the shortcut

Fullstack Web Development Tutorial Lesson 9

Today's lesson will cover

- Destructuring
- JSON Methods

JavaScript fundamentals

Destructuring

- *Destructuring assignment* is a special syntax that allows us to “unpack” arrays or objects into a bunch of variables
- Array destructuring:
  - “destructurizes” by copying items into variables. But the array itself is not modified.
  - ...newArrayName if want to get all values from certain point onwards
  - Default values if unavailable then it’s undefined
- Object destructuring:
  - Works almost similar to array destructuring
  - ...rest pattern works too
  - Nested destructuring is possible too

Exercise: Destructuring

- Write the destructuring assignment that reads, for the object `user` below:
  - `name` property into the variable `name`.
  - `years` property into the variable `age`.
  - `isAdmin` property into the variable `isAdmin` (false, if no such property)

- ```
  let user = {
    name: "John",
    years: 30
  };
  ```
- `console.log` the variable outputs
  ```

Exercise: Destructuring

- Destructure the following object to get the following string output: “JS Fullstack: `HTML`, `CSS`, `JavaScript` `nodeJS` `React`”
- ```
  const webTechObj = {
       browser : {
       markup: 'HTML',
       styling: 'CSS',
       programming: {
            browser: 'JavaScript',
            general: 'Python',
            framework: {
                 javascript: ['React', 'Angular'],
                 python: ['django', 'flask']
            }
       }
       },
       server: ['Java', 'nodeJS']
  }
  ```
  ```

JSON

- JSON stands for Javascript Object Notation
- “key” : “value” pair is must
- Strings need to be within “double-quotes”
- Function properties (methods) are skipped with JSON methods
- Trailing commas like object or arrays are prohibited
- Syntax

```
{
        “key” : “string value”,
        “Number key within quote” : number value without quote,
        “key” : true // boolean value without quotes
        “key” : { object},
        “key” : [array elements or empty array],
        “Key” : { Nested JSON properties },
        “key” : null
}
```

JSON Methods: JSON.stringify

- `JSON.stringify` to convert objects into JSON
- JSON is data-only language-independent specification, so some JavaScript-specific object properties are skipped by `JSON.stringify`.
  - Function properties (methods).
  - Symbolic properties.
  - Properties that store `undefined`
- Syntax
  - `JSON.stringify(value[, replacer, space])`
- *`replacer`* parameter can be either a function or an array.
- `space` parameter is used solely for logging and nice-output purposes.
- Custom “toJSON”

JSON Methods: JSON.parse

- `JSON.parse` method parses a JSON string, constructing the JavaScript value or object described by the string
- Syntax
  - `JSON.parse(str, [reviver])`
- If a `reviver` is specified, the value computed by parsing is *transformed* before being returned.

Exercise: JSON Stringify

- Write `replacer` function to stringify everything, but remove properties (convert to `undefined` when stringifying that reference `meetup`:
-

```
let room = {
  number: 23
};

let meetup = {
  title: "Conference",
 occupiedBy: [{name: "John"}, {name: "Alice"}],
  place: room
};

// circular references which can cause error without replacer function
room.occupiedBy = meetup;
meetup.self = meetup;
// Fix error with
function replacer(key, value)

console.log( JSON.stringify(meetup, replacer)
```

```
/* resulting JSON should be:
{
  "title":"Conference",
  "occupiedBy":[{"name":"John"},{"name":"Alice"}],
  "place":{"number":23}
}
*/
```

Summary: Destructuring

- Destructuring assignment allows for instantly mapping an object or array onto many variables.
- The full object syntax:

  `let {prop : varName = default, ...rest} = object`
- This means that property `prop` should go into the variable `varName` and, if no such property exists, then the `default` value should be used.

  Object properties that have no mapping are copied to the `rest` object.
- The full array syntax:

  `let [item1 = default, item2, ...rest] = array`
- The first item goes to `item1`; the second goes into `item2`, all the rest makes the array `rest`.
- It's possible to extract data from nested arrays/objects, for that the left side must have the same structure as the right one.

Summary: JSON

- JSON is a data format that has its own independent standard and libraries for most programming languages.
- JSON supports plain objects, arrays, strings, numbers, booleans, and `null`.
- JavaScript provides methods JSON.stringify to serialize into JSON and JSON.parse to read from JSON.
- Both methods support transformer functions for smart reading/writing.
- If an object has `toJSON`, then it is called by `JSON.stringify`.

the shortcut

Self Study Assignments

To Dos

- For practical use case understanding of JSON, complete this tutorial: Active learning: Working through a JSON example
- 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.