Celebrating Glorious Platinum Jubilee Year 2022

Walchand College of Engineering, Sangli

Walchand Linux Users' Group

WLUG
COMMUNITY | KNOWLEDGE | SHARE

PRESENTS HANDS ON

JavaScript

- Asynchronous Javascript
- Callbacks
- Promises
- Async/Await
- JS BOM

LIVE DEMO

WWW.WCEWLUG.ORG

CONNECT WITH US

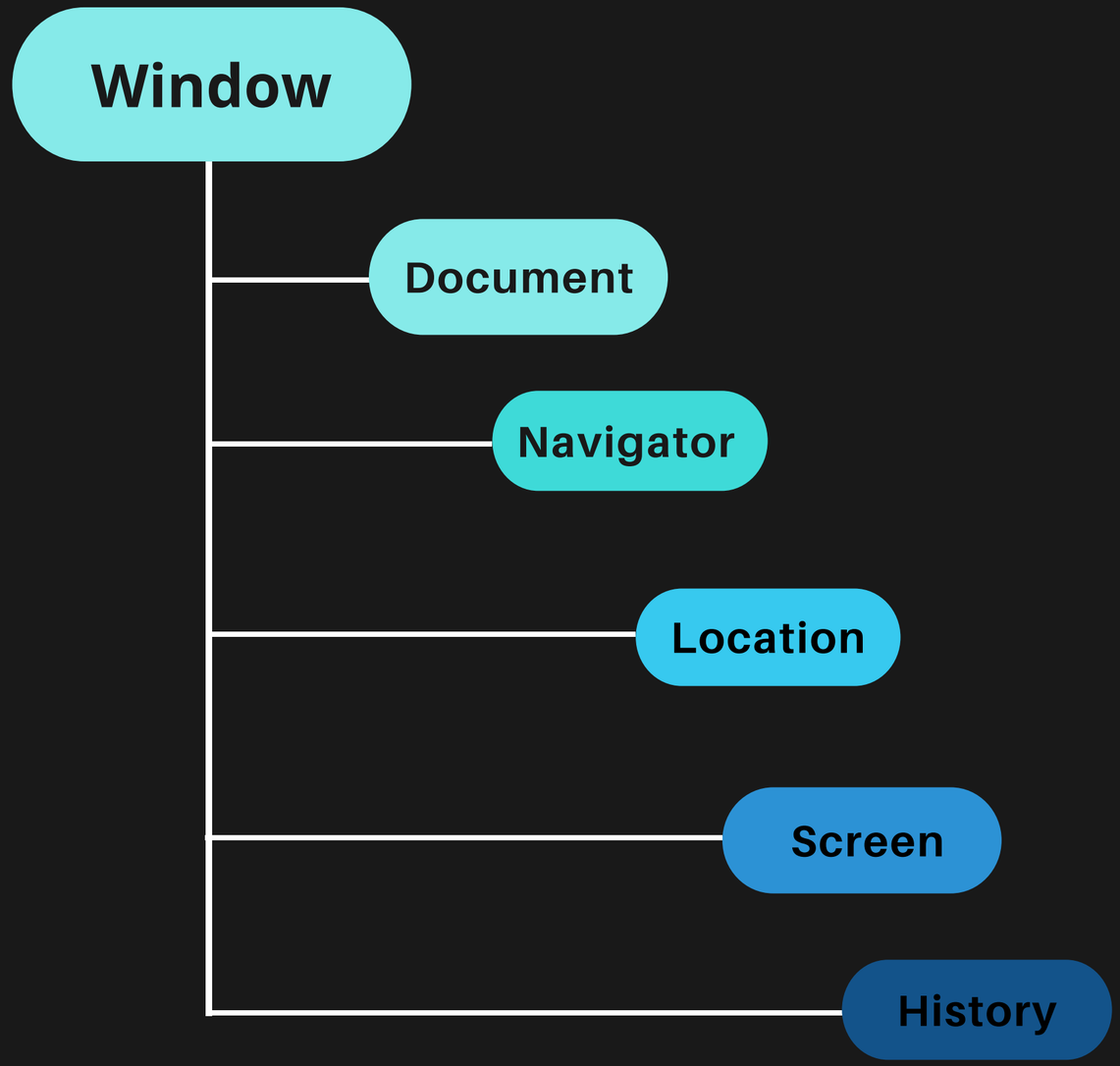
25th Sept
Sat 6:00 PM

Browser Object Model (BOM)

WLUG
COMMUNITY | KNOWLEDGE | SHARE

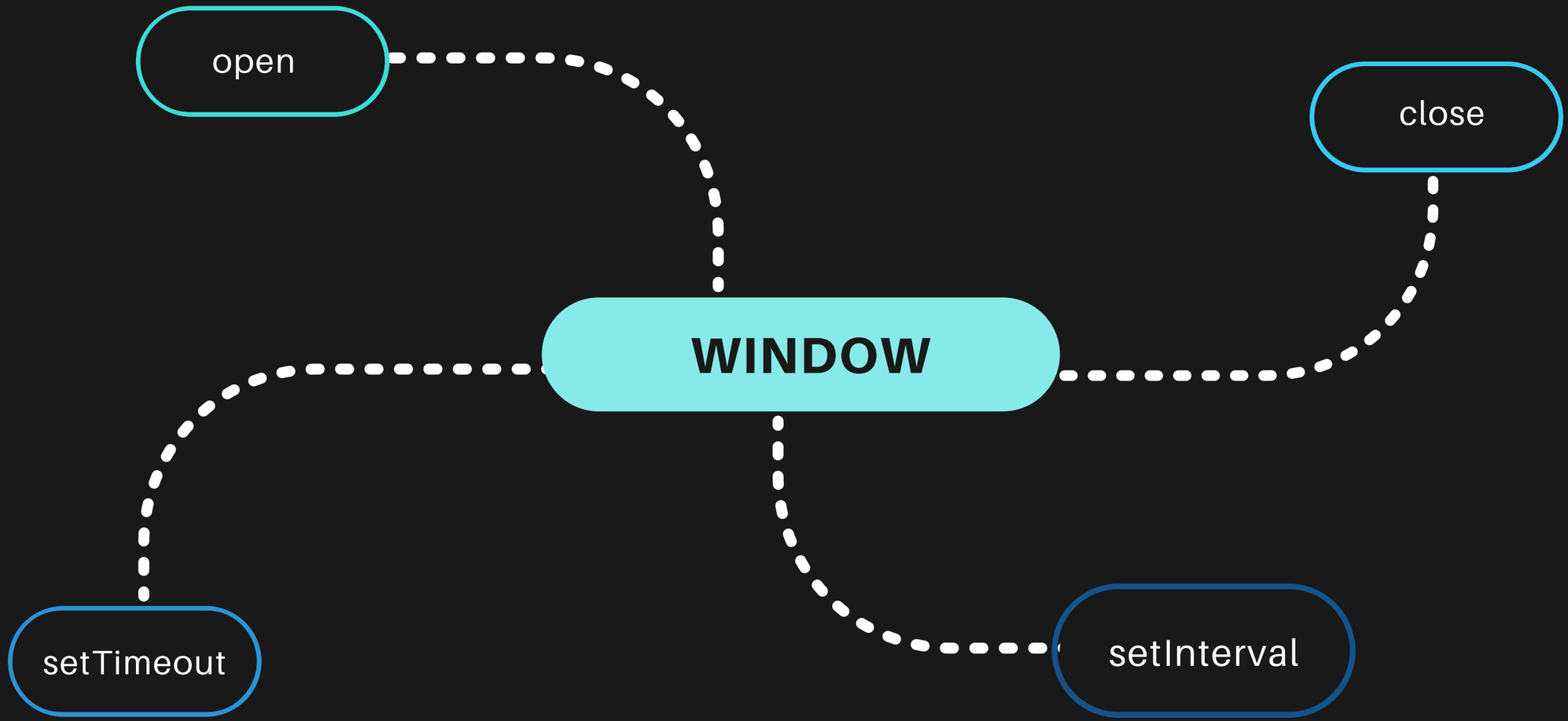
- The Browser Object Model (BOM) is used to interact with the browser.
- The BOM provides access to the various characteristics of a browser (Browser window, Screen characteristics etc).
- The default object of browser is window. Means we can call all the functions of window by specifying window or directly.

BOM Hierarchy

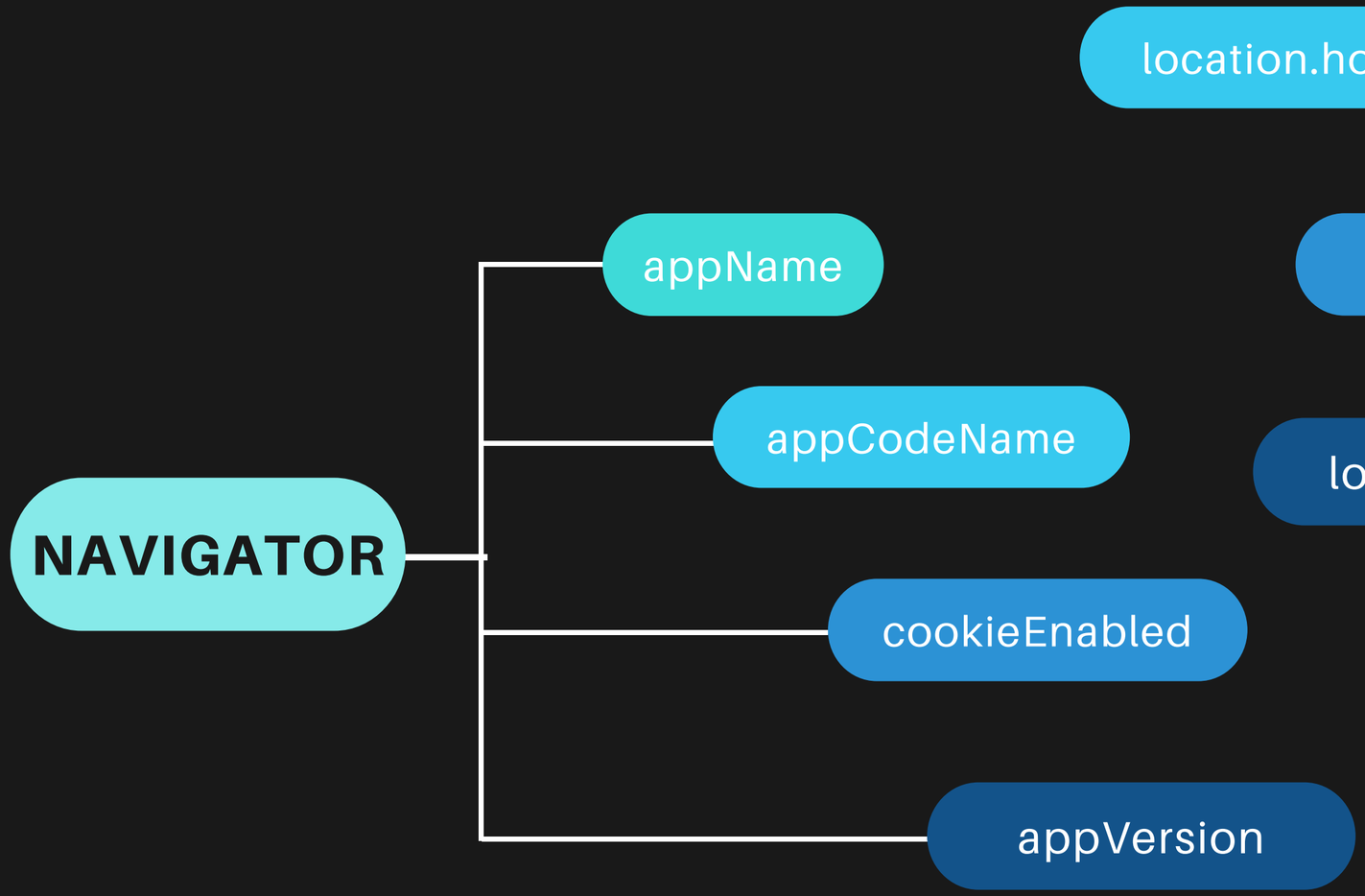


BOM - Window

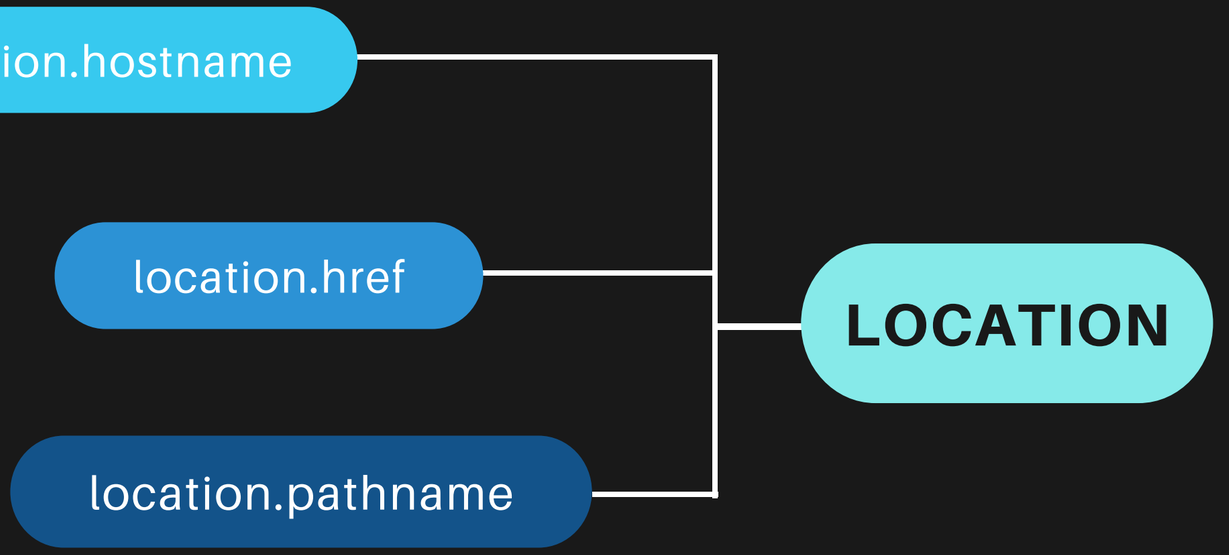
WLUG
COMMUNITY | KNOWLEDGE | SHARE



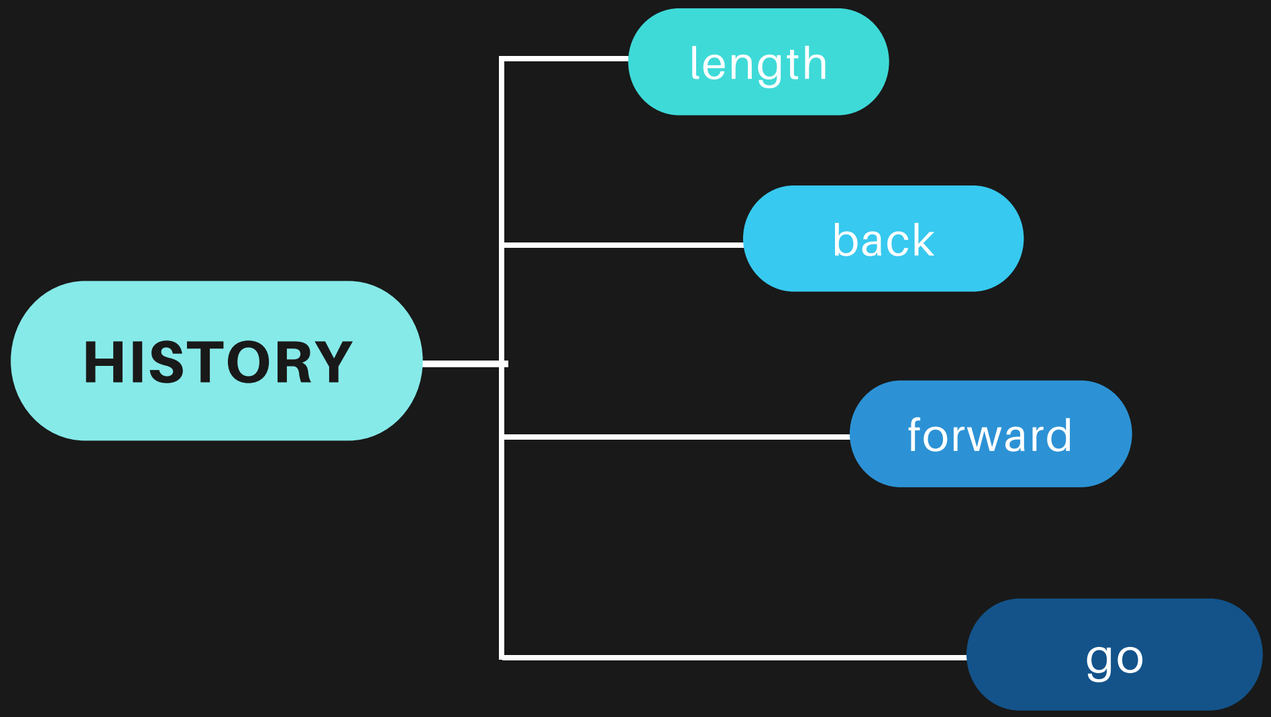
BOM - Navigator

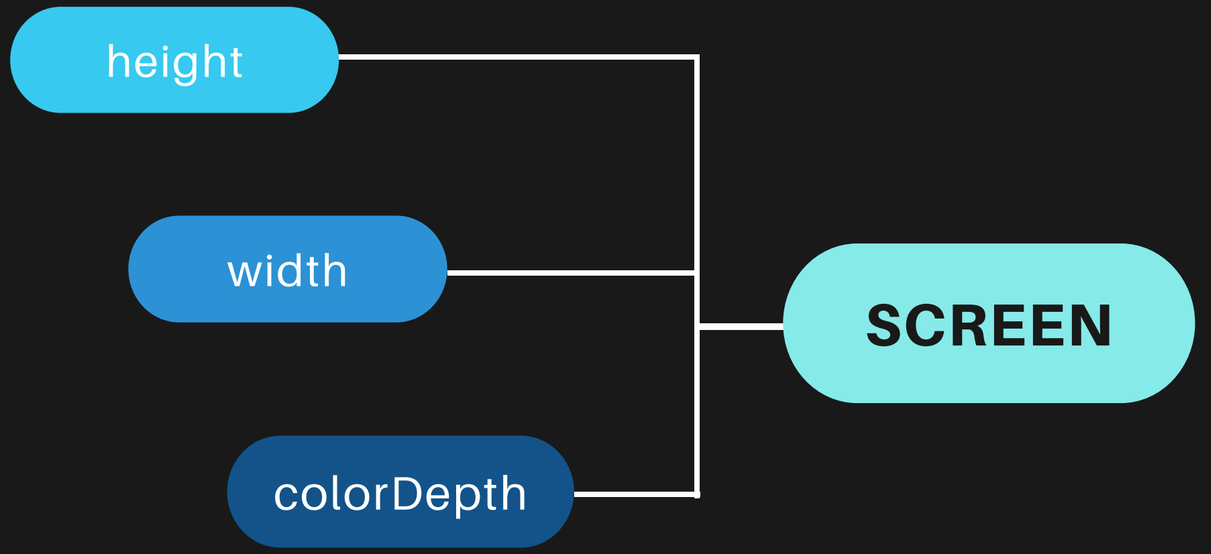


BOM - Location



BOM - History





BOM - Screen

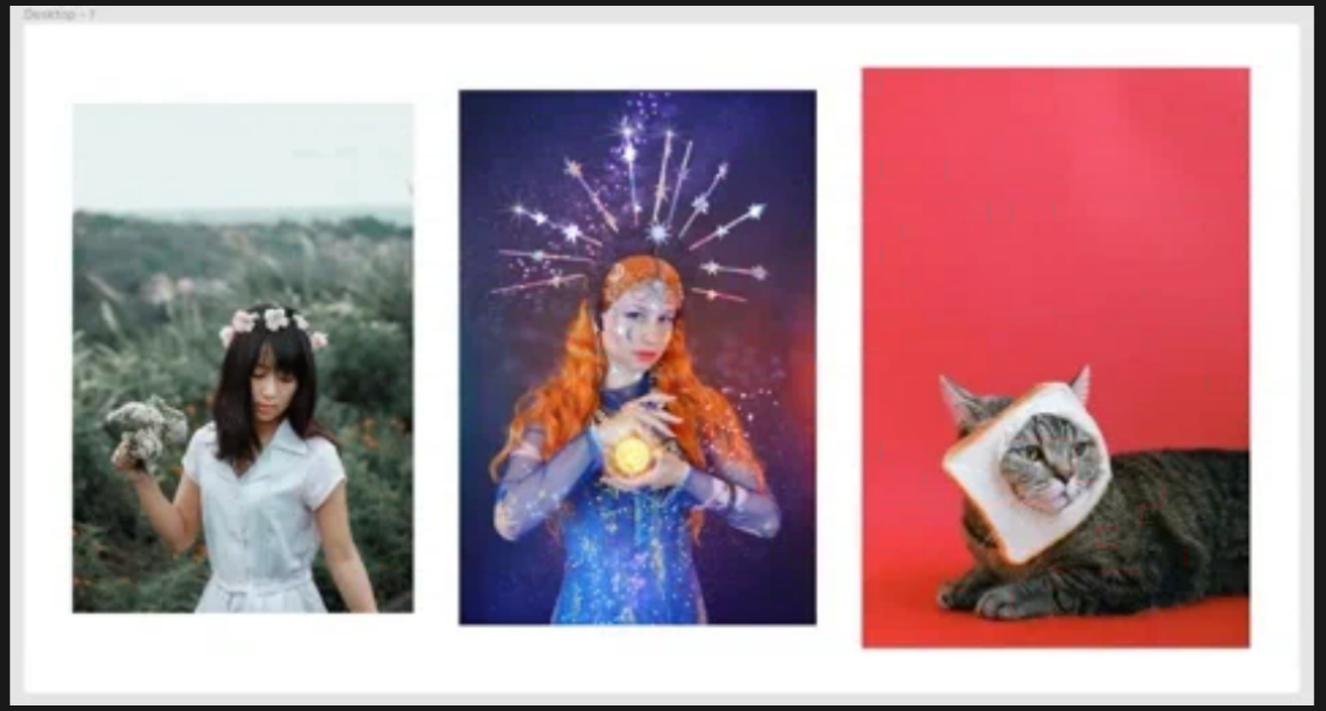
WLUG
COMMUNITY | KNOWLEDGE | SHARE

- Synchronous vs Asynchronous JavaScript
- How Callbacks Work in Js
- How Promises Work in Js
- How Async / Await Works in Js
- Event loop

Synchronous js

WLUG
COMMUNITY | KNOWLEDGE | SHARE

In a synchronous system, tasks are completed one after another.



WLUG
COMMUNITY | KNOWLEDGE | SHARE

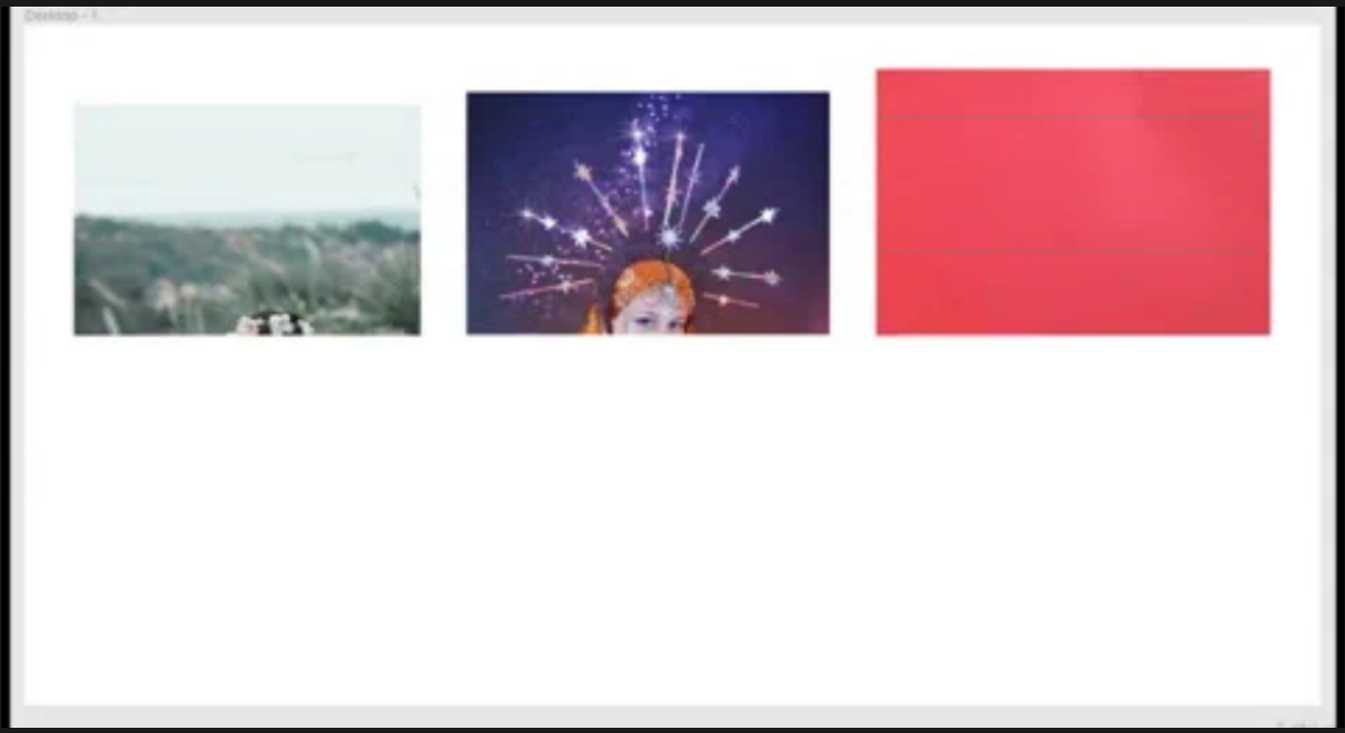
Drawback of Synchronous js

```
const processImage = (img) => {
    /**
     * doing some operation on image
     */
    console.log("Image processed");
}
const networkRequest = (url) => {
    /**
     * requesting network resources
     */
    return somedata;
}
const greeting = () => {
    console.log("Hello everyone");
}
processImage(WLUG.jpg);
networkRequest('http://www.wcewlug.org/');
greeting();
```

Asynchronous js

WLUG
COMMUNITY | KNOWLEDGE | SHARE

In Asynchronous system, tasks are completed independently.



Callbacks

WLUG
COMMUNITY | KNOWLEDGE | SHARE

A callback is a function passed into another function as an argument to be executed later

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Synchronous Callback

VS

Asynchronous Callback

Callback hell

WLUG
COMMUNITY | KNOWLEDGE | SHARE

```
function hell(win) {
 // for listener purpose
 return function() {
   loadLink(win, REMOTE_SRC+'/assets/css/style.css', function() {
     loadLink(win, REMOTE_SRC+'/lib/async.js', function() {
       loadLink(win, REMOTE_SRC+'/lib/easyXDM.js', function() {
         loadLink(win, REMOTE_SRC+'/lib/json2.js', function() {
           loadLink(win, REMOTE_SRC+'/lib/underscode.min.js', function() {
             loadLink(win, REMOTE_SRC+'/lib/backbone.min.js', function() {
               loadLink(win, REMOTE_SRC+'/dev/base_dev.js', function() {
                 loadLink(win, REMOTE_SRC+'/assets/js/deps.js', function() {
                   loadLink(win, REMOTE_SRC+'/src/' + win.loader_path + '/loader.js', function() {
                     async.eachSeries(SCRIPTS, function(src, callback) {
                       loadScript(win, BASE_URL+src, callback);
                     });
                   });
                 });
               });
             });
           });
         });
       });
     });
   });
 };
}
```

JAVASCRIPT PROMISES

WLUG
COMMUNITY | KNOWLEDGE | SHARE

- Promises in Real Life
- Promise definition
- Steps to create Promise
- Steps to use Promise
- Promise Applications

WHAT IS PROMISES

Promises
1. Solve array topics on leetcode
2. Read blogs
3. Play cricket
4. Learn shell scripting
5. Revise everything

States

- Pending
- Fulfilled
- Rejected

A promise is an object which keeps track whether a certain event has happened already or not

STEPS TO CREATE PROMISE

WLUG
COMMUNITY | KNOWLEDGE | SHARE

- Firstly, we use constructor to create a Promise object.

```
datatype mypromise= new Promise(function(resolve, reject)\
{  //
   //
};
```

- It takes two parameters, one for success(resolve) and one for fail(reject)

- Finally, there will be a condition. If the condition is met, the Promise will be resolved, otherwise it will be rejected

STEPS TO USE PROMISE

WLUG
COMMUNITY | KNOWLEDGE | SHARE

There are 2 cases:

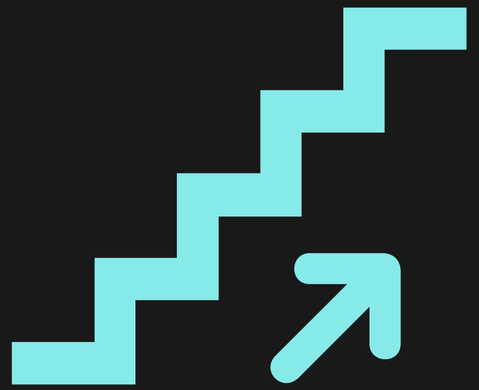
One for resolved promises and one for rejected.

1. Resolved Promises

```
mypromise.then();
```

2. Rejected Promises

```
mypromise.catch();
```



JS PROMISE EXAMPLES

WLUG
COMMUNITY | KNOWLEDGE | SHARE

1.Waiting for a Timeout:

Display a text on console screen after some delay by using settimeout() and Promise

2.Waiting for a File:

We will check whether a particular html file is present in our local machine or not

What is Async/Await

2 new keywords

async: special function
await: pauses execution of async function

EVENT LOOPS