CALLBACKS

Program

Date ....

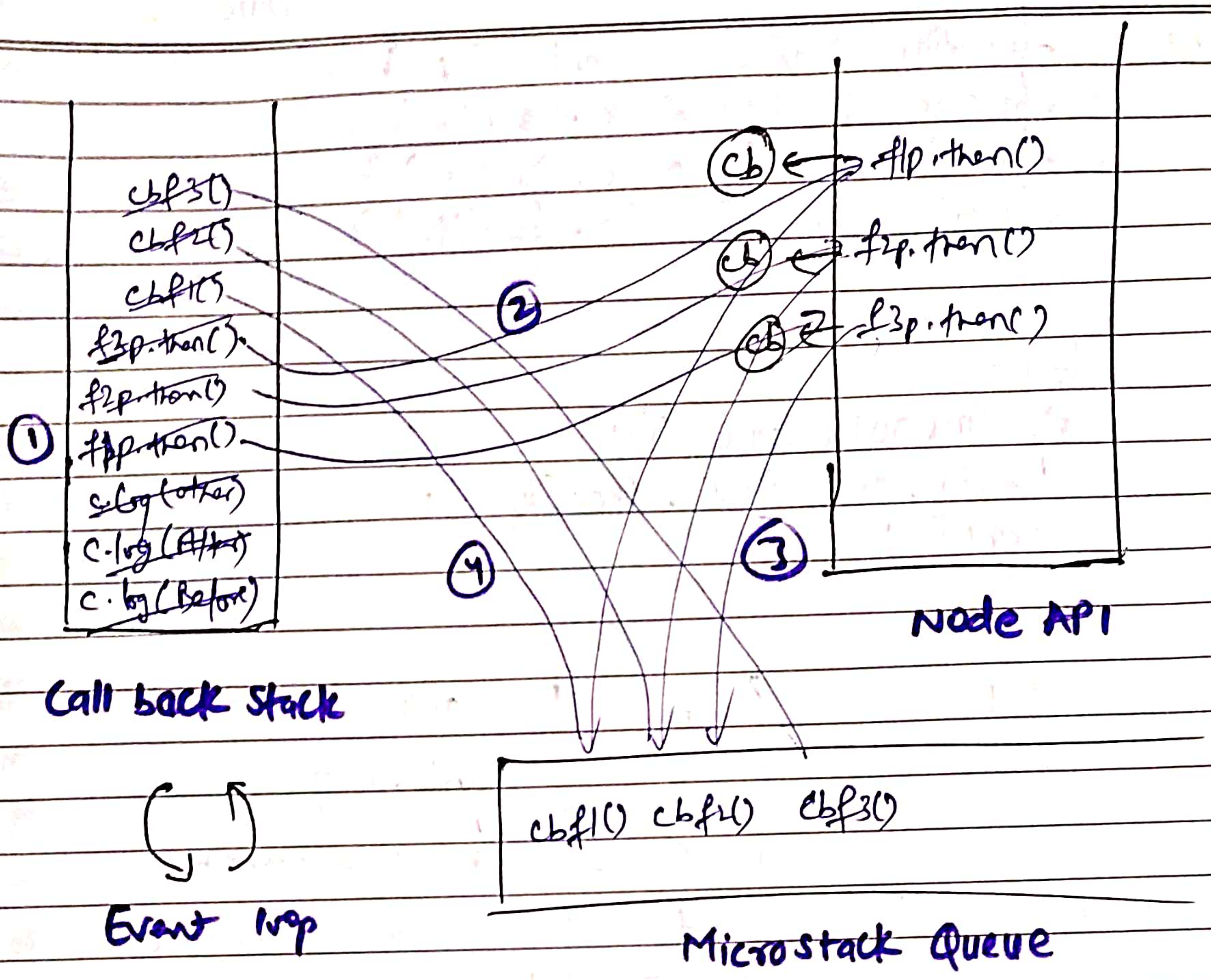
```
col.log ("Before");

let f1p= fs. promises. readFile ("f1.txt");
let f2p= fs. promises. readFile ("f2.txt");
let f3p= fs. promises. readFile ("f3.txt");

f1p. then(cb);      callbacks
f2p. then(cb);
f3p. then (cb);
                    Callback fn.
function cb( data)
        {  col. log ("contents->" + data);
        }

col. log (" after");
col. log (" others");
```

O/P

```
Before
after
others
f1 data
f2 data
f3 data
```

order can vary ie f2 f3 f1   f1 f3 f2   etc.

Date ....



(from Node API, callback's can come
in any order,
eg ~~f1~~ f2 f3
f3 f2 f1
f1 f3 f2
f2 f3 f1 etc.

(JS VISUALIZER 9000)
↑
<u>website</u>

(Task queue is Callback Queue).

Spiral

Date ....

eg

```
function logA() { c.log ('A') }
function log B() { c.log ('B') }
function log C() { c.log ('C') }
function log D() { c.log ('D') }
```

```
logA();
set Timeout( log B, 0);
Promise. resolve(). then( log C);
log D();
```

O/P →

A	only c.log
D	only. c.log.
C	bcz C ~~is~~ Promise
B	bcz it's a callback

⇒ (Promise has high priority than callback).

Use of Async Behaviour

Suppose you wants to download (5-6) videos, then It don't make sense that ~~the~~ in which orders the video is downloading.

Spiral