the shortcut

Fullstack Web Development Tutorial Lesson 7

Today's lesson will cover

- Methods of primitives
- Strings
- Arrays

JavaScript fundamentals

Methods of primitives

- JavaScript allows us to work with primitives (strings, numbers, etc.) as if they were objects
- The “object wrappers” are different for each primitive type and are called: `String`, `Number`, `Boolean` and `Symbol`. Thus, they provide different sets of methods.
    - Refer to MDN for each of their methods whenever needed for anything you are working on

Strings

- Single and double quotes are essentially the same. Backticks, however, allow us to embed any expression into the string, by wrapping it in `${...}`
- str.length provides length as a numeric property
- toLowercase() or toUpperCaseu() is used to change case
- Searching for substring is possible in different ways:
  - Modern methods:
    - str.includes()
    - str.startsWith()
    - str.endsWith()

Special Characters	
Character	**Description**
\n	New line
\r	Carriage return: not used alone. Windows text files use a combination of two characters \r\n to represent a line break.
\', \"	Quotes
\\	Backslash
\t	Tab
\b, \f, \v	Backspace, Form Feed, Vertical Tab – kept for compatibility, not used nowadays.

Arrays

- Objects are not handy to work in case of ordered collection
    - No way to insert properties in between existing ones
- Array is the special data structure to store ordered collections
- Syntax
    - `let arr = new Array(); // Good to know but rarely used`
    - `let arr = [];`
- Array elements are numbered, starting with zero
- You can replace, add, count or do any modifications to arrays
- Arrays can store any elements, including objects or functions
- **Length property** automatically updates when array is modified. To be precise, it is actually not the count of values in the array, but the greatest numeric index plus one.

Array Methods: pop/push, shift/unshift

- Array supports both operations of **queue**, which in computer science means an ordered collections of elements which supports:
  - `push` appends an element to the end.
  - `shift` get an element from the beginning, advancing the queue, so that the 2nd element becomes the 1st.
- Another use case of arrays is with the data structure named stack. It supports two operations:
  - `push` adds an element to the end.
  - `pop` takes an element from the end
  - So new elements are added or taken always from the “end”
- Arrays in JavaScript can work both as a queue and as a stack. They allow you to add/remove elements both to/from the beginning or the end.
  - In computer science the data structure that allows this, is called deque.
- Methods that work with the end of array: `pop` and `push`
- Methods that work with the beginning of the array: `shift` and `unshift`
- Array is a special kind of object. The square brackets used to access a property `arr[0]` actually come from the object syntax. That’s essentially the same as `obj[key]`, where `arr` is the object, while numbers are used as keys.

Loops in Arrays

- for loop works to loop over indexes
- for..of is meant for looping through arrays
- Possible to use for..in but with potential problems
    - for..in is optimized for generic objects, not arrays
    - Generally, shouldn't use for..in for arrays

Multidimensional Array

- Multidimensional arrays aren’t supported in Javascript natively
- Arrays can have items that are also arrays. We can use it for multidimensional arrays, for example to store matrices

Summary

- Array is a special kind of object, suited to storing and managing ordered data items.
- The `length` property is the array length or, to be precise, its last numeric index plus one. It is auto-adjusted by array methods.
  - If we shorten `length` manually, the array is truncated.
- We can use an array as a deque with the following operations:
  - `push(...items)` adds `items` to the end.
  - `pop()` removes the element from the end and returns it.
  - `shift()` removes the element from the beginning and returns it.
  - `unshift(...items)` adds `items` to the beginning.
- To loop over the elements of the array:
  - `for (let i=0; i<arr.length; i++)` – works fastest, old-browser-compatible.
  - `for (let item of arr)` – the modern syntax for items only,
  - `for (let i in arr)` – never use.

the shortcut

Self Study Assignments

To Dos

- Continue freecodecamp Javascript. Ideally finish before we resume after summer.
- Continue with FCC HTML, CSS lessons. Ideally finish all the lessons by end of this month.
- If you need help pushing your HTML CSS project on GIthub and using Github pages let me know right away.