

```

1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>promiseAll</title>
9  </head>
10
11 <body>
12
13     <script>
14         /*
15         ,           ***** promise all *****
16         ,
17         ,           Promise.all은 주어진 모든 Promise를 실행한 후 진행되는
18         ,           하나의 Promise를 반환한다.
19         ,
20         ,           가끔 우리는 하나의 API가 아닌 3개, 4개의 API에서 값을 불러와야 할 때가
21         ,           있을것이다.
22         ,           예를들자면 영화리스트를 볼 수 있는 앱을 만들었는데,
23         ,           영화들, 인기가많은 영화들, 항상인기가많은영화들 을 불러와야하는데
24         ,           내가 원하는것은 그 3개의 값만을 기다리는것인데
25         ,           그걸 p1 then, p2 then, p3 then으로 수많은 then을 넣어주는 것보다
26         ,           Promise.all 으로 한개의 Promise로 반환하는게 좋을것이다.
27         ,           모든 Promise가 전부 resolve되고 나면 마지막 Promise를 리턴해주는것
28         ,
29         ,           const p1 = new Promise(resolve => {
30         ,               setTimeout(resolve, 5000, "First")
31         ,           })
32         ,
33         ,           const p2 = new Promise(resolve => {
34         ,               setTimeout(resolve, 1000, "Second")
35         ,           })
36         ,
37         ,           const p3 = new Promise(resolve => {
38         ,               setTimeout(resolve, 3000, "Third")
39         ,           })
40         ,
41         ,           const motherPromise = Promise.all([p1, p2, p3]);
42         ,           motherPromise.then(items => console.log(items));
43         ,
44         ,           모든 Promise들이 진행이 되고 값을 얻었다.
45         ,           즉 Promise.all이 다른 promise들이 전부 진행될 때까지 기다렸다가 진행되었다는뜻.
46         ,           나오는 값들이 Array라는 뜻도 되는것이다. function 에서 resolve된 value들의 array
47         ,
48         ,           그리고 p1,p2,p3가 언제끝나는지에 상관없이 순서대로 받은것이다.
49         ,           이론에 따르면 p1은 마지막 값이 되어야하는데, 그런 작업들을 하지 않는것
50         ,           순서에 맞춰서 값을 제공한다.
51         ,
52         ,           * 이제 만약 마지막 promise가 reject되면 어떻게 될까
53         ,
54         ,           Uncaught (in promise) Third 이런 error가 나온다.
55         ,           Promise가 reject해서 제대로 작동하지 않은것
56         ,
57         ,           Promise.all은 현재 3개의 Promise들이 필요한데 제공한 Promise가
58         ,           하나라도 reject되면 motherPromise도 reject가 되는것이다.
59         ,           그래서 catch를 써주면
60         ,
61         ,           motherPromise

```

```
62 | ,                .then(items => console.log(items))
63 | ,                .catch(err => console.log(err));
64 | ,
65 | ,                보이는것처럼 reject가 되는 Promise에는 catch를 쓰지않았는데
66 | ,                에러값이 제대로 출력이된다.
67 | ,                p3 promise에 넣어준 값이 나온다. p3 Promise가 거절을 당했다는 뜻.
68 | ,                만약 p2에다가 같은식으로 해보면 그 다음것은 진행이 안된다. 이미 앞에 에러가 나서 소용이 없어
    | 친것이니.
69 | ,
70 | ,                */
71 |
72 |     const p1 = new Promise(resolve => {
73 |         setTimeout(resolve, 5000, "First")
74 |     })
75 |
76 |     const p2 = new Promise(resolve => {
77 |         setTimeout(resolve, 1000, "Second")
78 |     })
79 |
80 |     const p3 = new Promise((resolve, reject) => {
81 |         setTimeout(reject, 3000, "Third")
82 |     })
83 |
84 |     const motherPromise = Promise.all([p1, p2, p3]);
85 |     motherPromise
86 |     .then(items => console.log(items))
87 |     .catch(err => console.log(err));
88 |
89 |     </script>
90 | </body>
91 |
92 | </html>
```