# JavaScript for Developers

## Callbacks | Utility Functions

# Callbacks

- ✓ Asynchronous programming
- ✓ Use callbacks as utility functions

# Callbacks with array functions

- ✓ **For loop**

```
const arr = [1, 2, 3, 4, 5, 6];
```

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

```
for (const ai of arr) {
    console.log(ai);
}
```

# Callbacks with array functions

## ✓ For loop

```
const arr = [1, 2, 3, 4, 5, 6];
```

```
function printNumber(n){
    console.log(n);
}
```

```
arr.forEach(printNumber);
```

→

```
arr.forEach((n) => console.log(n));
```

```
const printNumber = function(n){
    console.log(n);
}
```

```
const printNumber = (n) => console.log(n);
```

# Callbacks with array functions

✓ For loop

```
const arr = [1, 2, 3, 4, 5, 6];
```

↓

```
arr = [2, 4, 6, 8, 10, 12];
```

```
arr.forEach((n) => {
    n = n * 2;
});
```

```
arr.forEach((n, i, a) => {
    a[i] = a[i] * 2;
});
```

# Callbacks with array functions

## ✓ filter

```
const arr = [1, 2, 3, 4, 5, 6];
```

↓

```
oddNumbers = [1, 3, 5];
```

```
const oddNumbers = arr.filter((n) => n%2 != 0);
```

```
function isOddNumber(n){
    return n%2 != 0;
}
```

```
const oddNumbers = [];
for (const n of arr) {
    if(isOddNumber(n)){
        oddNumbers.push(n);
    }
}
```

# Callbacks with array functions

✓ **map**

```
const arr = [1, 2, 3, 4, 5, 6];
```

↓

```
doubleArr = [2, 4, 6, 8, 10, 12];
```

```
const doubleArr = arr.map((n) => n*2);
```

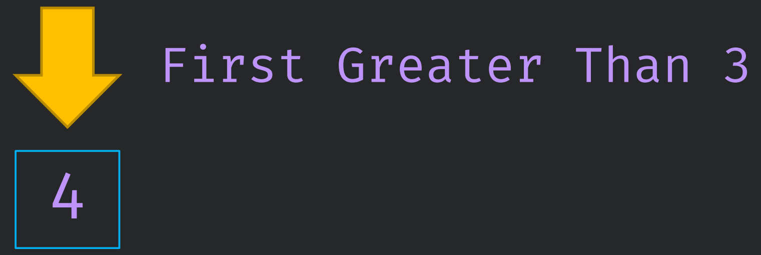
```
function doubleNumber(n){
    return n*2;
}
```

```
const doubleArr = [];
for (const n of arr) {
    doubleArr.push(doubleNumber(n));
}
```

# Callbacks with array functions

✓ **find**

```
const arr = [1, 2, 3, 4, 5, 6];
```



First Greater Than 3

4

```
const firstGreaterThan3 = arr.find((n) => n > 3);
```

```
function isGreaterThan3(n){
    return n>3;
}
```

```
let firstGreaterThan3;
for (const n of arr) {
    if(isGreaterThan3(n)){
        firstGreaterThan3 = n;
        break;
    }
}
```

# Callbacks with array functions

- ✓ **sort**

```
const arr = [1, 2, 3, 4, 5, 6];
```

↓

```
const arr = [1, 2, 3, 4, 5, 6];
```

```
arr.sort((a,b) => b-a);
```

# Refs

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array)

# JavaScript for Developers

## Callbacks | Utility Functions