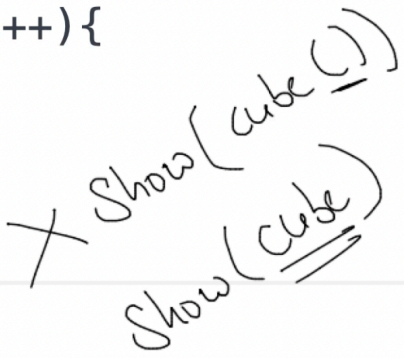
```
> function show(callBackFn){
      for(var i = 1; i<=10;i++){
          callBackFn(i);
      }
  }
< undefined
```




```
> function cube(num){
      console.log(num**3);
  }
< undefined
```

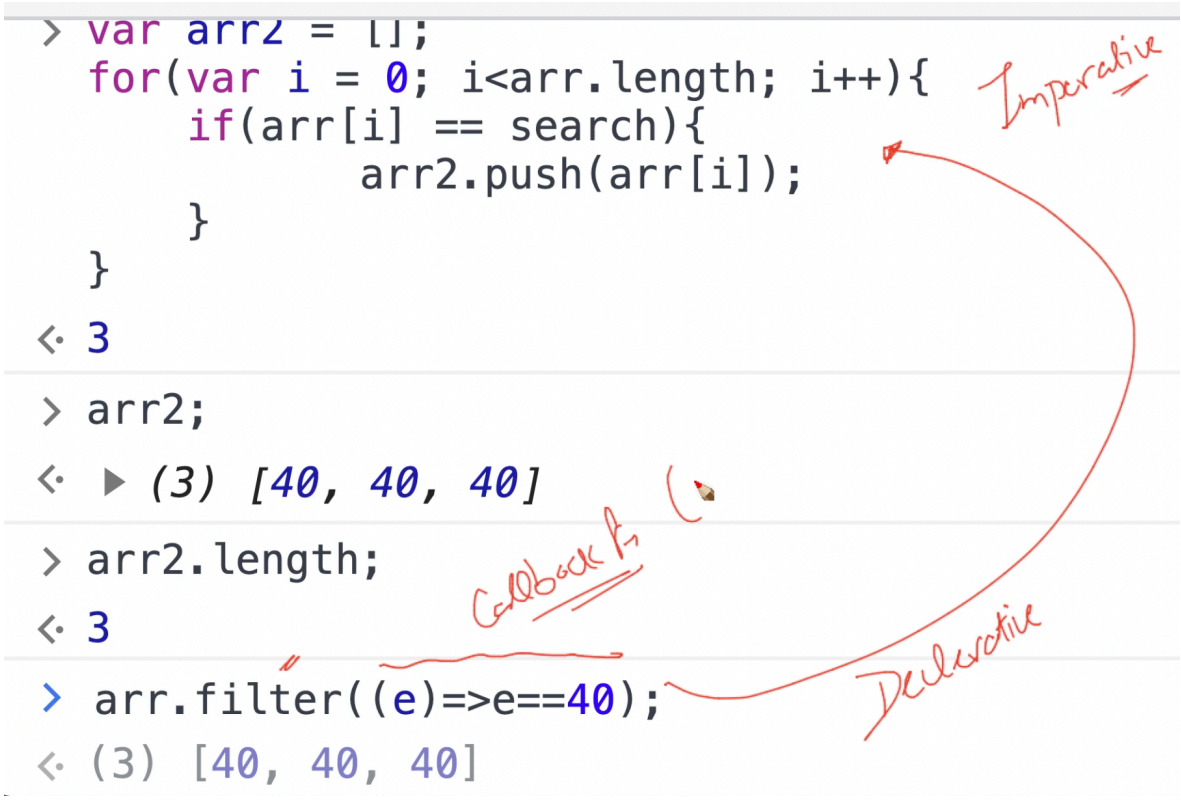
```
> show(cube);
```

```
> function calc(){
      // nested function
      var add = function(x,y){
          return x + y;
      }
      return add;
  }
```

```
> var f = calc();
< undefined
> f;
< f (x,y){
            return x + y;
        }
> typeof f;
< 'function'
> f(10,20);
< 30
```

```
> // Imperative Statements
< undefined
> var arr = [10,20,30,40,50,1,2,40,100,40];
< undefined
> var search = 40;
< undefined
> var
  for(var i = 0; i<arr.length; i++){
      if(arr[i] == search){
              count++;
      }
  }

> var arr2 = [];
  for(var i = 0; i<arr.length; i++){
      if(arr[i] == search){
              arr2.push(arr[i]);
      }
  }
< 3
> arr2;
< ▶ (3) [40, 40, 40]
```


var arr2 = [];
for(var i = 0; i<arr.length; i++){
if(arr[i] == search){
arr2.push(arr[i]);
}
}
3
arr2;
(3) [40, 40, 40]
arr2.length;
3
arr.filter((e)=>e==40);
(3) [40, 40, 40]
Imperative
Declarative

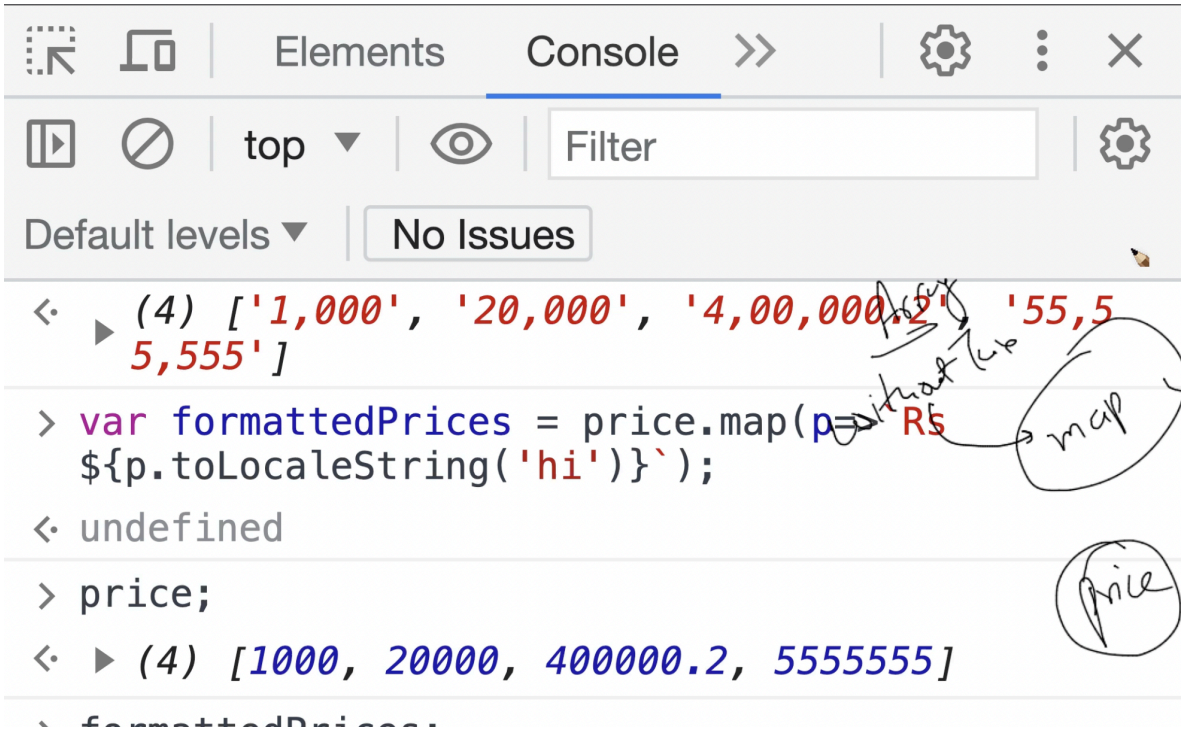
Clone

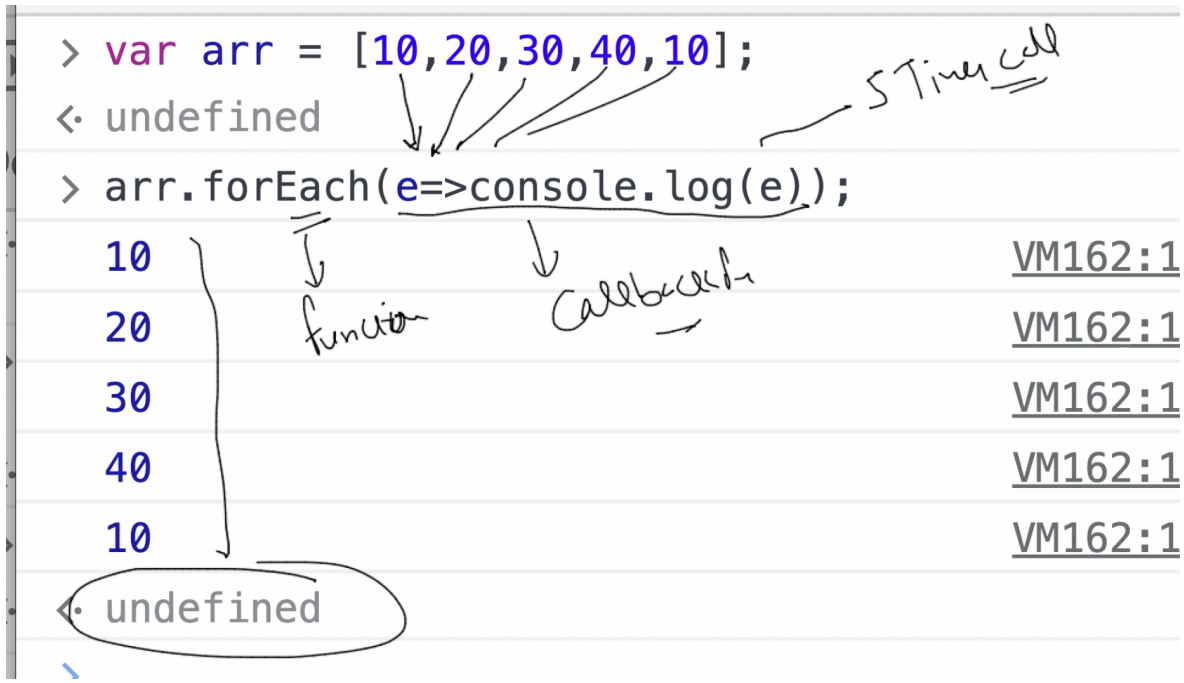
```
> var arr = [10,20,30,40,10];
< undefined
> var e = [...arr];
< undefined
> e;
< ▶ (5) [10, 20, 30, 40, 10]
> arr === e;
< false
```

```
> var f = arr.filter(w=>true);
< undefined
> f;
< ▶ (5) [10, 20, 30, 40, 10]
> f === arr;
< false
```

```
> // filter vs map
< undefined
> arr;
< ▶ (5) [10, 20, 30, 40, 10]
> arr;
< ▶ (5) [10, 20, 30, 40, 10]
> arr.map(e=>e + e * 0.18);
< (5) [11.8, 23.6, 35.4, 47.2, 11.8]

> var h = arr.map(e=>e);
< undefined
> h;
< ▶ (5) [10, 20, 30, 40, 10]
> h === arr;
< false
```


Elements
Console
top
Filter
Default levels
No Issues
(4) ['1,000', '20,000', '4,00,000.2', '55,55,555']
var formattedPrices = price.map(p=> `Rs ${p.toLocaleString('hi')}`);
undefined
price;
(4) [1000, 20000, 400000.2, 5555555]
map
price


var arr = [10,20,30,40,10];
undefined
arr.forEach(e=>console.log(e));
5 Times call
function
Callback
10
20
30
40
10
VM162:1
undefined

```
> arr.forEach((element,index, a)=>console.log(element, index, a));
  10 0 ▶ (5) [10, 20, 30, 40, 10]   VM277:
  20 1 ▶ (5) [10, 20, 30, 40, 10]   VM277:
  30 2 ▶ (5) [10, 20, 30, 40, 10]   VM277:
  40 3 ▶ (5) [10, 20, 30, 40, 10]   VM277:
  10 4 ▶ (5) [10, 20, 30, 40, 10]   VM277:
```

```
> arr.find(e=>e==10)?"Found":"Not found";
< 'Found'
> arr.findIndex(e=>e==10);
< 0
> arr;
< ▶ (5) [10, 20, 30, 40, 10]
> arr.findIndex(e=>e==1000);
< -1
```

```
> arr.filter(e=>e>20);
```

```
▶ (2) [30, 40]
```

```
> arr.reduce((result, e)=>{
      if(e>20){
      result.push(e);
      }
      return result;
  },[]);
```

```
▶ (2) [30, 40]
```

```
> arr.reduce((result, e, i)=>{
      if(e>20){
      result.push({element: e, index : i});
      }
      return result;
  },[]);
```