Iterative Algorithms - Notes

Iteration is the process of accessing each of the items contained within a data structure. There are multiple ways to iterate through an array in JavaScript. They all have a time complexity of O(n) since the iteration is visiting n number of elements.

for loop

```
Syntax:- for (Variables; Condition; Modification)
```

for is the most common method of iteration. It is most often used in this form:

```
for ( var i=0, len=array1.length; i<len; i++ ) {
console.log(array1[i]);
}
```

The previous code simply means initialize the variable i, check whether the condition is false before executing the body (i<len), and then modify (i++) until the condition is false. Similarly, you can use a while loop. However, the counter will have to be set outside.

while loop

```
Syntax:- variable initialization
while( condition ){
// insert code here
modifications
}
```

```
var counter=0;
while( counter<array1.length){
```

```
// insert code here
counter++;
}
```

You can implement an infinite loop using a while loop, as shown here:

```
while(true){
if (breakCondition){
break;}
}
```

Similarly, a for loop can implement an infinite loop by not setting a condition, as shown here:

```
for ( ; ;) {if (breakCondition) {

break;}

}
```

for (in)

Another way to iterate a JavaScript array is to call the indices one by one. The variable specified before in is the index of the array, as follows:

```
var array1 = ['all', 'insects', 'are', 'small'];
for (var index in array1) {
console.log(index);
}
```

This prints the following: 0,1,2,3.

To print the content, use this:

```
for (var index in array1) {
console. log(array1 [index]);
}
```

This prints all, insects, are, and small.

for (of)

The variable specified before of is the element (the value) of the array, as follows:

```
for (var element of array1) {
console.log(element);
}
```

This prints out all, insects, are, and small.

forEach( )

The big difference between forEach and other methods of iteration is that forEach cannot break out of the iteration or skip certain elements in the array. forEach is more expressive and explicit by going through each element.

```
var array1 = ['all', 'cows', 'are', 'big'];
array1.forEach( function (element, index){
console.log (element);
});
array1. forEach( function (element, index){
console. log(array1 [index]);
});
```

Both print all, cows, are, and big.

do while loop

The do...while statements combo defines a code block to be executed once, and repeated as long as a condition is true.

The do...while is used when you want to run a code block at least one time.
Syntax:- variable initialization.

```
do {
// insert code here
modifications
}while( condition );
```

Example:-

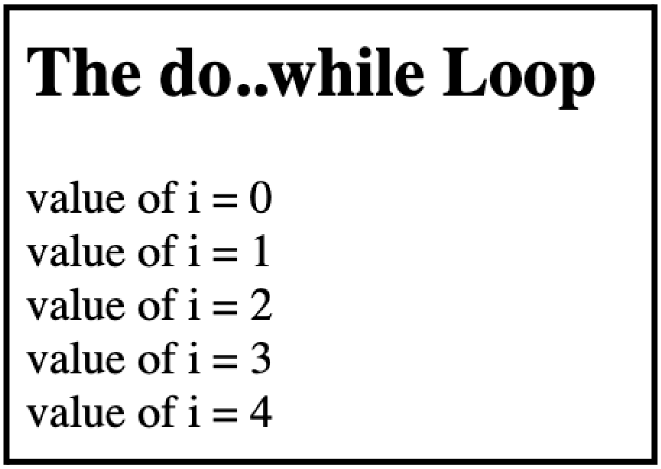
```
<!DOCTYPE html>
<html>
<body>

<h2>The do..while Loop</h2>

<p id="dowhile"></p>

<script>
let text = "";
let i = 0;
do {
  text += "value of i = " + i + "<br>";
  i++;
}
while (i < 5);
document.getElementById("dowhile").innerHTML = text;
</script>

</body>
</html>
```



The do..while Loop

```
value of i = 0
value of i = 1
value of i = 2
value of i = 3
value of i = 4
```