-내용추가 : 버튼 클릭시 tb id를 가진 tbody의 내용 추가(tr, td). 넣을 내용 문자열 처리

|  |
| --- |
| var idx = 1;  function tbAdd() {  var html = "";  idx++;  html += **"<tr id=\"tr"** + idx + **"\">"**;  html += "<td>" + idx + "</td> ";  html += "<td>Test" + idx + "</td> ";  html += "</tr> ";    var tb = document.getElementById("tb");    tb.**innerHTML** += html;  } |

**\(역슬래시)** : 뒤에 오는 특수문자를 일반문자로 인식  
**innerHTML** : 엔티티 추가(내용 늘어남 - 기존 내용 뒤에 추가)

-내용삭제 : 버튼 클릭 시 tr 삭제

|  |
| --- |
| function tbDel() {  if(idx > 0) {  var obj = document.getElementById("tr" + idx);  var tb = document.getElementById("tb");  tb.**removeChild(obj)**;    // obj.parentNode.removeChild(obj); 도 가능하나 권장사항은 아님    idx--;  }  } |

**removeChile(객체)** : 자식 객체 중에 해당 객체와 동일한 것을 제거(obj.remove()도 동일한 기능을 하나 ie에서 작동 안함. obj.parentNode.removeChild(obj); 도 가능하나 권장사항은 아님)

-이동 : 앵커 태그 대신 location.href를 많이 이용

|  |
| --- |
| function go() {  location.href = "Test10.html";  } |

-css, js 파일 분리 : 1)공통적으로 쓸 경우. 2)특정 파일에서만 쓸 경우 내용관리가 필요하다면 분리

-css 분리 : link 태그 이용.

|  |
| --- |
| <link rel="stylesheet" type="text/css" href="./css/Test10.css" /> |

-script 분리 : script의 src속성 이용. 스크립트에서 src가 선언되면 해당 Entity를 해당 파일의 내용으로 채우겠다는 의미(덮어씀). 별도의 스크립트를 이용하고 싶다면 스크립트를 추가하면 됨

|  |
| --- |
| <script type="text/javascript" src="./js/Test10.js"></script> |

-이전 페이지 : history.back()을 주로 이용함. 몇 단계 이전으로 이동하려면 history.go(숫자) 이용

|  |
| --- |
| function goBack() {  // history.back()  // history.back();  // history.go(숫자) : 숫자만큼 이동했던 페이지 이동  history.go(-1);  } |

-confirm : 경고창을 통하여 값을 출력하며, 확인, 취소 버튼을 제공한다. 버튼을 통하여 결과를 boolean으로 받는다. 옛날 방식

|  |
| --- |
| function a() {  if(confirm("Hello?")) {  alert("Yes!");  } else {  alert("No~~~~!");  }  } |