

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>HTML Fragments</title>
8  </head>
9  <body>
10     <div class="wrapper"></div>
11
12     <script>
13         /*
14         ,           ***** HTML Fragments *****
15         ,
16         ,           string안에 표현식을 넣을 수 있다는 점 말고
17         ,           template literal을 멋지게 사용하는 경우는
18         ,           javascript안에서 html을 쓸 수 있다는 점이다.
19         ,
20         ,           예를들어 인터넷에서 어떤 이름을 가져오는 function이 있다고해보자
21         ,           이름이나 어떤 정보를 가져온다고 가정하고
22         ,           welcome 메시지를 추가한다고 해보자
23         ,
24         ,           const wrapper = document.querySelector(".wrapper");
25         ,
26         ,           const addWelcome = () => {
27         ,               const potato = document.createElement("div");
28         ,               const h1 = document.createElement("h1");
29         ,               h1.innerText = "Hello";
30         ,               potato.append(h1);
31         ,               wrapper.append(potato);
32         ,           }
33         ,
34         ,           setTimeout(addWelcome, 3000);
35         ,
36         ,           로딩하고 3초 후에 addWelcome이 실행되면서
37         ,           Hello를 가지고 있는 엘리먼트를 반환하게된다.
38         ,
39         ,           만약 복잡한 구조의 html 을 만들거나 className을 같이 만들게 된다면 어떻게될까.
40         ,           h1.className = "sexyTitle"
41         ,           이런식으로 한 줄씩 계속 커지면서 길어질것이다.
42         ,
43         ,           그렇게 하는 대신 1개의 div만 만들고
44         ,           template literal 안에 원하는것을 써주면 된다.
45         ,           물론 변수도 안에 넣어줄 수 있다.
46         ,
47         ,           전보다 더 보기 편해졌다.
48         ,           우리는 HTML Fragments를 만든것이다.
49         ,           html을 위한 fragments로만 동작하는것이 아니라
50         ,           어떤것이든 될 수 있다.
51         ,
52         ,           여기서 중요한것은 template literal이 우리가 만드는 space도 고려해준다는것이다
53         ,           enter한것도 알아준다.
54         ,           */
55
56     const wrapper = document.querySelector(".wrapper");
57
58     const addWelcome = () => {
59         const potato = document.createElement("div");
60         const h1 = document.createElement("h1");
61         h1.innerText = "Hello";

```

```
62     potato.append(h1);
63     wrapper.append(potato);
64 }
65
66 // setTimeout(addWelcome, 3000);
67
68 const hello = "Hello"
69 const addTemplate = () => {
70     const div = `
71 ,         <div class="hello">
72 ,             <h1 class="title">${hello}</h1>
73 ,             </div>
74 ,         `;
75     wrapper.innerHTML = div;
76 }
77 addTemplate();
78
79 console.log(`
80 ,
81 ,
82 ,         enter , space
83 ,
84 ,     `) // enter,space를 고려하기 때문에 이런것도 가능하다.
85
86 </script>
87 </body>
88 </html>
89
```