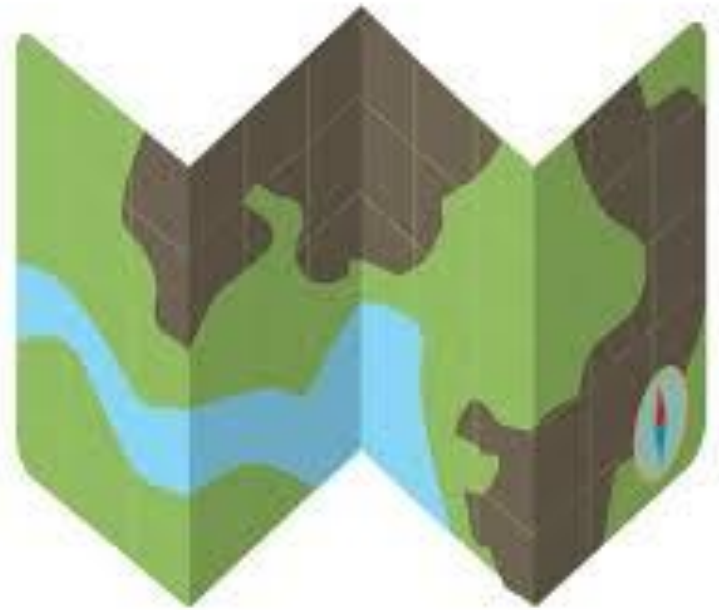


JAVASCRIPT MAP

MET CS
Andrew Sheehan

WHAT IS A MAP?



Map is a built-in
ES6 data type that offers a
key, value pairing

CANIUSE?

IE	Edge *	Firefox	Chrome	Safari	Opera	Safari on iOS *	Opera Mini *	Android Browser *
6-10	12-91	2-90	4-91	3.1-14	10-77	3.2-14.4		2.1-4.4.4
11	92	91	92	14.1	78	14.7	all	92
		92-93	93-95	15-TP				

WHAT IS ORDER MATTERS



The keys in Map are ordered.

When iterating, a Map object
returns keys in order of insertion

BUILT IN PART OF THE CORE

You can use any valid
Javascript data type as **key** or
as value.



DATA TYPE

EXAMPLE OF A MAP

Target a key.



Aug

KEYS	VALUES
Jan	327.2
Feb	368.2
Mar	197.6
Apr	178.4
May	100.0
Jun	69.9
Jul	32.3
Aug	37.3
Sep	19.0
Oct	37.0
Nov	73.2
Dec	110.9
Annual	1551.0

Get the value.



37.3

MAP

PURPOSE

Designed as an alternative to using Object Literals for storing key, value pairs.

```
// Example of an Object Literal
const Car = {
  color: Color.UNKNOWN,
  height: 66.2,
  averageWeight: "981 lbs."
}
```

USING MAPS

INSTANTIATION

// Must use the new keyword

```
const nameTolds = new Map();
```



MAP:

ADDING TO MAP

```
const IdToName =  
    new Map();
```

```
IdToName.set("U-13-32", // key  
             "John"); // value
```

MAP: REMOVING FROM MAP

```
IdToName.delete(// key“U-13-32”);
```

MAP: DOES KEY EXIST?

```
if ( IdToName.// keyhas("U-13-32") ) {  
    // logic goes here...  
}
```

MAP: TO DELETE ALL KEYS/VALUES

```
// All key/values are deleted  
IdToName.clear();
```

MAP: RETRIEVING THE SIZE

```
If ( IdToName.size > 0) {  
    // logic goes here  
}
```

MAP: INITIALIZE::EXAMPLES

```
const kvArray = [['key1', 'value1'], ['key2', 'value2']]
```

```
// Use the regular Map constructor to transform a 2D key-value Array into a map  
const myMap = new Map(kvArray)
```

MAP ITERATING

```
IdToName.forEach(element => {  
    // logic goes here.  
});
```

USING MAPS

ITERATING – ANOTHER WAY

```
for (const key of m.keys() ) {  
    // logic goes here.  
}
```


USING MAPS

ONLY THE KEYS

```
IdToName.keys().forEach(key  
=> {  
    // logic goes here.  
});
```

USING MAPS

ONLY THE VALUES

IdToName

```
.values()
```

```
.forEach(value => {
```

```
    // logic goes here.
```

```
});
```

WEAKMAP

IT IS DIFFERENT...

Items are never garbage collected in a standard Map.

Every key of a [WeakMap](#) is an object.

When the reference to this object is lost, the value can be garbage collected

WEAKMAP

IT IS DIFFERENT...

The main differences

- a. Cannot iterate over the keys or values (or key-values) of a WeakMap
- b. Cannot clear all items from a WeakMap
- c. Cannot check the size