

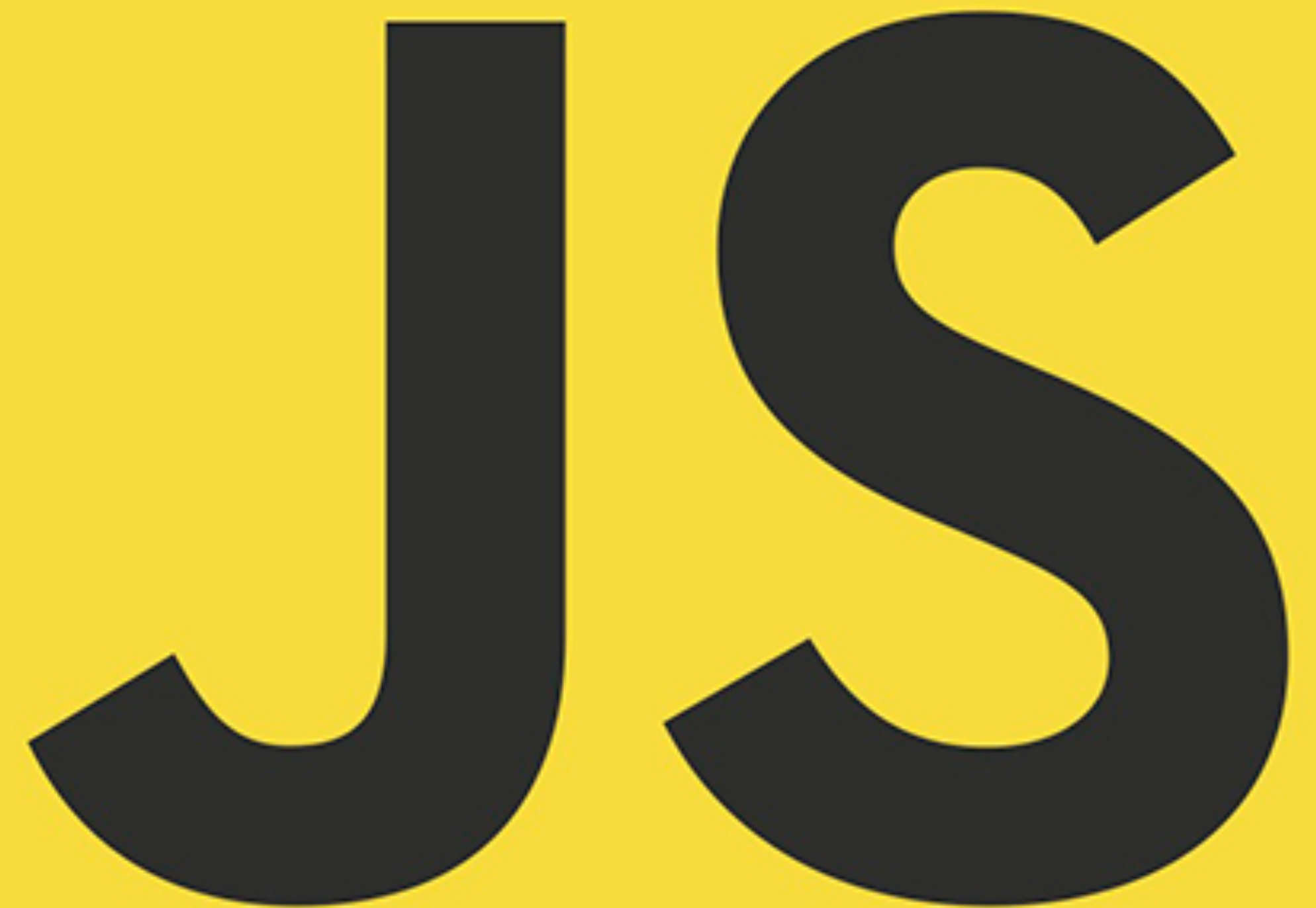
# Array - Manipulation

## Javascript I

# Today's Agenda

## JS - Array Manipulation

- Adding Elements
- Removing Elements
- Locating Elements
- Slicing
- Reversing
- Sorting
- Converting between strings and array
  - Splitting
  - Joining

A large, bold, dark gray 'JS' logo is centered on a bright yellow background. The letters are thick and have a slightly rounded, modern font style.

# Adding Elements

## Unshift()

- Adds the element at the **START** of an array.

### Basic Syntax:



```
arrayName.unshift(element1, element2, ...)
```

## push()

- Adds the element at the **END** of an array.

### Basic Syntax:



```
arrayName.push(element1, element2, ...)
```

# EXAMPLE

Unshift()



```
arrayName.unshift(element1, element2, ...)
```

push()



```
arrayName.push(element1, element2, ...)
```

# Removing Elements

shift()

- Removes the element at the **START** of an array.

Basic Syntax:



```
arrayName.shift()
```

pop()

- Removes the element at the **END** of an array.

Basic Syntax:



```
arrayName.pop( )
```

# EXAMPLE

shift()



```
arrayName.shift()
```

pop()



```
arrayName.pop()
```



# Locating Elements

indexOf()

- **Search** for the element starting at the **BEGINNING** of the array
- Returns index

Basic Syntax:



```
arrayName.indexOf(element);
```

lastIndexOf()

- **Search** for the element starting at the **END** of the array
- Returns index

Basic Syntax:



```
arrayName.lastIndexOf(element);
```

# EXAMPLE

`indexOf()` & `lastIndexOf()`





# Slicing

indexOf()

- Allows us to **copy** a portion of an array
- Returns a new array

Basic Syntax:



```
arrayName.slice(startIndex, endIndex);
```



# EXAMPLE

slice()





# Reverse

reverse()

- **Reverse** the array by making the last item first, and first item last
- Will modify the array

Basic Syntax:



```
arrayName.reverse( );
```



# EXAMPLE

reverse()

Reversible



# SORTING

sorting()

- **Sorting** is used when you want to sort an array in an alphabetically
- By default sort values as strings

Basic Syntax:



```
arrayName.sort();
```





# EXAMPLE

sort()



```
arrayName.sort( );
```

# Converting String to Arrays & Arrays to Strings

`split()`

`join()`

# split()



- Used to split a string into an array based on a specified delimiter
- Delimiters indicate where the string should split

Basic Syntax:



```
stringVariable.split('separator');
```



# EXAMPLE

split()

# join()

- Used to create an array from a string
- Can specify a separator that will be placed between elements

Basic Syntax:



```
arrayName.join(separator);
```



# EXAMPLE

join()

# Further Reading

# Exercise Time

## JavaScript I - Manipulation