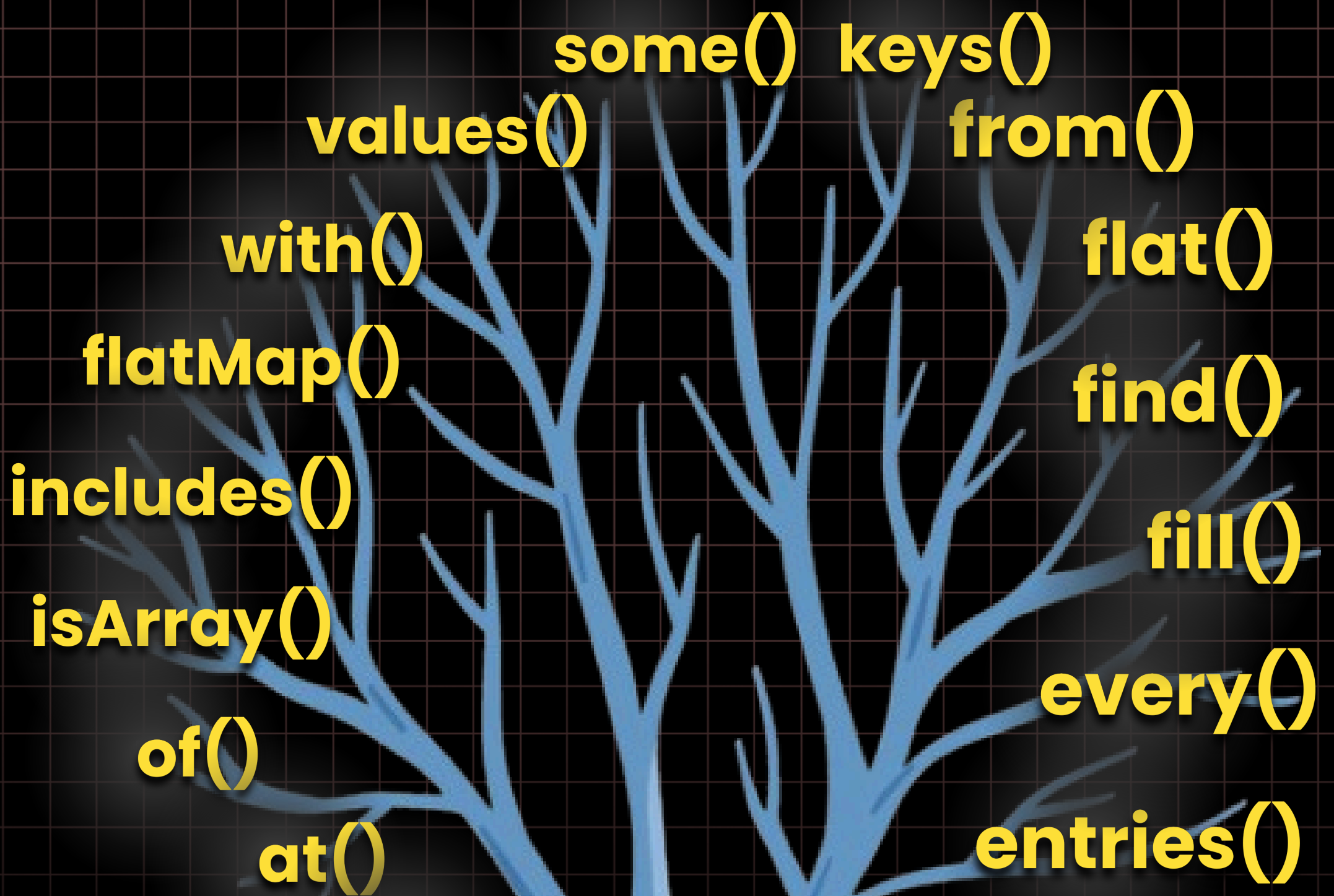


JAVASCRIPT **ARRAY**

METHODS (Part 3)



1. at

The `at()` method **returns** an **element** from an array at specified **index**.

It makes it **easier** to **get** the **last element** using `array.at(-1)`

The `at` method **works** on both **arrays** and **strings**

`array.at(index)`



Talha shiekh
@talhashiekh



Save

2. of

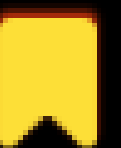
The `Array.of()` method **creates** a new **array** from any number of specified **arguments**

The `Array.of()` method can **take** **any type** of arguments

```
Array.of(elem1, elem2, ... , elemN)
```



Talha shiekh
@talhashiekh



Save

3. isArray

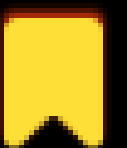
The `isArray()` method returns **true** if given **value** is an **array**, otherwise **false**.

It helps **distinguish** arrays from other **objects**

`Array.isArray(obj)`



Talha shiekh
@talhashiekh



Save

4. includes

The `includes()` method returns **true** if an array **contains** a specified **value**

The `includes()` method **returns false** if the value is **not found**

The `includes()` method is **case sensitive**

```
array.includes(element, start)
```



Talha shiekh
@talhashiekh



Save

5. flatMap

The flatMap() method **maps** all array **elements** and creates a **new flat array**

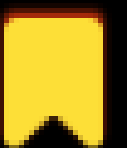
flatMap() does **not execute** the function for **empty** elements

flatMap() does **not change** the **original** array.

```
arr.flatMap(fun(currVal, ind, arr), thisVal)
```



Talha shiekh
@talhashiekh



Save

6. with

The with() method **updates** a **specified** array **element**.

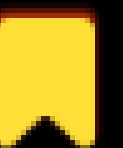
The with() method **returns** a new **array**

The with() method does **not** **change** the original **array**

```
array.with(index, value)
```



Talha shiekh
@talhashiekh



Save

7. values

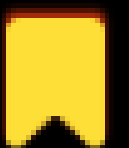
The values() method **returns** an **Iterator** containing the **values** of an **array**

The values() method does **not** **change** the **original** array

```
array.values()
```



Talha shiekh
@talhashiekh



Save

8. some

The `some()` method **checks** if at least **one element** in an array **meets** a specified **condition**

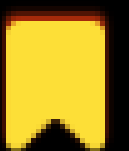
It returns **true** if any element **passes** the test otherwise **false**

It does **not execute** the function for **empty** elements

```
arr.some(fun(val, ind, arr), this)
```



Talha shiekh
@talhashiekh



Save

9. keys

The `keys()` method **returns** an **Iterator** containing the **keys** of an **array**

The `keys()` method does **not** **change** the **original** array

```
array.keys()
```



Talha shiekh
@talhashiekh



Save

10. from

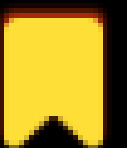
The `Array.from()` method **returns** an **array** from any **object** with a `length` property

It can also **take** a mapping **function** to **modify** elements while **creating** the array.

```
Array.from(object, mapFun, thisVal)
```



Talha shiekh
@talhashiekh



Save

11. flat

The `flat()` method **flattens** nested **arrays** into a single array up to a **specified depth** value.

We can use **`flat(Infinity)`** to completely **flatten** a deeply **nested array**

```
array.flat(depth)
```



Talha shiekh
@talhashiekh



Save

12. find

The find() method **returns** the value **first** element that **meets** a given **condition**

The find() method does **not** **execute** the function **for empty** elements

The find() method **returns** **undefined** if **no** elements are **found**.

```
arr.find(fun(currVal, ind, arr), thisVal)
```



Talha shiekh
@talhashiekh



Save

13. fill

The fill() method **fills** all or specified **elements** in an array with a **value**

The fill() method **overwrites** the **original** array

```
array.fill(value, start, end)
```



Talha shiekh
@talhashiekh



Save

14. every

The `every()` method **checks** if all **elements** in an array **meets** a specified **condition**

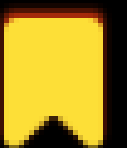
It returns **true** if all elements **pass** the test otherwise **false**

It does **not execute** the function for **empty** elements

```
arr.every(fun(currVal, ind, arr), thisVal)
```



Talha shiekh
@talhashiekh



Save

15. entries

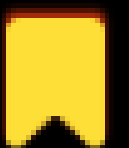
The `entries()` method **returns** an **Iterator** containing the **key value pairs** from an **array**

The `entries()` method does **not** **change** the **original** array

```
array.entries()
```



Talha shiekh
@talhashiekh



Save

KEEP EXPLORING JAVASCRIPT WITH US

Dont forget to follow us on



@talhashiekh



@codepalace



@shiekhuuu.dev



Links in the comments

