@startwithmani

@M.serisha Kothapalli

Map Vs Objects in JS

swipe

@startwithmani

@M.serisha Kothapalli

Quick Intro To Map and Object

JavaScript **map** and **object** is both data structures with similar concepts then why two?? As expected there are subtle differences between them

In JavaScript, objects are handy. They allow us to easily group multiple pieces of data together. **After ES6**, we got a new addition to the language - **Map**

lets check few **differences** between them

swipe

Flexibility of Key types:

- Key type of an object can be only either string or a symbol.
- Map allows keys of any type.It can be functions, objects, or primitive

Objects

```
let obj ={

    1:'hello'
}

obj['1']   // hello
obj[1]    //hello
```

Map

```
let map = new Map([
  ['1',  'hello'],
  [1,    'Bye']
]);

map.get('1') //hello
map.get(1)   //Bye
```

To conclude - remember the regular **Object** would **convert keys to string** . whereas **Map** keeps the type, so these two are **different**

swipe

Iteration

- Objects are not directly iterable and needs Object.keys or Object.entries methods.

```
for (const [key, value] of Object.entries(obj)) {....}

Object.entries(obj).forEach(([key, value]) => {....}
```

- Map has a built-in forEach method, similar to Array.

```
map.forEach(([key, value]) => {....}
```

swipe

@startwithmani

@M.serisha Kothapalli

Ways of creating

- There are many ways to create an object.

```
. Using Object literal  -> {}
. Using constructor method  -> new Object()
. using Object.create method
```

- Map has only one way of creation.

```
. Using constructor method -> new Map()
```

swipe

@startwithmani

@M.serisha Kothapalli

Getting & Setting properties

- Below are ways to access objects

```
. Dot property accessor: object. property.
. Square brackets property access: object['property']
. Object destructuring: const { property } = object.
```

- Below are ways to access map

```
. Using get and set method to access the properties and set the
  properties.

. You can't destructure Map directly, you will have to convert to
  object first.
```

swipe

Check if key exists: To check if a particular key already exists.

- Using "in" operator

```
let obj = {'name' : 'mani'}
if(name in obj){// code to be executed}
```

- Using inbuilt **has()** method

```
let map = new Map([ ['name',  'hello'] ]);
if(map.has('name')){// code to be executed}
```

swipe

Getting size

- To find the object size we need to depend on keys or values or entries method

```
Object.keys(obj).length
```

- Map has an inbuilt size property to caluculate total items

```
map.size
```

Support of JSON

- JSON gives direct support for object but not with map.

NOTE

❌ Although we can set map[key] = 2, **Never ever do this** as this is treating map as a plain JavaScript object, so it ends up behaving as a regular Object.

❌ **Map** has dedicated methods like **set, get** etc.....Always ensure to use them.

swipe