Array Methods in JavaScript

Edition Series
Cute Kitty CheatSheets

Second Edition
PART I
MUTATING methods

Follow us meow! /kittycat kittycat-tech

Edition Series

Edition 1

Thank you
for loving
our work!

Edition 2

We are here,
meow!
Part 1 & 2

Edition 3

Coming soon,
meow!

Follow us meow! /kittycat kittycat-tech

Part 1
Mutating Array Methods!

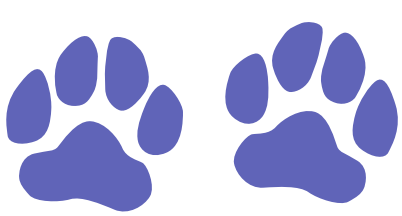
They **Mutate** the original array **right away,** when executed.

Strategic Thinking:
You can **study first these 9 mutating methods**, it will greatly help to :

1) **Avoid accidentally mutating** the original array that should not be mutated right away.

2) **Do easier 'mind map'** by classifying just a few 'mutating right away' ones,
then think of the rest as 'non-mutating right away'.

3) Give a signal that **these mutating methods need to be executed first,** prior logging them in console.log or prior returning the mutated values.



Follow us meow!

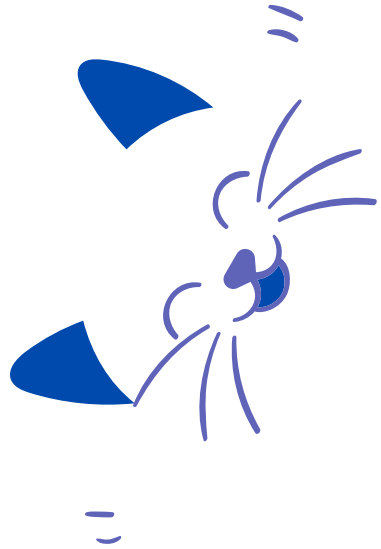
in/kittycat

kittycat-tech

References: https://developer.mozilla.org/ || MIT xPro Full Stack Course Modules, Prof. Zahid || https://betterprogramming.pub

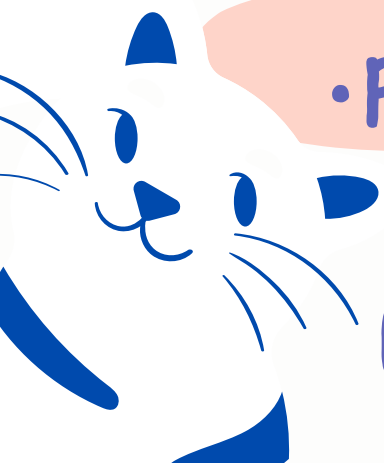
Part 1

Mutating



They **Mutate** the **original array right away** when executed

- .copyWithin()
- .reverse()
- .splice()
- .sort()
- .fill()
- .unshift()
- .pop()
- .shift()
- .push()



Follow us meow! in/kittycat

References: https://developer.mozilla.org/ || MIT xPro Full Stack Course Modules, Prof. Zahid || https://betterprogramming.pub

.copyWithin() method

MUTATING method

We use this to **copy** elements **from within** the array. It then **replaces** the pre-existing elements accordingly.

Arguments

1ST **First Argument** is the **index** of the element, that you want to access, to be replaced by the copied element.

2ND **Second Argument** is the **inclusive START index** which you want to start copying the elements

3RD **Third Argument** is the **exclusive END index** which you use as your stop point on copying more elements

```
> //Copying 1 element
  MeowArray = ["Coding","with","Meow", "is", "fun!","Cutie","Kittycat!"]
  MeowArray.copyWithin(2, 0, 1)
  console.log(MeowArray)

                                                                    VM2
  ▶ (7) ['Coding', 'with', 'Coding', 'is', 'fun!', 'Cutie', 'Kittycat!']
```

Follow us meow! /kittycat kittycat-tech

References: https://developer.mozilla.org/ || MIT xPro Full Stack Course Modules, Prof. Zahid || https://betterprogramming.pub

.copyWithin() method

MUTATING method

Multiple Elements:

As **more new elements** will be copied and added, they will **overtake and replace** the other **existing** elements. Be careful on this.

You can **copy then replace** as much as you want.

Multiple Elements Explanation:

1ST — **5** is the destination Index that you want to access for the copied elements to start doing consecutive replacements

2ND — **1** is the **inclusive START index** which you want to start copying the elements

3RD — **4** is the **exclusive END index** that you use as your stop point on copying more elements

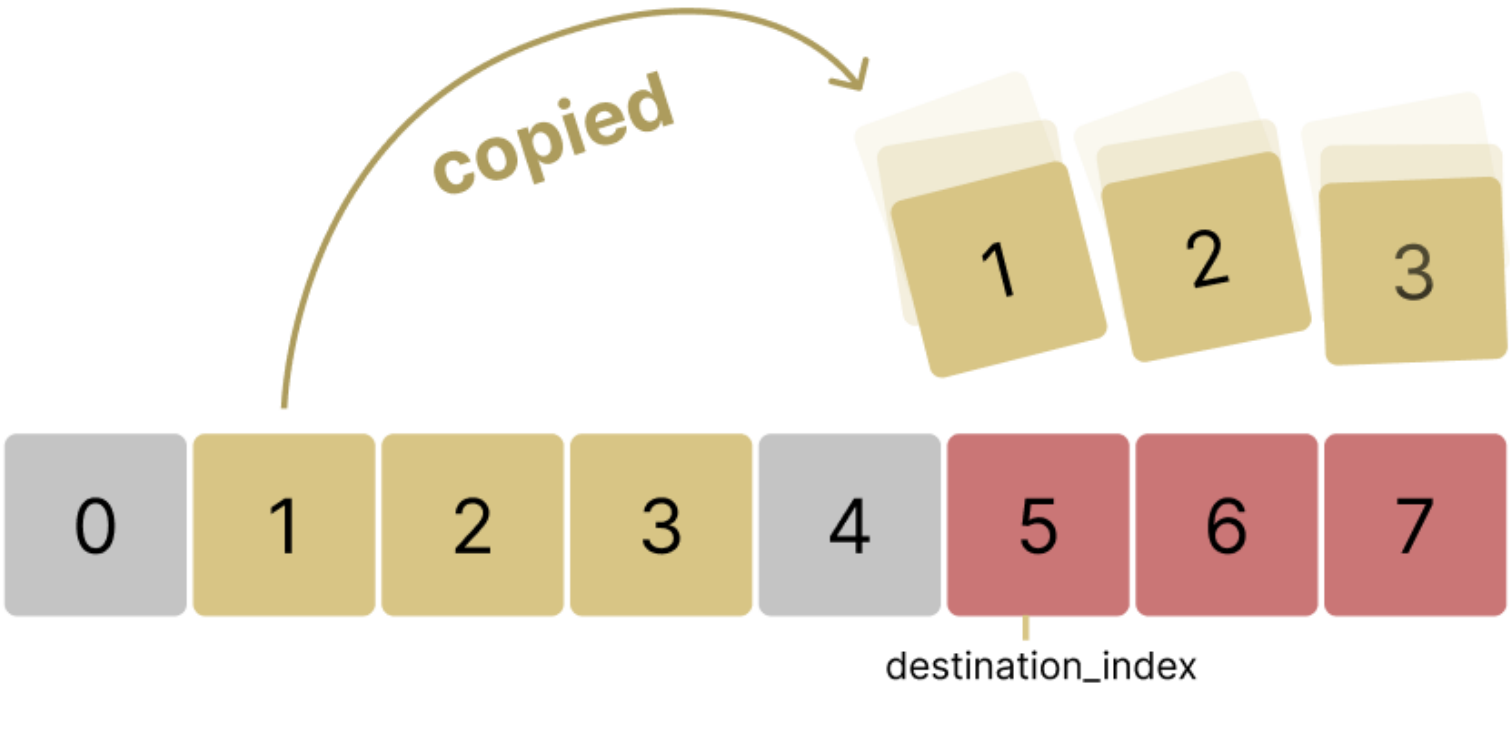

```
let MeowArray = ["Coding","with","Meow", "is", "fun!","Cutie","Kitty","Cat!"]
MeowArray.copyWithin(5,1,)
console.log(MeowArray)
```

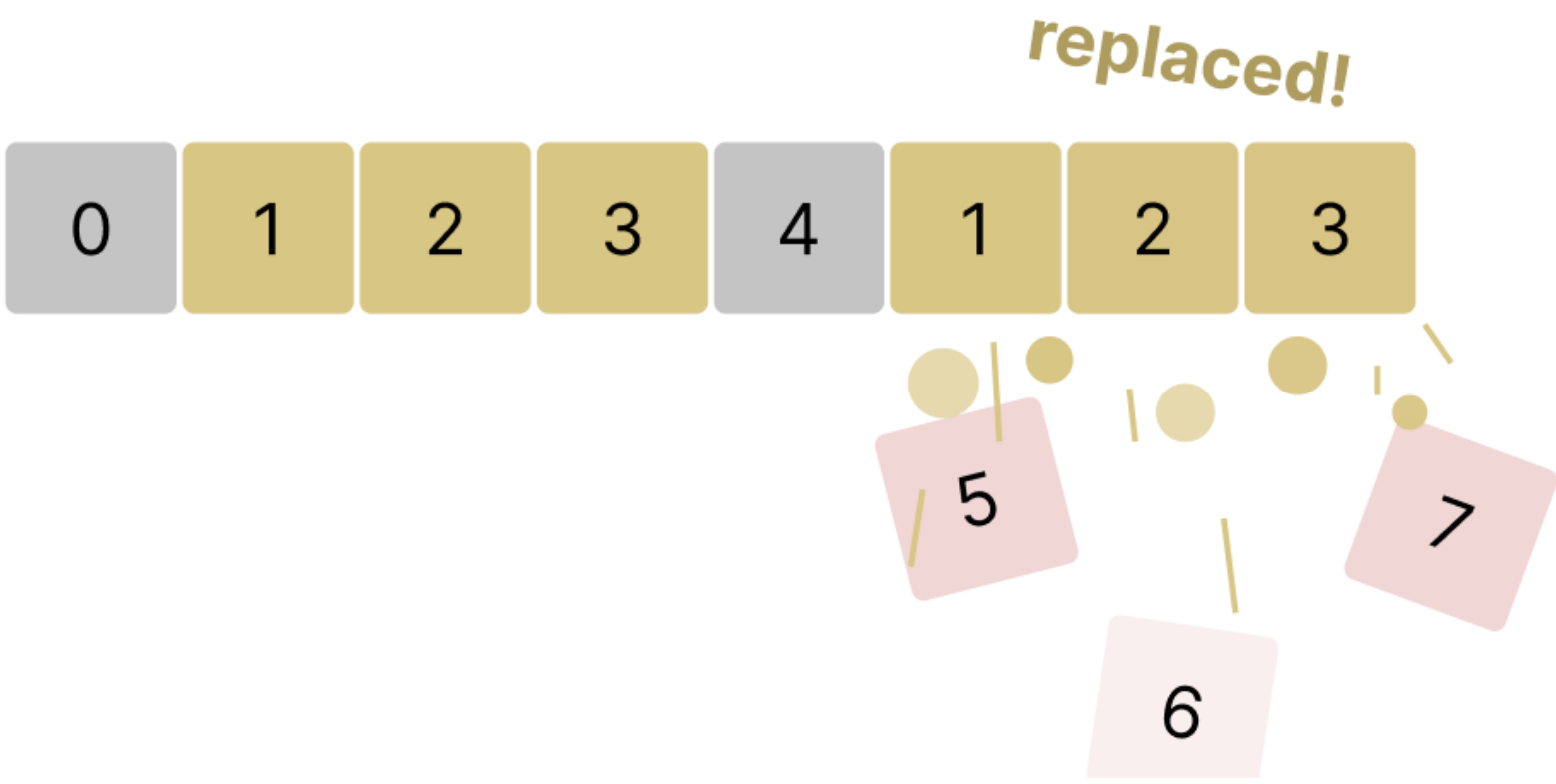
```
▼(8) ['Coding', 'with', 'Meow', 'is', 'fun!', 'with', 'Meow', 'is'] i
```

Follow us meow! in/kittycat kittycat-tech

References: https://developer.mozilla.org/ || MIT xPro Full Stack Course Modules, Prof. Zahid || https://betterprogramming.pub

.copyWithin(5,1,4)





Follow us meow!

/kittycat

kittycat-tech

Box Visuals Credit to: handsome Nikita Madebeykin

.splice method

MUTATING method

For Deleting Elements

Using 1 argument only:
We use this to **delete all** the pre-existing elements, starting from the index you provided.

Arguments

1ST

First Argument is the **index** of the element, that you want to be your **Start Index** of deletion. All other indexes after this index will be deleted together with their values.

```
let MeowArray = ["Coding","with","Meow", "is", "fun!","Cutie","Kittycat!"]
MeowArray.splice(1)
console.log(MeowArray)
```

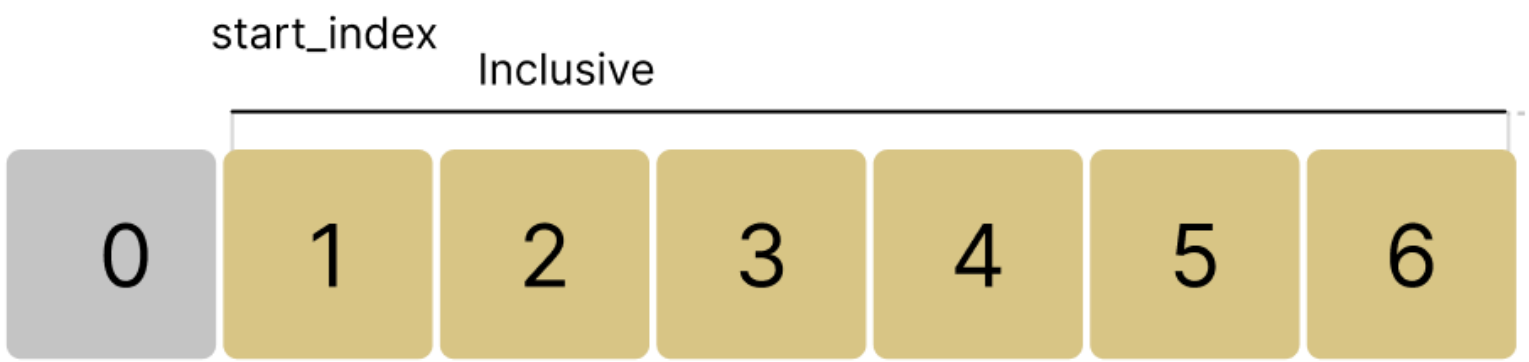
```
▶ ['Coding']
```

Follow us meow! /kittycat kittycat-tech

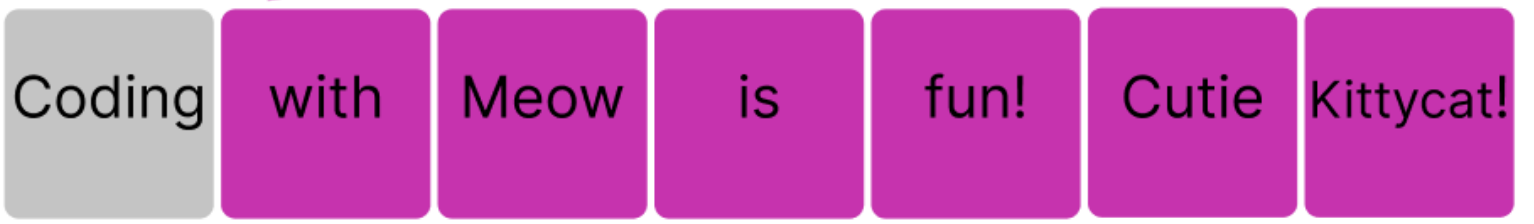
References: https://developer.mozilla.org/ || MIT xPro Full Stack Course Modules, Prof. Zahid || https://betterprogramming.pub || Marouane Souda

.splice(1)

.splice(start_index)



**From here,
Delete all elements**



deleted!

Follow us meow! /kittycat kittycat-tech

References: https://developer.mozilla.org/ || MIT xPro Full Stack Course Modules, Prof. Zahid || https://betterprogramming.pub

.splice method

MUTATING method

For Deleting Elements

Using 2 arguments:
We use this to **delete** **specific** pre-existing elements

Arguments

1ST

First Argument is the **index** of the element, that you want to access **for deletion.**
In case you would want to delete multiple indexes, it will be your **Start Index** amongst the other indexes.

2ND

From your Start Index, how many indexes do you want to delete in total?
That will be your **count** for your **Second Argument.**

```
let MeowArray = ["Coding","with","Meow", "is", "fun!","Cutie","Kittycat!"]
MeowArray.splice(3,2)
console.log(MeowArray)
```

```
▶ (5) ['Coding', 'with', 'Meow', 'Cutie', 'Kittycat!']
```

Follow us meow! /kittycat kittycat-tech

References: https://developer.mozilla.org/ || MIT xPro Full Stack Course Modules, Prof. Zahid || https://betterprogramming.pub

.splice(3,1)

.splice(start_index,count)

start_index

Inclusive

0	1	2	3	4	5	6

Delete this one only

Coding	with	Meow	is	fun!	Cutie	Kittycat!

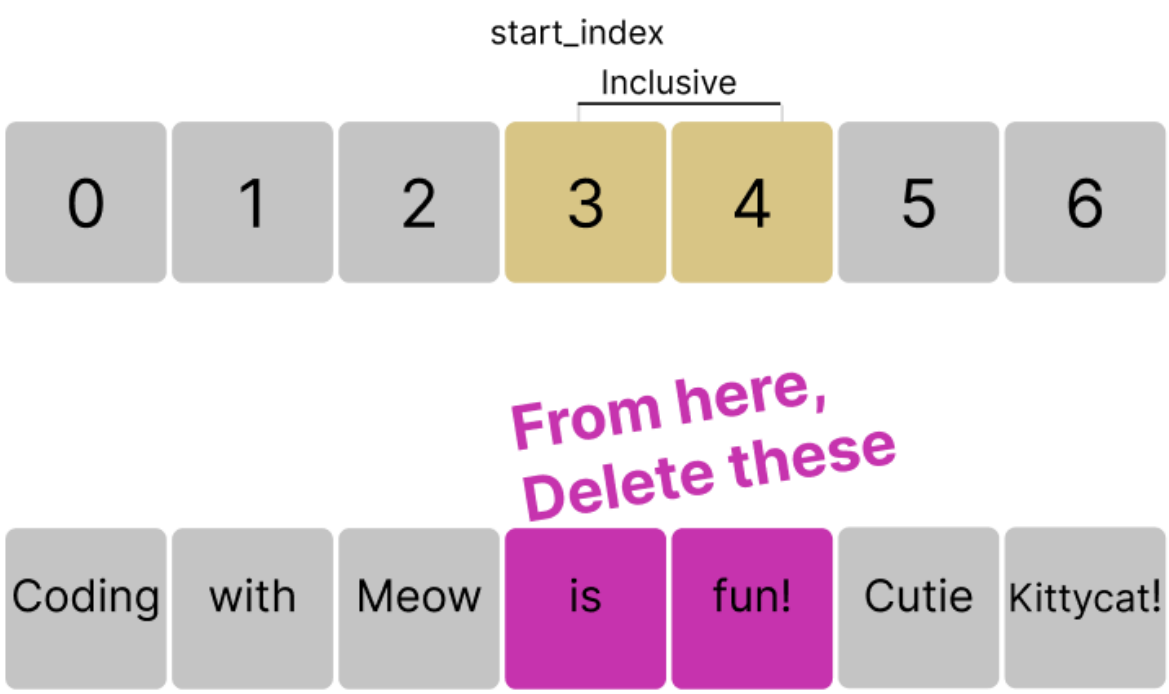
Coding	with	Meow	fun!	Cutie	Kittycat!

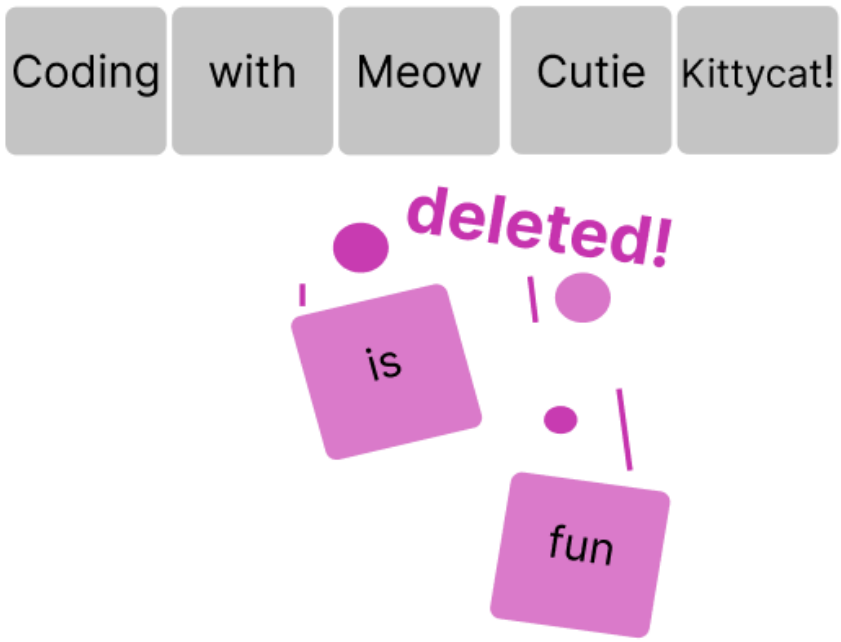
Follow us meow! /kittycat kittycat-tech

References: https://developer.mozilla.org/ || MIT xPro Full Stack Course Modules, Prof. Zahid || https://betterprogramming.pub

.splice(3,2)

.splice(start_index, count)





Follow us meow! /kittycat kittycat-tech

References: https://developer.mozilla.org/ || MIT xPro Full Stack Course Modules, Prof. Zahid || https://betterprogramming.pub

.splice method

MUTATING method

For Adding New Elements

3, 4, or more arguments:

We can use splice to **add** element/s **in the middle** or **anywhere** in the array. The new element/s just takes the index position and push the other pre-existing elements to the right.

You can add as much as you want using 3rd, 4th arguments and so fort.

```
> let MeowArray = ["Coding","with","Meow", "is", "fun!",]
  MeowArray.splice(4,0,"ultra","super")
  console.log(MeowArray)
  ▶ (7) ['Coding', 'with', 'Meow', 'is', 'ultra', 'super', 'fun!']
```

Next Page: Arguments explanation

References: https://developer.mozilla.org/ || MIT xPro Full Stack Course Modules, Prof. Zahid || https://betterprogramming.pub

Arguments: .splice(1st, 2nd, 3rd, 4th,..)

For Adding New Elements

1ST

First Argument is the **index** of the element, that you want to access **for adding/inserting new element.**

In case you would want to add more elements, it will be your **Start Index** where to start adding new elements.

2ND

From your Start Index, how many indexes do you want to delete in total?

Since you don't want to delete any elements in this case, Zero will be you **count** for your **Second Argument.**

3RD

Third Argument is **the element,** that you want to **ADD or Insert.**

This new element will be **inserted** on the **Start Index** of your first Argument. **Nothing gets deleted** in this process.

4TH

+

You can add your **4th argument** or more arguments as much as you want.

Remember that these will be the **new elements** that you want to **add in addition** to third Argument.

Follow us meow! /kittycat kittycat-tech

References: https://developer.mozilla.org/ || MIT xPro Full Stack Course Modules, Prof. Zahid || https://betterprogramming.pub

.splice(4,0,"ultra","super")

.splice(start_index, count, Add_new, Add_new)

start_index

0	1	2	3	4
Coding	with	Meow	is	fun!

ADD new elements

Coding	with	Meow	is	ultra	super

added!

Coding	with	Meow	is	ultra	super	fun!

new length: 7

0	1	2	3	4	5	6

Follow us meow! /kittycat kittycat-tech

References: https://developer.mozilla.org/ || MIT xPro Full Stack Course Modules, Prof. Zahid || https://betterprogramming.pub

.splice method

MUTATING method

For Deleting then Adding New Elements

3, 4, or more arguments:

We use this to **remove** the specific existing elements **then add new** elements declared in Argument 3 and so fort.

```
let MeowArray = ["Coding","with","Meow", "is", "fun!","Cutie","Kittycat!"]
MeowArray.splice(4,3,"full","of","cuteness","meow")
console.log(MeowArray)
```

```
▶ (8) ['Coding', 'with', 'Meow', 'is', 'full', 'of', 'cuteness', 'meow']
```

Next Page: Arguments explanation

Follow us meow! /kittycat kittycat-tech

References: https://developer.mozilla.org/ || MIT xPro Full Stack Course Modules, Prof. Zahid || https://betterprogramming.pub

Arguments: .splice(1st, 2nd, 3rd, 4th,..)

For Deleting then Adding New Elements

1ST

First Argument is the **index** of the element, that you want to access **for deletion.**
In case you would want to delete multiple indexes, it will be your **Start Index** amongst the other indexes.

2ND

From your Start Index, how many indexes do you want to delete in total?
That will be your **count** for your **Second Argument.**

3RD

Third Argument is **the element,** that you want to **ADD or Insert** to replace the removed element/s.
This new element will be **inserted** on the **Start Index** of your first Argument.

4TH

+

You can add your 4th argument or more arguments as much as you want.
Remember that these will be the **new elements** that you want to **add in addition** to third Argument.

Follow us meow! /kittycat kittycat-tech

References: https://developer.mozilla.org/ || MIT xPro Full Stack Course Modules, Prof. Zahid || https://betterprogramming.pub

.splice(4,3,"full","of","cuteness","meow")

Delete then Add, Meow!

.splice(start_index, count, Add_new, Add_new..)

start_index Inclusive

0	1	2	3	4	5	6

From here, Delete these 3

Coding	with	Meow	is	fun!	Cutie	Kittycat!

Coding	with	Meow	is

then ADD new elements

Coding	with	Meow	is	full	of	cuteness	meow

added!

new length: 8

0	1	2	3	4	5	6	7

Follow us meow! /kittycat kittycat-tech

References: https://developer.mozilla.org/ || MIT xPro Full Stack Course Modules, Prof. Zahid || https://betterprogramming.pub

.copyWithin() vs .splice()

Differences:

.copyWithin()

It **copies first from within,**
then the copied elements **replaces and deletes** the respective pre-existing Index elements. If 5 new elements are added, it means 5 elements from pre-existing indexes will be deleted. **The Array's length will always stay the same.**

Thoughts:
With 3 Arguments, this method allows you to specify the indexes that hold the elements you want to copy, but then the **arguments do not clearly state** which indexes (aside from Start Index) will be deleted by the copied elements.

.splice()

It **deletes first** the specific elements held by the indexes declared in the arguments 1&2, **then add new element/s** declared in the argument 3, 4, and so on, by simply inserting them in the array.

New added elements, if more than the number of deleted elements will just end up pushing the other pre-existing elements to the right. (They consume the needed indexes but do not delete the pre-existing values in contrast to .copyWithin() .)
The Array's length then becomes longer.

Thoughts:
The method we can use to **add elements anywhere in the array**.
Also **with 3 or more arguments,** this method is strict on which specific indexes should be first deleted, before you type for the new added elements.
No copying! :)

Follow us meow! /kittycat kittycat-tech

References: https://developer.mozilla.org/ || MIT xPro Full Stack Course Modules, Prof. Zahid || https://betterprogramming.pub

.fill method

MUTATING method

We use this to **fill in** new element/s, **replacing** selected or all values in the Array based on declared argument/s.

1 Argument

```
let MeowArray = ["Coding","with","Meow", "is", "fun!","Cutie","Kitty","Cat!"]
MeowArray.fill("hey")
console.log(MeowArray)
```

```
▶ (8) ['hey', 'hey', 'hey', 'hey', 'hey', 'hey', 'hey', 'hey']
```

3 Arguments

```
> let MeowArray = ["Coding","with","Meow", "is", "fun!","Cutie","Kitty","Cat!"]
  MeowArray.fill("hey",1,4)
  console.log(MeowArray)
```

```
  ▶ (8) ['Coding', 'hey', 'hey', 'hey', 'fun!', 'Cutie', 'Kitty', 'Cat!']
```

Next Page: Arguments explanation

Follow us meow! /kittycat kittycat-tech

References: https://developer.mozilla.org/ || MIT xPro Full Stack Course Modules, Prof. Zahid || https://betterprogramming.pub

3 Arguments: .fill (1st, 2nd, 3rd)

For Filling New Elements

1ST

First Argument is the value or **new element** that you want to fill in the array.

It will replace the pre-existing elements and will delete them in the process.

2ND

Second Argument is the **inclusive Start Index** that you want to be replaced by the new elements

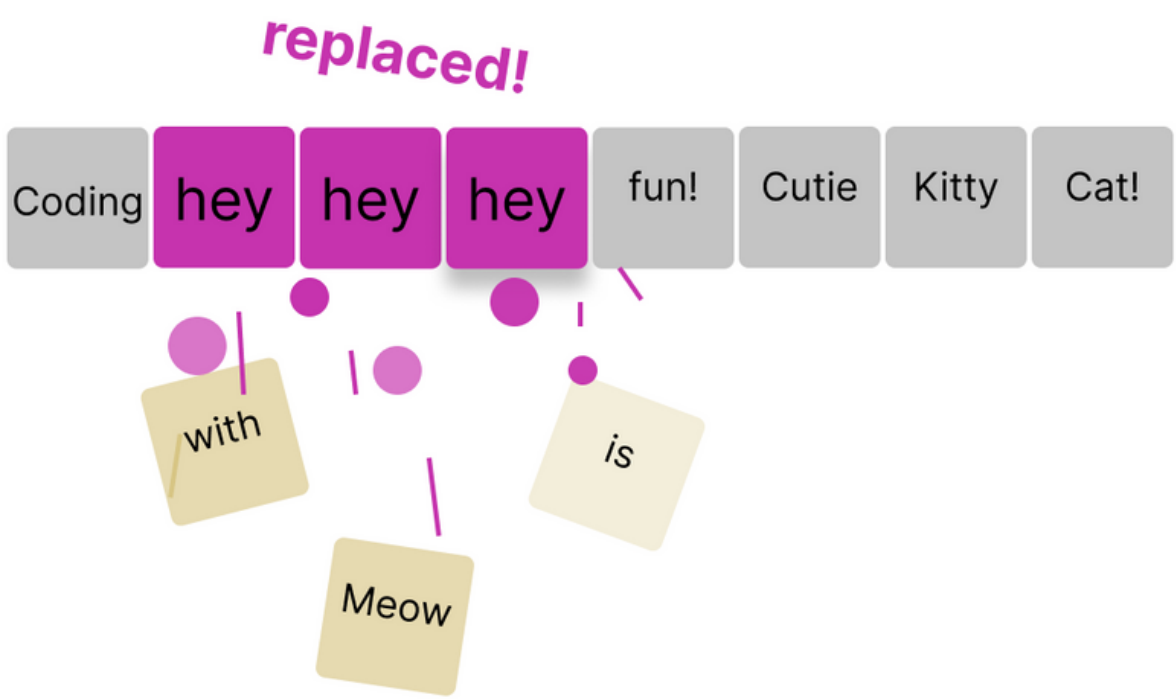
3RD

3rd Argument is the **exclusive END index** that you use as your stop point on filling more elements

Follow us meow! /kittycat kittycat-tech

References: https://developer.mozilla.org/ || MIT xPro Full Stack Course Modules, Prof. Zahid || https://betterprogramming.pub

.fill("hey",1,4)



Follow us meow! /kittycat kittycat-tech

References: https://developer.mozilla.org/ || MIT xPro Full Stack Course Modules, Prof. Zahid || https://betterprogramming.pub

.pop() method

MUTATING method

We use this to **remove** the **last element** of an array

```
> let MeowArray = ['Learning', 'is','fun'];
  MeowArray.pop();
  console.log(MeowArray);
  ▶ (2) ['Learning', 'is']
```

Follow us meow! /kittycat kittycat-tech

References: https://developer.mozilla.org/ || MIT xPro Full Stack Course Modules, Prof. Zahid || https://betterprogramming.pub

.push() method

MUTATING method

We use this to **add** an element **at the end** of an array

```
let MeowArray = ['Learning', 'is','fun'];
MeowArray.push('Meow!');
console.log(MeowArray);
```

```
▼ (4) ['Learning', 'is', 'fun', 'Meow!']
```

Follow us meow! /kittycat kittycat-tech

References: https://developer.mozilla.org/ || MIT xPro Full Stack Course Modules, Prof. Zahid || https://betterprogramming.pub

.reverse() method

MUTATING method

We use this to **reverse** the elements in an array

```
let MeowArray = ['Learning', 'is','fun'];
console.log(MeowArray.reverse());
```

```
▶ (3) ['fun', 'is', 'Learning']
```

Follow us meow! /kittycat kittycat-tech

References: https://developer.mozilla.org/ || MIT xPro Full Stack Course Modules, Prof. Zahid || https://betterprogramming.pub

.sort() method

MUTATING method

We use this to **sort** the elements of an array.

```
> let MeowArray = ['Learning', 'is','fun'];
  console.log(MeowArray.sort());
  ▶ (3) ['Learning', 'fun', 'is']
```

Note:
An exception comes when the array has numbers wrapped in string,
Sort will then need to be executed with a function.
On upcoming Edition 3, we will study this topic deeper, with functions

Follow us meow! **/kittycat** **kittycat-tech**

References: https://developer.mozilla.org/ || MIT xPro Full Stack Course Modules, Prof. Zahid || https://betterprogramming.pub || Faiz Ahmed Rizvi

.shift() method

MUTATING method

We use this to **remove** an element **at the beginning** of an array

```
> let MeowArray = ['Learning', 'is','fun'];
  MeowArray.shift();
  console.log(MeowArray);
  ▶ (2) ['is', 'fun']
```

Follow us meow! /kittycat kittycat-tech

References: https://developer.mozilla.org/ || MIT xPro Full Stack Course Modules, Prof. Zahid || https://betterprogramming.pub

.unshift() method

MUTATING method

We use this to **add** an element
at the beginning of an array

```
> let MeowArray = ['Learning', 'is','fun'];
  MeowArray.unshift('Wow!');
  console.log(MeowArray);
  ▶ (4) ['Wow!', 'Learning', 'is', 'fun']
```

Follow us meow! /kittycat kittycat-tech

References: https://developer.mozilla.org/ || MIT xPro Full Stack Course Modules, Prof. Zahid || https://betterprogramming.pub

Upcoming 3rd Edition

Message to our Meow Friends:

Our **3rd Edition** for Array Methods in JavaScript would be filled with a bit of advanced JS: combinations of multiple topics.

And of course, **more** daily dose of **cutie patootie cats** for us! **Meow!**

Inclusions

01 JavaScript Array topics will be executed with Functions, Loops and more!

02 Much deeper understanding of Array Methods in JavaScript

03 It will include the usage of the codes in real life scenarios.

04 A lot more cutie patootie cats for you!

It could be your ultimate (cuteness overload) Kittycat Cheatsheets!

Follow us meow! /kittycat kittycat-tech

Follow us meow! /kittycat kittycat-tech

What do Edition Series mean, meow?

First Edition

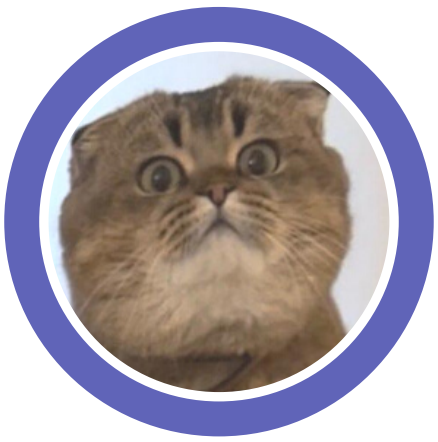
The most basic version for newborn babies in programming.

The goal of this edition is for the newbies to have a smooth onboarding and to digest easier the basic concepts.

Second Edition (We are here, meow)

This is the material is for those who have already grasped their knowledge and just got on board for programming.

It consists of basics and a bit more foundation than First Edition, usually with a step by step thinking method.



Third Edition

This is for a bit advanced version, for beginner level.
Usually a combination of multiple topics,
a deeper understanding of the topics together
and a real-life usage of the concepts.

Website: kittycat.tech (coming soon)

Hi amazing you!
Let's be friends!
Add Meow!

Sweet and friendly

(I code randomly with paws)

/kittycat

kittycat-tech

kittycat.tech

Website: kittycat.tech (coming soon)