Contents

Day-1	4
1.HTML ATTRIBUTES:	4
2. HTML INPUT TAGS:	5
CSS:	6
INLINE CSS:	6
INTERNAL CSS:	7
EXTERNAL CSS:	8
CSS SELECTORS:	9
CSS ID SELECTORS:	10
CLASS SELECTOR:	11
UNIVERSAL SELECTOR:	12
CSS VARIABLES	13
Local Variables:	13
Override/Global Variables:	13
DAY-2	14
Internal Js:	14
External Js:	15
Functions	16
SCOPE:	16
LOCAL SCOPE:	16
GLOBAL SCOPE:	17
ES6 STANDARDS:	17
let:	17
Const:	18
Arrow Function:	18
Promises:	19
JS CLASSES:	19
JS class syntax:	19
Js Class Methods:	20
JavaScript Event Handlers	20
addEventListner	21

remove:	21
DAY-3	22
What are the different types of event handlers provided by JavaScript?	22
Pass by Value:	24
Pass by Reference:	25
JS ASYNC/AWAIT:	29
Async/Await:	29
Object Keys:	30
Object Entries:	32
HTML_JS_CUSTOM ATTRIBUTES:	34
EXAMPLE-1:	34
EXAMPLE-2	35
EXAMPLE:3	36
DAY-4	37
LOCAL STORAGE:	37
Example1:	37
Objects:	39
Creating a JavaScript Object	39
Using an Object Literal	39
Using the JavaScript Keyword new	40
Example1:	40
Example2:	41
Example3:	41
Example4:	41
Arraylteration:	43
arrayForEach:	43
ArrayMap:	44
ArrayFilter:	44
ArrayReduce:	45
ArrayIndexOf:	45
ArrayLastIndexOf:	46
ArrayFind:	46
CreateElement	47

JSON:	48
Why Use JSON?	48
Storing Data	48
JSON PARSE:	49
Example1:	49
Example2:	49
Example3:	50
JSON Stringify:	50
Stringify object:	50
Example-1:	50
Stringify Array:	51
Example-2:	51
JS STRING PROPERTIES:	52
Sample example:	52
Example1:	52
JS STRING METHODS:	52
indexOf():	52
Example2:	52
last Index Of ():	53
Example3:	53
Replace():	54
Example4:	54
Slice():	54
Example5:	54
toString():	55
Example6:	55
toLowerCase():	55
Example7:	55
toUpperCase():	56
Example8:	56
JS Current Date and Time:	57

JSON:

- JSON stands for JavaScript Object Notation
- JSON is a lightweight data-interchange format
- JSON is plain text written in JavaScript object notation
- JSON is used to send data between computers
- JSON is language independent *

Why Use JSON?

The JSON format is syntactically similar to the code for creating JavaScript objects. Because of this, a JavaScript program can easily convert JSON data into JavaScript objects.

Since the format is text only, JSON data can easily be sent between computers, and used by any programming language.

JavaScript has a built in function for converting JSON strings into JavaScript objects:

JSON.parse()

JavaScript also has a built in function for converting an object into a JSON string:

JSON.stringify()

You can receive pure text from a server and use it as a JavaScript object.

You can send a JavaScript object to a server in pure text format.

You can work with data as JavaScript objects, with no complicated parsing and translations.

Storing Data

When storing data, the data has to be a certain format, and regardless of where you choose to store it, text is always one of the legal formats.

JSON makes it possible to store JavaScript objects as text.

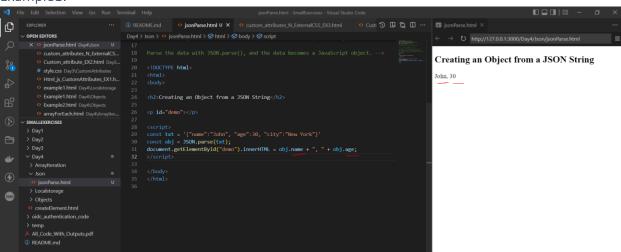
JSON PARSE:

A common use of JSON is to exchange data to/from a web server.

When receiving data from a web server, the data is always a string.

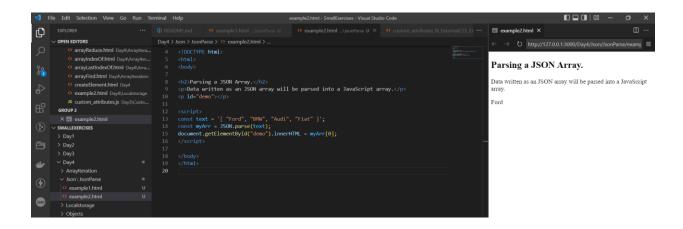
Parse the data with JSON.parse(), and the data becomes a JavaScript object.

Example1:



Example2:

Json Array:



Example3:

Parsing Dates:

```
| File Edit Selection View Go Run Terminal Help | example12 | December 1 | December 2 | December 3 | December 3 | December 3 | December 3 | December 4 | December 3 | December 4 | December 4 | December 5 | December
```

JSON Stringify:

Stringify object:

Example-1:

```
File Edit Selection View Go Run Terminal Help

Proposer Running Research View Go Run Terminal Help

Proposer Running Research View Go Run Terminal Help

Proposer Running Research Running Running Research Running Running Research Running Running Research Running Running Research Running Research
```

Stringify Array:

Example-2:

