WORKING WITH ARRAYS

THE COMPLETE JAVASCRIPT COURSE

FROM ZERO TO EXPERT!

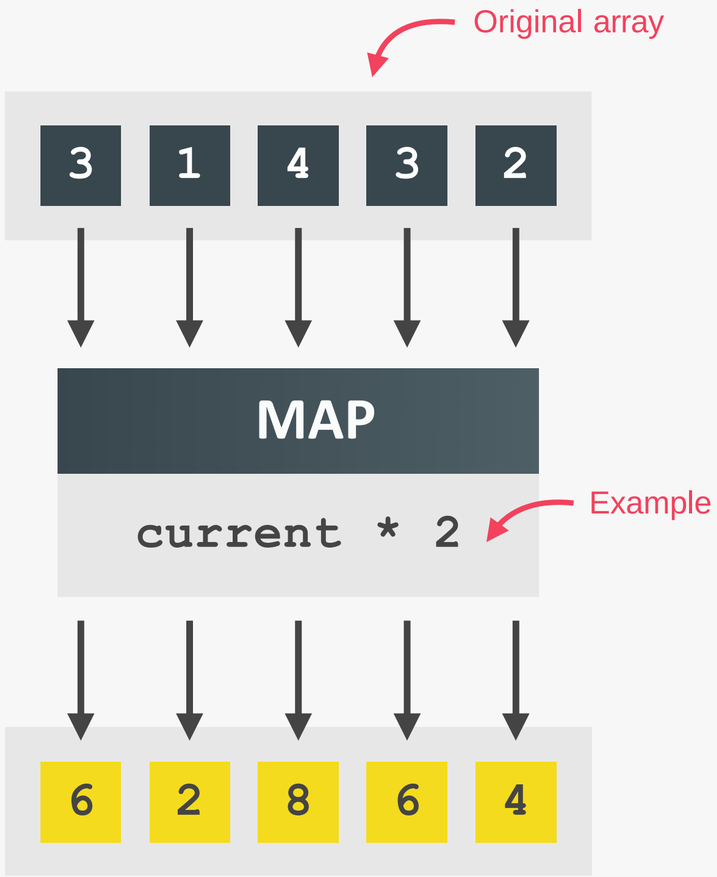
SECTION

WORKING WITH ARRAYS

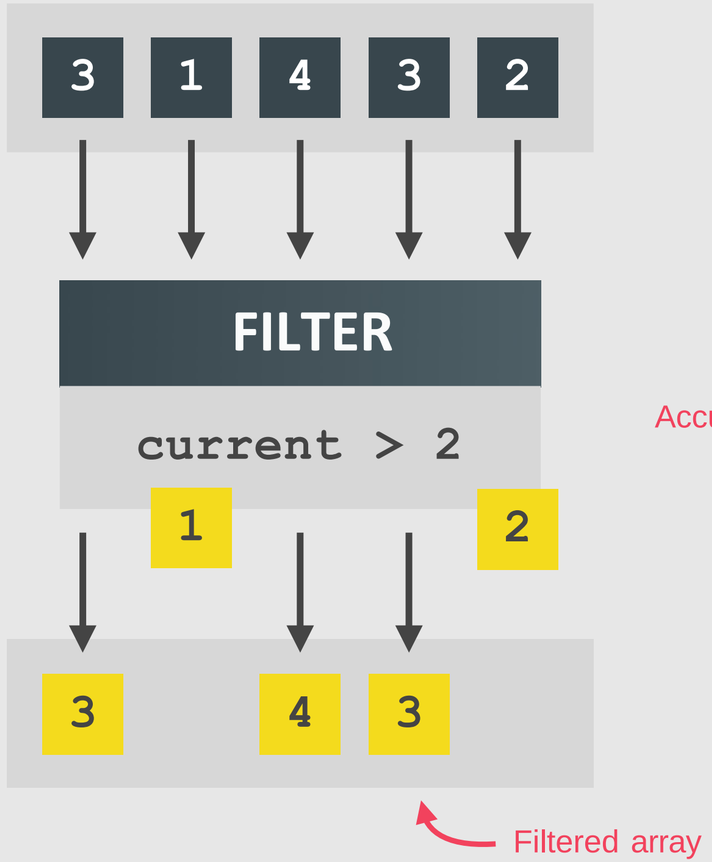
LECTURE

DATA TRANSFORMATIONS:
MAP, FILTER, REDUCE

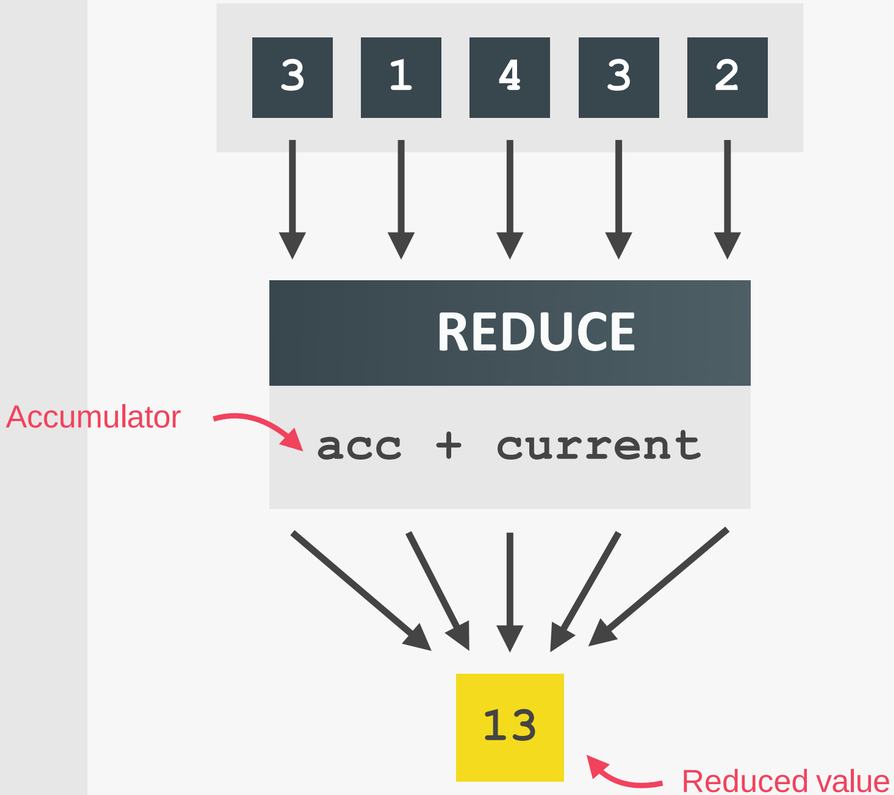
DATA TRANSFORMATIONS WITH MAP, FILTER AND REDUCE



👉 `map` returns a **new array** containing the results of applying an operation on all original array elements



👉 `filter` returns a **new array** containing the array elements that passed a specified **test condition**



👉 `reduce` boils (“reduces”) all array elements down to one single value (e.g. adding all elements together)

THE COMPLETE JAVASCRIPT COURSE

FROM ZERO TO EXPERT!

SECTION

WORKING WITH ARRAYS

LECTURE

SUMMARY: WHICH ARRAY METHOD TO USE?

WHICH ARRAY METHOD TO 🤔 USE?

"I WANT...:"

To mutate original array

👉 Add to original:

- `.push` *(end)*
- `.unshift` *(start)*

👉 Remove from original:

- `.pop` *(end)*
- `.shift` *(start)*
- `.splice` *(any)*

👉 Others:

- `.reverse`
- `.sort`
- `.fill`

A new array

👉 Computed from original:

- `.map` *(loop)*

👉 Filtered using condition:

- `.filter`

👉 Portion of original:

- `.slice`

👉 Adding original to other:

- `.concat`

👉 Flattening the original:

- `.flat`
- `.flatMap`

An array index

👉 Based on value:

- `.indexOf`

👉 Based on test condition:

- `.findIndex`

An array element

👉 Based on test condition:

- `.find`

Know if array includes

👉 Based on value:

- `.includes`

👉 Based on test condition:

- `.some`
- `.every`

A new string

👉 Based on separator string:

- `.join`

To transform to value

👉 Based on accumulator:

- `.reduce`

(Boil down array to single value of any type: number, string, boolean, or even new array or object)

To just loop array

👉 Based on callback:

- `.forEach`

(Does not create a new array, just loops over it)

ADVANCED DOM AND EVENTS

THE COMPLETE JAVASCRIPT COURSE

FROM ZERO TO EXPERT!

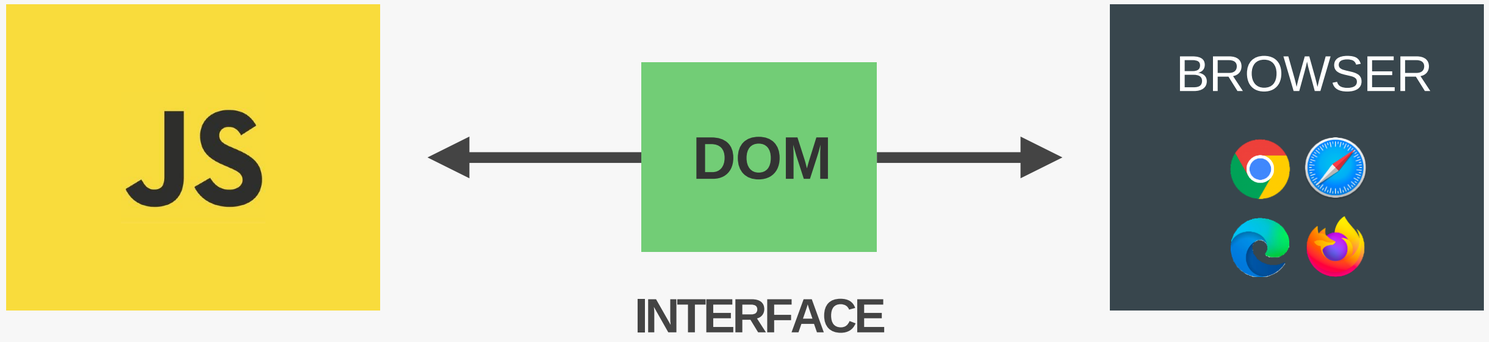
SECTION

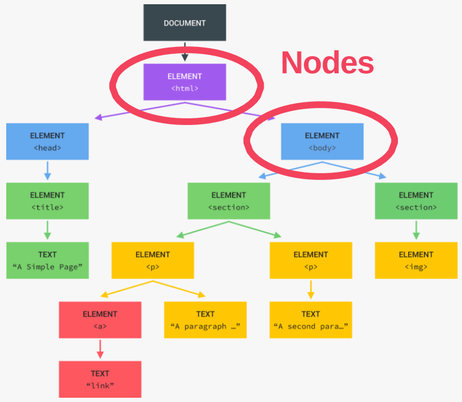
ADVANCED DOM AND EVENTS

LECTURE

HOW THE DOM REALLY WORKS

REVIEW: WHAT IS THE DOM?



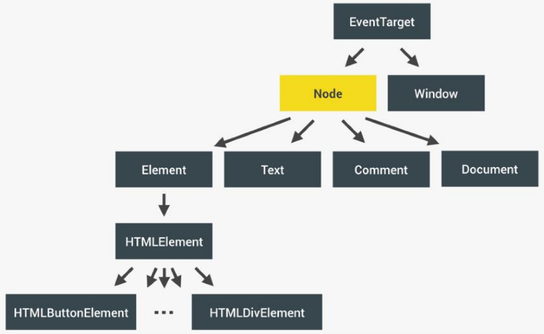


DOM tree

- Allows us to make JavaScript interact with the browser;
- We can write JavaScript to create, modify and delete HTML elements; set styles, classes and attributes; and listen and respond to events;
- DOM tree is generated from an HTML document, which we can then interact with;
- DOM is a very complex API that contains lots of methods and properties to interact with the DOM tree

Application Programming Interface

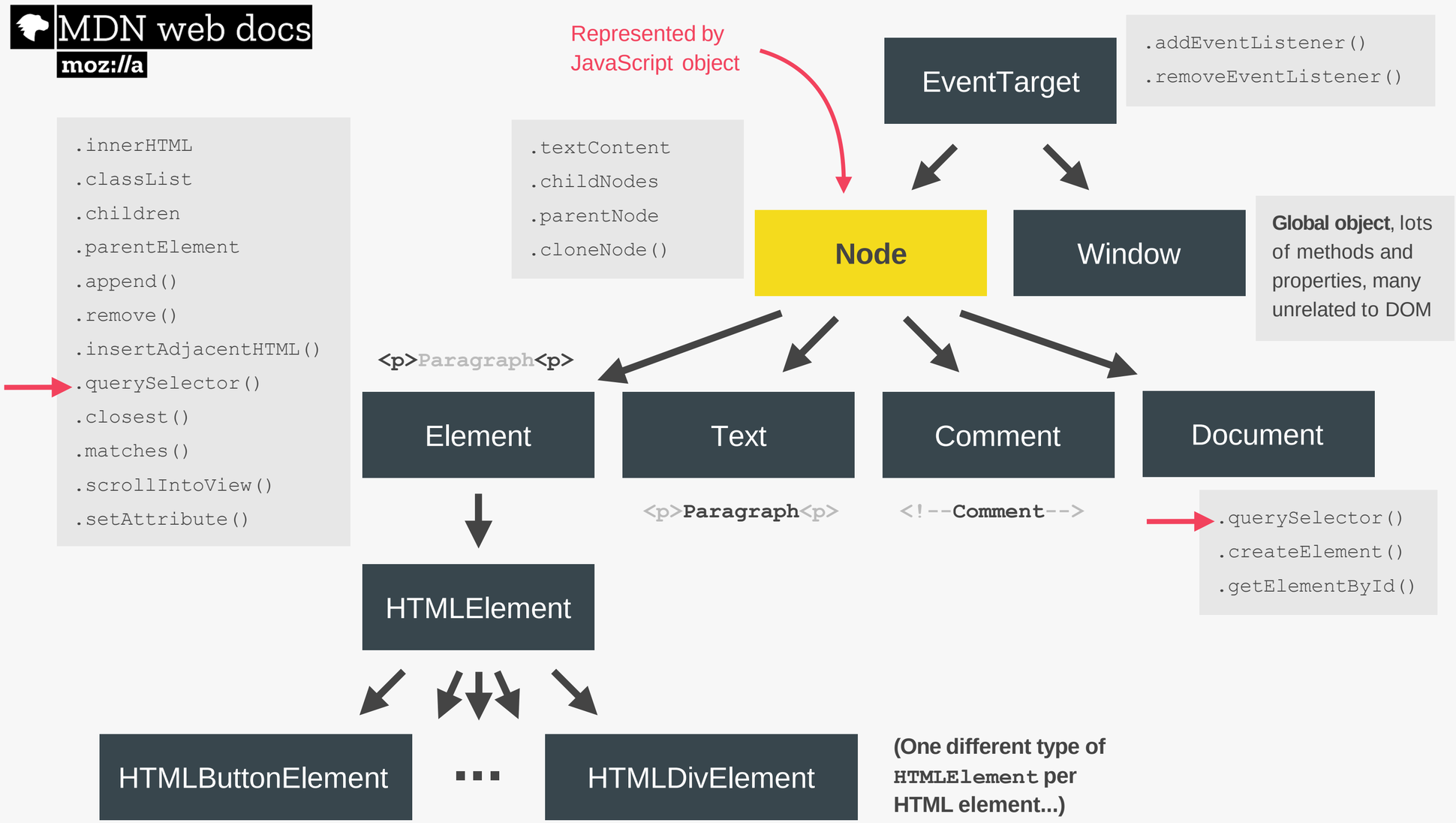
```
.querySelector() / .addEventListener() / .createElement() /
.innerHTML / .textContent / .children / etc ...
```



"Types" of DOM objects (next slide)

HOW THE DOM API IS ORGANIZED BEHIND THE SCENES

MDN web docs
moz://a



INHERITANCE OF METHODS AND PROPERTIES

Example:

Any `HTMLElement` will have access to `.addEventListener()`, `.cloneNode()` or `.closest()` methods.

(THIS IS NOT A DOM TREE)