

```
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>promiseRace</title>
9  </head>
10
11 <body>
12
13     <script>
14         /*
15         ,           ***** promise race *****
16         ,
17         ,           Promise Race는 Promise All 이랑 사용법은 같은데 다르다.
18         ,           다른점은 이 세개중 하나라도 resolve되거나 reject되면 된다는것.
19         ,
20         ,           Promise Race가 resolve되어서 then으로 넘어가거나
21         ,           reject되어서 catch로 넘어가려면
22         ,           p1,p2,p3중 하나만 resolve되거나 reject되면 가능.
23         ,
24         ,           여기서 가장 빠른것이 결과를 정한다.
25         ,
26         ,           const p1 = new Promise((resolve, reject) => {
27         ,               setTimeout(resolve, 5000, "First")
28         ,           })
29         ,
30         ,           const p2 = new Promise((resolve, reject) => {
31         ,               setTimeout(reject, 1000, "Second")
32         ,           })
33         ,
34         ,           const p3 = new Promise((resolve, reject) => {
35         ,               setTimeout(resolve, 3000, "Third")
36         ,           })
37         ,
38         ,           const motherPromise = Promise.race([p1,p2,p3]);
39         ,           motherPromise
40         ,               .then(value => console.log(value))
41         ,               .catch(err => console.log(err))
42         ,
43         ,           를 해보면 Second가 출력된다. 왜냐하면 p2가 가장 빠르기 때문.
44         ,           reject하는데 1초밖에 걸리지않는다.
45         ,
46         ,           어느것이 먼저 되는지 상관 없을 때 Race를 사용하면 된다.
47         ,           현재 p2가 reject된 이후에 다른것들이 실행되어야하는데
48         ,           Promise Race는 다른 Promise들에 대해 아무것도 하지않는다.
49         ,
50         ,           ** Race는 이 중 어떤 것이든지 성공하거나 reject가 발생하지 않으면
51         ,           resolve 된다.
52         ,
53         ,           Promise.race([p1,p2,p3])
54         ,               .then(value => console.log(value))
55         ,               .catch(err => console.log(err))
56         ,
57         ,           이런식으로 써주어도 정상작동될것이다.
58         ,           */
59
60         const p1 = new Promise((resolve, reject) => {
61             setTimeout(resolve, 5000, "First")
```

```
62     })
63
64     const p2 = new Promise((resolve, reject) => {
65         setTimeout(reject, 5000, "Second")
66     })
67
68     const p3 = new Promise((resolve, reject) => {
69         setTimeout(resolve, 3000, "Third")
70     })
71
72     Promise.race([p1,p2,p3])
73         .then(value => console.log(value))
74         .catch(err => console.log(err))
75
76
77
78     </script>
79 </body>
80
81 </html>
```