Es.Next
Lecture 1

By Mona Soliman

Agenda

- Introduction about ES.Next
- Const & let
- New String Methods(startswith , endswith , includes)
- Template Literals (Template Strings)
- Arrow function
- Destructuring
- Rest parameter , Spread Operator
- Default Parameters
- New Array Methods (filter , find , findIndex , map)
- For of loop

Const & let

-`*let*` is a signal that the variable may be reassigned, such as a counter in a loop, or a value swap in an algorithm. It also signals that the variable will be used only in the block it’s defined in, which is not always the entire containing function.

-`*const*` is a signal that the identifier won’t be reassigned

New String Methods

-**startsWith()** returns a Boolean value—true or false—depending on whether the string starts with the substring you have specified.

-**endsWith()** is the opposite of the startsWith() method. The endsWith() method determines if a string ends with a particular set of characters.

-**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.

Template Literals

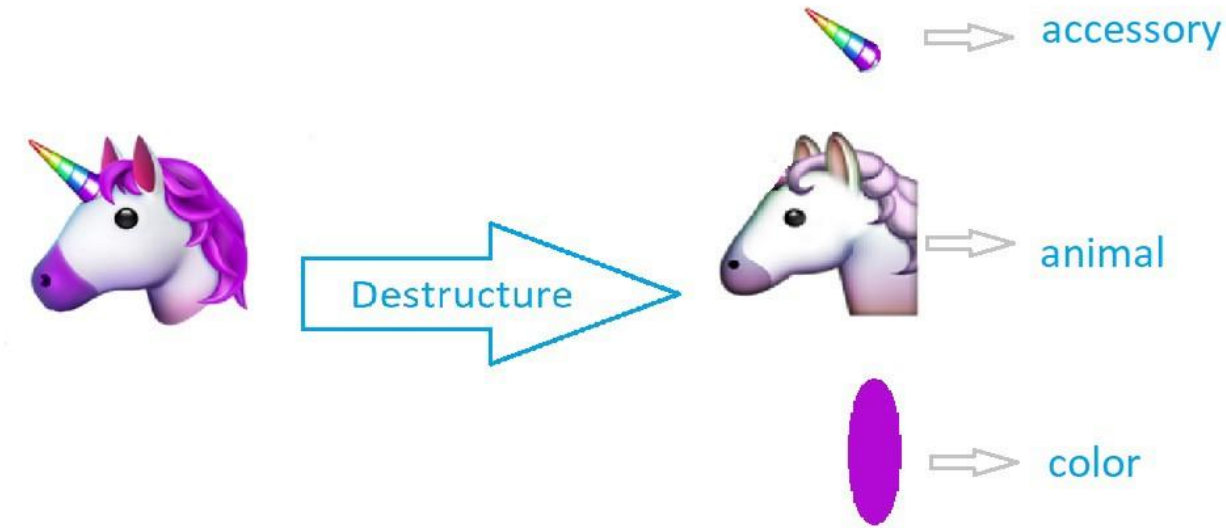
ARROW FUNCTION

An arrow function expression is a compact alternative to a traditional function expression, but is limited and can't be used in all situations.

```
// ES5
var add = function (num1, num2) {
    return num1 + num2;
}

// ES6
var add = (num1, num2) => num1 + num2
```

Destructuring



Rest Parameters & Spread Operator

The spread operator allows us to spread the value of an array (or any iterable) across zero or more arguments in a function or elements in an array (or any iterable).

The rest parameter allows us to pass an indefinite number of parameters to a function and access them in an array.

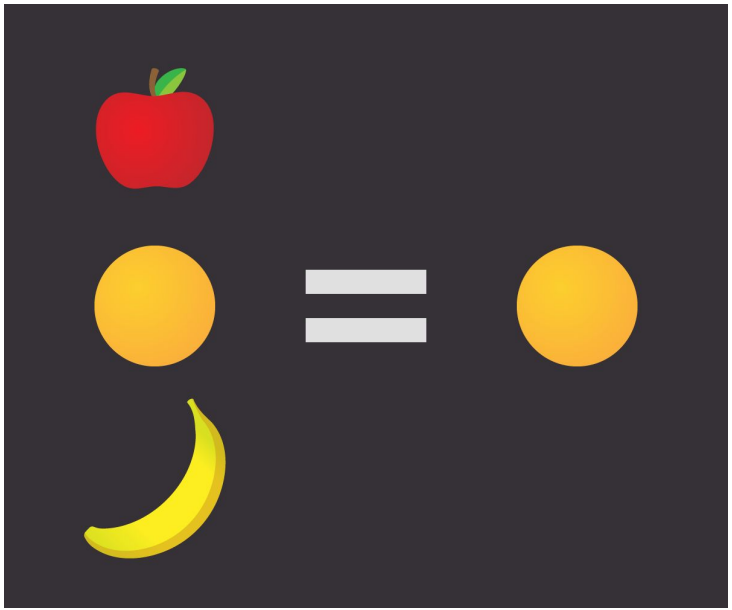
Default Parameters

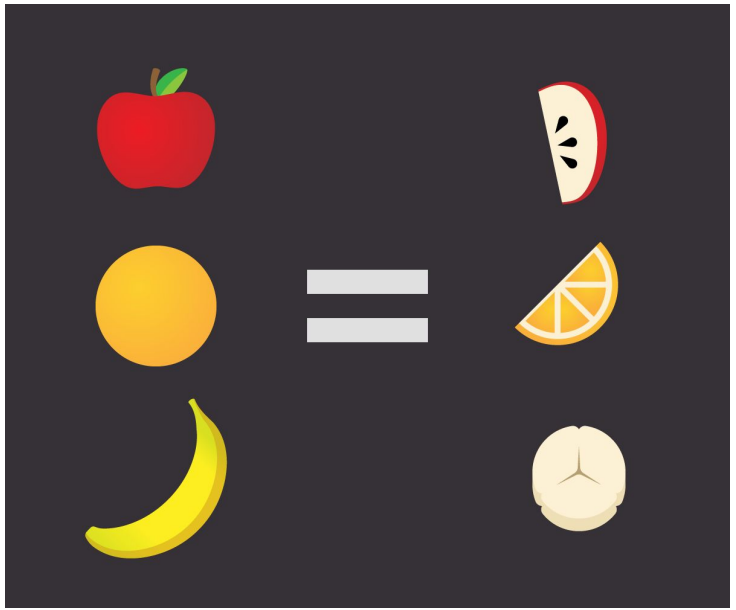
```
function say(message='Hi') {
    console.log(message);
}

say(); // 'Hi'
say('Hello') // 'Hello'
```

New Array Methods

`filter` vs `map`





For of

```
<script>
    let numeros = [11, 33, 55, 77, 99];

    for (let i of numeros) {
        console.log(i);      // log -> 11, 33, 55, 77, 99
    }
</script>
```

Thank you