

Promesas

DEV.F
DESARROLLAMOS(PERSONAS);

dev

Promesas

- Es usado para interacciones asíncronas
- Se compone de dos aspectos:
 - **Resolve:** Se ejecuta cuando el objetivo de la promesa se efectuó de manera correcta
 - **Reject:** Se ejecuta cuando el objetivo de la promesa ocasionó un error o no se llegó a cumplir.
- Se utiliza la palabra reservada ***“Promise”***
- En las promesas se tienen siempre tres estados:
 - **Pendiente:** Estado inicial de la promesa
 - **Resuelta:** Todo se ejecutó correctamente
 - **Rechazada:** Hubo un problema

En otras palabras..



Ejemplo

```
const promise = new Promise((resolve, reject) => {
  const number = Math.floor(Math.random() * 10);

  setTimeout(
    () => number > 5
      ? resolve(number)
      : reject(new Error('Menor a 5')),
    1000
  );
});

promise
  .then(number => console.log(number))
  .catch(error => console.error(error));
```

Callback Hell

DEV.F
DESARROLLAMOS(PERSONAS);

dev

```

1 function hell(win) {
2   // for listener purpose
3   return function() {
4     loadLink(win, REMOTE_SRC+'/assets/css/style.css', function() {
5       loadLink(win, REMOTE_SRC+'/lib/async.js', function() {
6         loadLink(win, REMOTE_SRC+'/lib/easyXDM.js', function() {
7           loadLink(win, REMOTE_SRC+'/lib/json2.js', function() {
8             loadLink(win, REMOTE_SRC+'/lib/underscore.min.js', function() {
9               loadLink(win, REMOTE_SRC+'/lib/backbone.min.js', function() {
10                loadLink(win, REMOTE_SRC+'/dev/base_dev.js', function() {
11                  loadLink(win, REMOTE_SRC+'/assets/js/deps.js', function() {
12                    loadLink(win, REMOTE_SRC+'/src/' + win.loader_path + '/loader.js', function() {
13                      async.eachSeries(SERIALS, function(src, callback) {
14                        loadScript(win, BASE_URL+src, callback);
15                      });
16                    });
17                  });
18                });
19              });
20            });
21          });
22        });
23      });
24    });
25  });
26 }

```



```

pan.pourWater(function() {
  range.bringToBoil(function() {
    range.lowerHeat(function() {
      pan.addRice(function() {
        setTimeout(function() {
          range.turnOff();
          serve();
        }, 15 * 60 * 1000);
      });
    });
  });
});

```



pyramid of doom