document.write(map.size);
</script>
</body>
</html>
```

3. The Map get() method

- The JavaScript map get() method returns the value of specified key of an element from a Map object.

```
<!DOCTYPE html>
<html>
<body>
<script>
var map=new Map();
map.set(1,"jQuery");
map.set(2,"javascript");
map.set(3,"html");
document.write(map.get(1));
document.write("<br>");
document.write(map.get(2));
document.write("<br>");
document.write(map.get(3));
</script>
</body>
</html>
```

4. Map keys() method

- The JavaScript map keys() method returns an object of new Map iterator. This object contains the key for each element. It maintains insertion order.

```
<!DOCTYPE html>
<html>
<body>
<script>
var map=new Map();
map.set(1,"jQuery");
map.set(2,"javascript");
map.set(3,"html");
var itr=map.keys();
document.write(itr.next().value);
document.write("<br>");
document.write(itr.next().value);

document.write("<br>");

document.write(itr.next().value);

 </script>

</body>

</html>
```

5.The Map set() method

- The JavaScript map set() method is used to add or updates an element to map object with the particular key-value pair. Each value must have a unique key.

```
<!DOCTYPE html>

<html>

<body>

<script>

var map=new Map();

map.set(1,"jQuery");

map.set(2,"javascript");

map.set(3,"html");

document.write(map.get(1));

document.write("<br>");

document.write(map.get(2));
document.write("<br>");
document.write(map.get(3));
</script>
</body>
</html>
```

JavaScript WeakSet Methods

- The JavaScript WeakSet object is the type of collection that allows us to store weakly held objects. Unlike Set, the WeakSet are the collections of objects only.
- The main difference is that references to objects in Set are strong while references to objects in WeakSet are weak.
- This means that an object in WeakSet can be garbage collected if there is no other reference to it.

1.Weakset add() method

```
<!DOCTYPE html>
<html>
<body>
<script>
var ws = new WeakSet();
var obj1={};
var obj2={};
ws.add(obj1);
ws.add(obj2);

document.write(ws.has(obj1));

document.write("<br>");

document.write(ws.has(obj2));

</script>

</body>

</html>
```

2 WeakSet delete() method

- The JavaScript WeakSet delete() method is used to remove the specified object from WeakSet object.

```
<!DOCTYPE html>

<html>

<body>

<script>

var ws = new WeakSet();

var obj={};

ws.add(obj);

document.write(ws.has(obj));

ws.delete(obj);
```

```
document.write("<br>");

document.write(ws.has(obj));

</script>

</body>

</html>
```