**1) Typeof data types present in javascript:-**

1. console.log(typeof(1));--->**number**.
2. console.log(typeof(1.1));--->**number**.
3. console.log(typeof('1.1'));--->**string**.
4. console.log(typeof(true));--->**boolean**.
5. console.log(typeof(null));--->**object**.
6. console.log(typeof(undefined));--->**undefined**.
7. console.log(typeof([]));--->**object**.
8. console.log(typeof({}));--->**object**.

**2) Write a blog about objects and its internal representation in javascript:-**

Conceptually, Objects are the same in all programming languages i.e they represent real-world things that we want to represent inside our programs with characteristics/properties and methods.

For Eg. If your object is a student, it will have properties like name, age, address, id, etc and methods like updateAddress, updateNam, etc.

Let’s see an example:-

let firstObj = {

1: “deepak”,

“age”: 28

}

firstObj is an object with 2 properties 1 and age and value as deepak and 28 .

JavaScript objects are somewhat different in the way they are created. There is no requirement for a class as such.

## **2.1) Object Creation:-**

Object literals are a comma-separated list of key-value pairs wrapped in curly braces. Object literal property values can be of any data type, including array literals, functions, nested object literals or primitive data type.

Ex1:-

let student = {

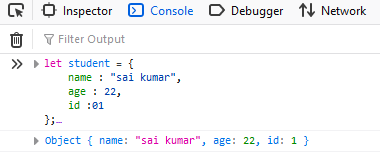
name : "sai kumar",

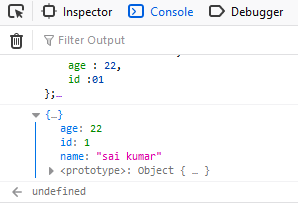
age : 22,

id :01

};

console.log(student);

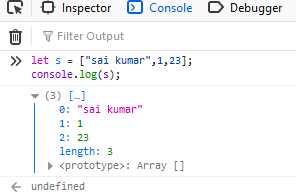




Ex2:-

let s = ["sai kumar",1,23];

console.log(s);



In the above picture , the output of given array shown in object representation as [key,value]

Pairs.

-> s[0] is the student name

-> s[1] is the student id number.

-> s[2] is the student age.

The length of the given array is 3.It also shown in the above picture.

**3) Host JSON server in Heroku:-**

[**https://charu-jsonserver.herokuapp.com/**](https://charu-jsonserver.herokuapp.com/)

sai kumar@SHIVA-SAI-KUMAR MINGW64 /e/heroku/json-server-heroku (master)

$ heroku login

heroku: Press any key to open up the browser to login or q to exit:

Opening browser to https://cli-auth.heroku.com/auth/cli/browser/0b690670-09ca-4008-80d0-5ede24a10b88?requestor=SFMyNTY.g3QAAAACZAAEZGF0YW0AAAAMMjcuNTkuMTgwLjY4ZAAGc2lnbmVkbgYAExYmYnQB.YI8cgs4JaRo5ClUJbnthZ5GdFI5Kb4GHdPv-zltenqg

Logging in... done

Logged in as charusivasaikumar@gmail.com

sai kumar@SHIVA-SAI-KUMAR MINGW64 /e/heroku/json-server-heroku (master)

$ heroku create charu-jsonserver

Creating ⬢ charu-jsonserver... done

https://charu-jsonserver.herokuapp.com/ | https://git.heroku.com/charu-jsonserver.git

sai kumar@SHIVA-SAI-KUMAR MINGW64 /e/heroku/json-server-heroku (master)

$ git remote -v

heroku https://git.heroku.com/charu-jsonserver.git (fetch)

heroku https://git.heroku.com/charu-jsonserver.git (push)

origin https://github.com/reach2arunprakash/json-server-heroku.git (fetch)

origin https://github.com/reach2arunprakash/json-server-heroku.git (push)

sai kumar@SHIVA-SAI-KUMAR MINGW64 /e/heroku/json-server-heroku (master)

$ git push heroku master

Enumerating objects: 31, done.

Counting objects: 100% (31/31), done.

Delta compression using up to 4 threads

Compressing objects: 100% (29/29), done.

Writing objects: 100% (31/31), 222.05 KiB | 672.00 KiB/s, done.

Total 31 (delta 13), reused 0 (delta 0), pack-reused 0

remote: Compressing source files... done.

remote: Building source:

remote:

remote: -----> Node.js app detected

remote:

remote: -----> Creating runtime environment

remote:

remote: NPM\_CONFIG\_LOGLEVEL=error

remote: NODE\_ENV=production

remote: NODE\_MODULES\_CACHE=true

remote: NODE\_VERBOSE=false

remote:

remote: -----> Installing binaries

remote: engines.node (package.json): unspecified

remote: engines.npm (package.json): unspecified (use default)

remote:

remote: Resolving node version 12.x...

remote: Downloading and installing node 12.18.3...

remote: Using default npm version: 6.14.6

remote:

remote: -----> Installing dependencies

remote: Installing node modules (package.json)

remote: added 224 packages from 121 contributors and audited 224 packages in 7.857s

remote:

remote: 1 package is looking for funding

remote: run `npm fund` for details

remote:

remote: found 2 low severity vulnerabilities

remote: run `npm audit fix` to fix them, or `npm audit` for details

remote:

remote: -----> Build

remote:

remote: -----> Caching build

remote: - node\_modules

remote:

remote: -----> Pruning devDependencies

remote: audited 224 packages in 1.922s

remote:

remote: 1 package is looking for funding

remote: run `npm fund` for details

remote:

remote: found 2 low severity vulnerabilities

remote: run `npm audit fix` to fix them, or `npm audit` for details

remote:

remote: -----> Build succeeded!

remote: -----> Discovering process types

remote: Procfile declares types -> (none)

remote: Default types for buildpack -> web

remote:

remote: -----> Compressing...

remote: Done: 24.7M

remote: -----> Launching...

remote: Released v3

remote: https://charu-jsonserver.herokuapp.com/ deployed to Heroku

remote:

remote: Verifying deploy... done.

To https://git.heroku.com/charu-jsonserver.git

\* [new branch] master -> master

sai kumar@SHIVA-SAI-KUMAR MINGW64 /e/heroku/json-server-heroku (master)

$ heroku open

**4) All tags in HTML with explanations:-**

**Head tag:** Head tag is used to contain all the head elements in the html file. It contains the title, style, meta, … etc tag.

**Syntax:**

<head> Statements... </head>

**Body tag:** It is used to define the body of a html document. It contains images, tables, lists, … etc.

**Syntax:**

<body> Statements... </body>

**Title tag:** It is used to define the title of a html document.

**Syntax:**

<title> Statements... </title>

**Content container tag:**

**Heading tag:** It is used to define the heading of a html document.

**Syntax:**

<h1> Statements... </h>

<h2> Statements... </h2>

<h3> Statements... </h3>

<h4> Statements... </h4>

<h5> Statements... </h5>

<h6> Statements... </h6>

**Paragraph tag:** It is used to define paragraph content in html documents.

**Syntax:**

<p> Statements... </p>

**Emphasis tag:** It is used to render as emphasized text.

**Syntax:**

<em> Statements... </em>

**Bold tag:** It is used to specify bold content in html documents.

**Syntax:**

<b> Statements... </b>

**Italic tag:** It is used to write the content in italic format.

**Syntax:**

<i> Statements... </i>

**Small (text) tag:** It is used to set the small font size of the content.

**Syntax:**

<small> Statements... </small>

**Underline tag:** It is used to set the content underline.

**Syntax:**

<u> Statements... </u>

**Deleted text tag:** It is used to represent as deleted text. It crosses the text content.

**Syntax:**

<strike> Statements... </strike>

**Anchor tag:** It is used to link one page to another page.

**Syntax:**

<a href="..."> Statements... </a>

**List tag:** It is used to list the content.

**Syntax:**

<li> Statements... </li>

**Ordered List tag:** It is used to list the content in a particular order.

**Syntax:**

<ol> Statements... </ol>

**Unordered List tag:** It is used to list the content without order.

**Syntax:**

<ul> Statements... </ul>

**Comment tag:** It is used to set the comment in the html document. It is not visible on the browser.

**Syntax:**

<!-- Statements... -->

**Scrolling Text tag:** It is used to scroll the text or image content.

**Syntax:**

<marquee> Statements... </marquee>

**Center tag:** It is used to set the content into the center.

**Syntax:**

<center> Statements... </center>

**Font tag:** It is used to specify the font size, font color and font-family in html document.

**Syntax:**

<font> Statements ... <font>

**Empty (Non-Container) Tags:**

**Line break tag:** It is used to break the line.

**Syntax:**

<br>

**Image tag:** It is used to add image element in html document.

**Syntax:**

<img>

**Link tag:** It is used to link the content from external sources.

**Syntax:**

<link>

**Horizontal rule tag:** It is used to display the horizontal line in an html document.

**Syntax:**

<hr/>

**Meta tag:** It is used to specify the page description. For example: last modifier, authors, … etc.

**Syntax:**

<meta> Statements ... <meta>

**Tables Tags:** Table tag is used to create a table in html document.

**Table tag:**

**Syntax:**

<table> Statements... </table>

**Tr tag:** It is used to define row of html table.

**Syntax:**

<tr> Statements... </tr>

**Th tag:** It defines the header cell in a table. By default it set the content with bold and center property.

**Syntax:**

<th> Statements ... <th>

**Td tag:** It defines the standard cell in html document.

**Syntax:**

<td> Statements ... <td>

**Form tag:** It is used to create html form for user.

**Syntax:**

<form> Statements ... <form>

**Submit input tag:** It is used to take the input from the user.

**Syntax:**

<input>

**Dropdown option tag:** It is used to select an option from drop-down list.

**Syntax:**

<option> Statements ... <option>

**Radio button tag:** It is used to select only one option from the given options.

**Syntax:**

<input>