Promises

Agenda :

1. What are promises
2. Async Programming with Promises
3. Chaining of promises

CALL~~BACK~~ HELL

$f_1(\ )$

$f_2(\ )$

$f_3(\ )$

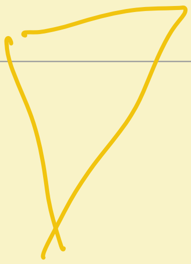
Promises

{ }

Object representing eventual completion or failure of an asynchronous function

$$\begin{tikzcd}
 & \text{Object} \arrow[dl] \arrow[d] \arrow[dr] & \\
\checkmark\ \text{pending} & \text{resolved} & \text{rejected}
\end{tikzcd}$$

$$\underbrace{\checkmark\ \text{(fulfilled)} \qquad \checkmark\ \text{(failure)}}_{\text{settled}}$$



Pyramid of Doom!!

CALLBACK HELL

```
getData (function (a) {
    getMore (a, function(b) {
        getEvenMore (b, function(c) {
            console.log (c)
        }
```

Cleaner / Easier to read

```
getData ( )
  .then (getMore)
  .then (getEvenMore)
  .then (console.log)
```

(Promises)

Async programming with Promises

1. Pending.
2. Fulfilled.
3. Rejected.

Promise is created using Promise Constructor.

new Promise

for classes class

Promises

(1) Creation

(2) Consumption

executor

success]

FAIL

callbacks

```
let myPromise = new Promise (function (res,
                                        rej ){
     if (true) {
          resolve ('success')
     }
     else { reject ('Fail')
     }
}
)
```

[PROMISE]

Consume a

promise

myPromise.then(. - - -)

.catch(. - — . -)

.finally(- . - -)

then → on success of a promise

catch → on fail/reject of a promise

finally → Always. (Settled of a promise)