

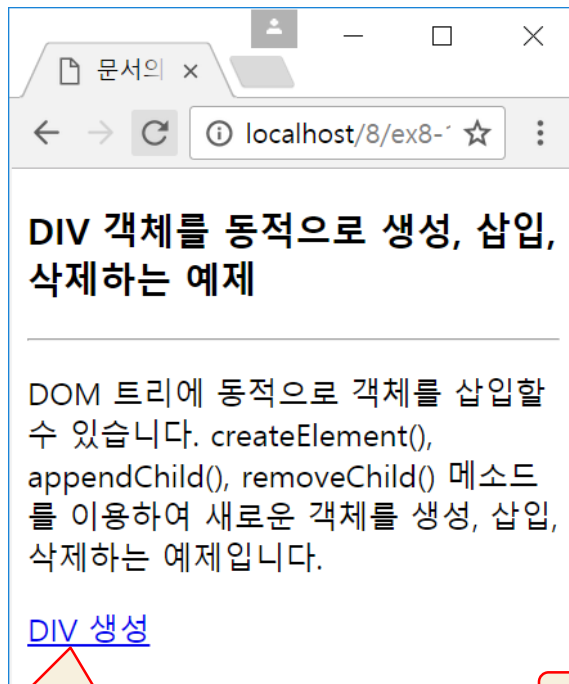


JS JavaScript

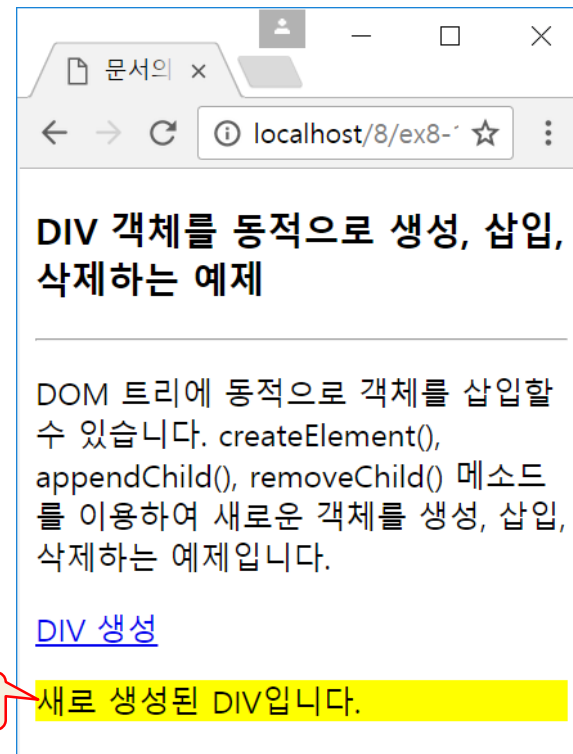
— JavaScript를 활용한 웹 페이지 제작

보너스 - HTML 태그의 동적 추가 및 삭제

- createElement, appendChild, createTextNode
- setAttribute



클릭하면 아래와 같이 <div> 태그가 삽입



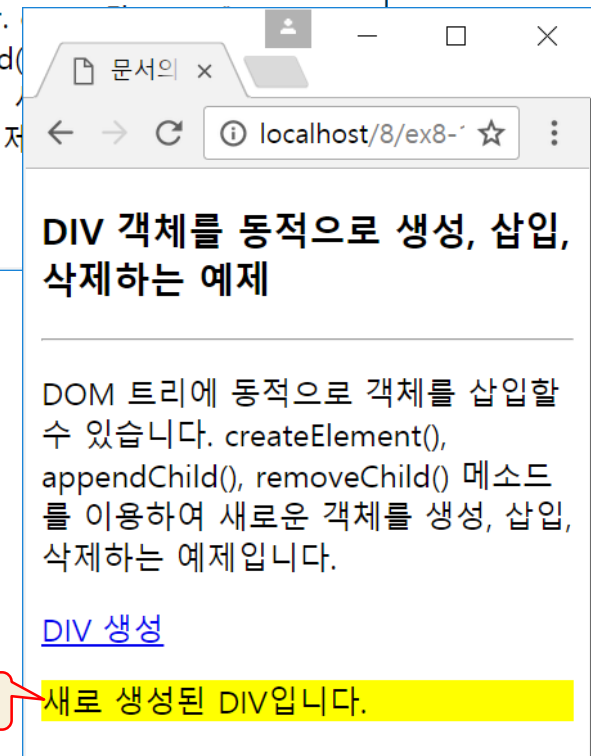
클릭하면 삭제

보너스 - HTML 태그의 동적 추가 및 삭제

```
<!DOCTYPE html>
<html>
<head><title>문서의 동적 구성</title>
<script>
function createDIV() {
    var obj = document.getElementById("parent");
    var newDIV = document.createElement("div");
    newDIV.innerHTML = "새로 생성된 DIV입니다.";
    newDIV.setAttribute("id", "myDiv");
    newDIV.style.backgroundColor = "yellow";
    newDIV.onclick = function() {
        var p = this.parentElement; // 부모 HTML 태그 요소
        p.removeChild(this); // 자신을 부모로부터 제거
    };
    obj.appendChild(newDIV);
}
</script>
</head>
<body id="parent">
<h3>DIV 객체를 동적으로 생성, 삽입, 삭제하는 예제</h3>
<hr>
<p>DOM 트리에 동적으로 객체를 삽입할 수 있습니다.
createElement(), appendChild(),
removeChild() 메소드를 이용하여 새로운 객체를 생성,
삽입, 삭제하는 예제입니다.</p>
<a href="javascript:createDIV()">DIV 생성</a><p>
<p>
</body>
</html>
```



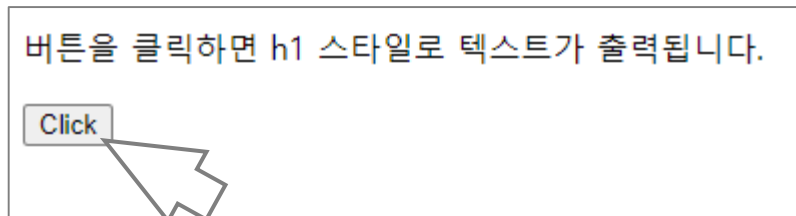
클릭하면 아래와 같이
<div> 태그가 삽입



클릭하면 삭제

HTML DOM 구조 연습1

- 버튼을 클릭하면 h1 스타일 텍스트 출력하기
 - ▣ createElement, createTextNode, appendchild 사용



HTML DOM 구조 연습2

- <select> 태그를 이용한 구구단 출력
 - ▣ 목록에 있는 단수를 선택하면 그림과 같이 출력되게 만들 것
 - ▣ 삭제 버튼을 누르면 출력되었던 구구단 내용이 삭제되게 만들 것

구구단 출력

단 :

3단

3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27

구구단 출력

단 :

3단

3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27

4단

4 * 1 = 4
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16
4 * 5 = 20
4 * 6 = 24
4 * 7 = 28
4 * 8 = 32
4 * 9 = 36

구구단 출력

단 :

Element의 attribute 값 획득/설정/제거

□ Element 속성 값 조작

- ▣ `getAttribute()` : 선택한 요소의 속성값을 가져옴

```
element.getAttribute(attrName);
```

- ▣ `setAttribute()` : 선택한 요소의 속성값을 변경

```
element.setAttribute(attrName);
```

- ▣ `removeAttribute()` : 선택한 요소의 속성값을 제거

```
element.removeAttribute(attrName);
```

setAttribute() – 요소의 속성 값을 정하는 메소드

□ setAttribute()

- 선택한 요소(element)의 속성(attribute) 값을 정합니다.
- 문법

```
element.setAttribute( 'attributename', 'attributevalue' )
```

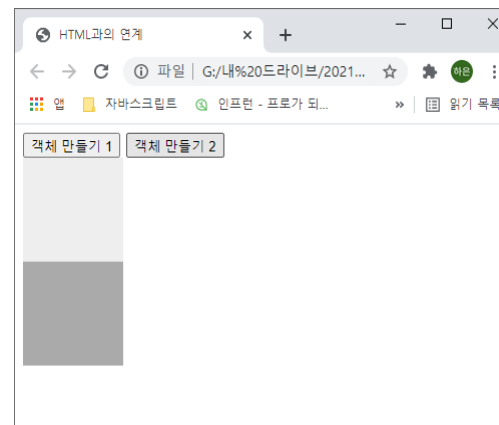
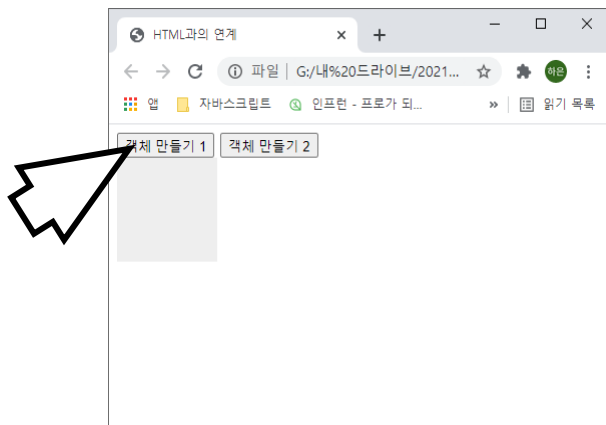
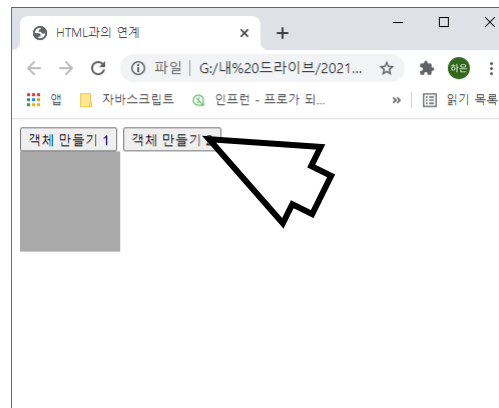
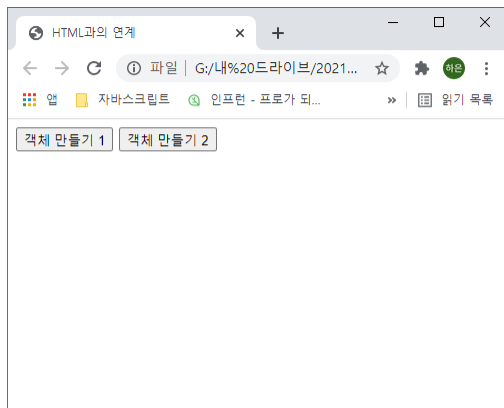
- attributename: 속성이름
- attributevalue: 속성값

- Id값이 test인 요소의 href 속성값을 변경하는 예제

```
<html lang="ko">
  <head>
    <title>JavaScript - setAttribute()</title>
  </head>
  <body>
    <p><a id="test" target="_blank">Web main page</a></p>
    <script>
      document.getElementById('test').setAttribute('href','https://www.naver.com' );
    </script>
  </body>
</html>
```

HTML DOM 구조 연습3

- 버튼을 누르면 생성되는 사각형 만들기
 - ▣ createAttribute, setAttribute, appendChild



Input 태그

- Form 태그 안에서 추가되는 입력 요소들 중 가장 중요한 태그
- 사용자로부터 정보를 받아들이는 용도로 사용됨
- Input 태그 형식
 - ▣ 1. type: 입력 태그의 유형
 - ▣ 2. value: 입력 태그의 초기값을 말하며 사용자가 변경가능함
 - ▣ 3. name: 서버로 전달되는 이름을 말함(사용자 임의 지정)

type 속성값	내용
text	텍스트를 입력하는 창을 생성 합니다.
password	비밀번호를 입력하는 창을 생성 합니다.
radio	라디오 버튼을 생성 합니다.
checkbox	체크박스를 생성 합니다.
file	파일 이름을 입력하는 창을 생성 합니다.
image	이미지를 전송 버튼으로 만듭니다.
hidden	사용자에게 보이지는 않지만 서버로 전송됩니다.
submit	서버로 제출/전송하는 버튼을 만듭니다.
button	일반 버튼을 생성 합니다.

Input 태그 value 속성

- <input> 요소의 초깃값(value)을 명시
 - ▣ “button”, “reset”, “submit” : 버튼 내의 텍스트를 정의
 - ▣ “hidden”, “password”, “text” : 입력 필드의 초깃값을 정의
 - ▣ “checkbox”, “image”, “radio” : 해당 입력 필드를 선택 시 서버로 제출되는 값을 정의

```
<!DOCTYPE html>
<html lang="ko">
<head>
    <meta charset="UTF-8">
    <title>HTML input tag - value attribute</title>
</head>
<body>

<form action="/examples/media/action_target.php" method="get">
    이름 : <input type="text" name="st_name" value="홍길동"><br>
    남 <input type="radio" name="gender" value="male">
    여 <input type="radio" name="gender" value="female"><br>
    <input type="submit" value="서버로 제출하기">
</form>

</body>
</html>
```

최종실습 - 전화번호부 만들기

- 저장 버튼: 이름, 전화번호를 입력하고 저장하면 하단의 주소록 목록에 나타난다.
- 리셋 버튼: 주소록 목록에 저장 되어있는 항목들이 삭제되며 배경을 변경했으면 배경도 원래의 색상으로 바꿔준다.
- 배경 변경: 배경 색상을 노란색으로 바꿔준다.
 - <form>태그를 이용하여 이름, 전화번호 입력 창 만들기
 - <button> or <input> 태그를 사용하여 버튼 만들기
 - onclick()메소드 사용
 - createTextNode(), appendChild(), querySelector(), innerHTML 사용

주소록 예제

이름:

전화번호:

주소록 목록

주소록 예제

이름:

전화번호:

주소록 목록

- 홍길동 , 010-1234-5678
- 김철수 , 010-1111-1111

주소록 예제

이름:

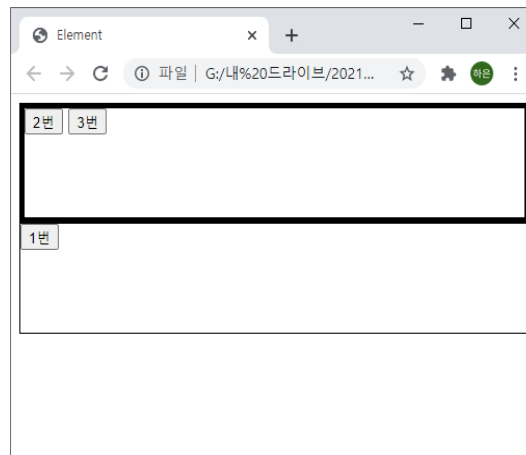
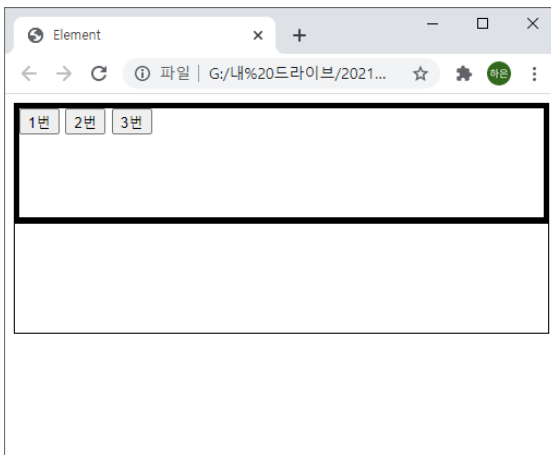
전화번호:

주소록 목록

- 홍길동 , 010-1234-5678
- 김철수 , 010-1111-1111

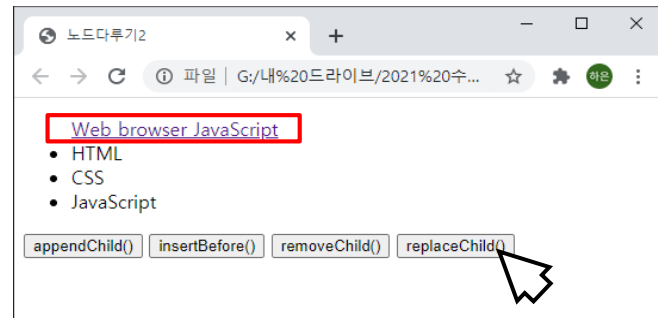
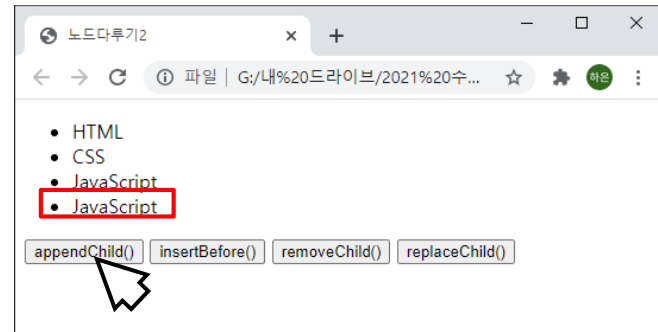
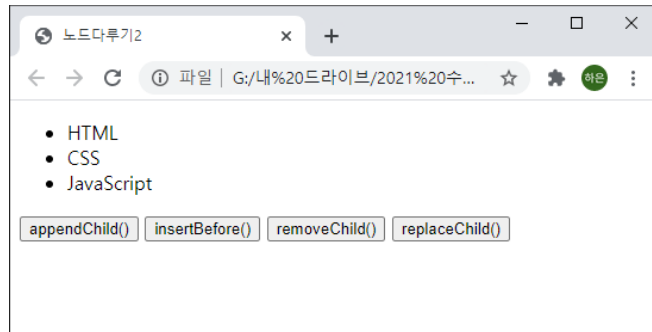
Node 다루기1

- 각 버튼을 누르면 아래줄로 이동하도록 만들기
 - ▣ getElementById 사용
 - ▣ click 이벤트 적용(HTML 태그 리스너 or addEventListener)
 - ▣ appendChild 사용



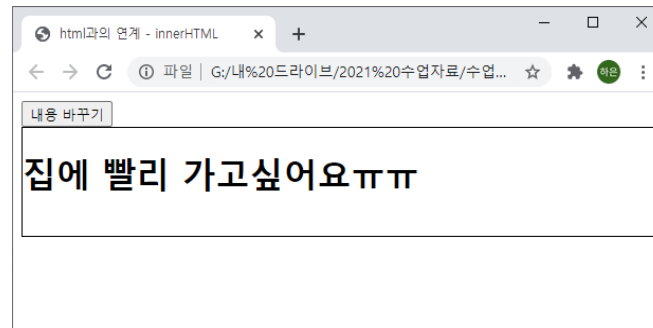
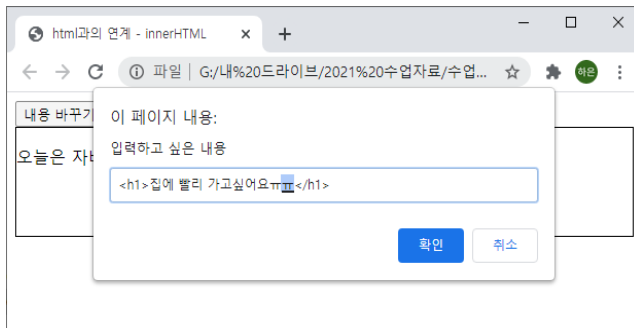
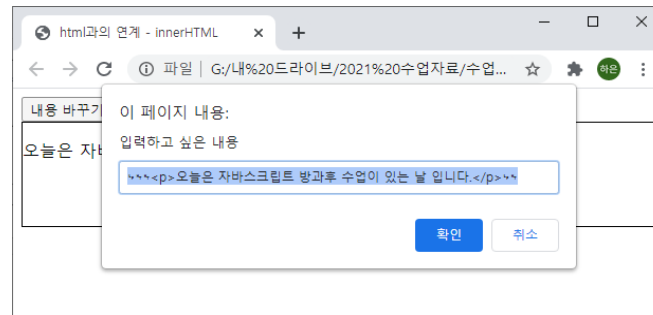
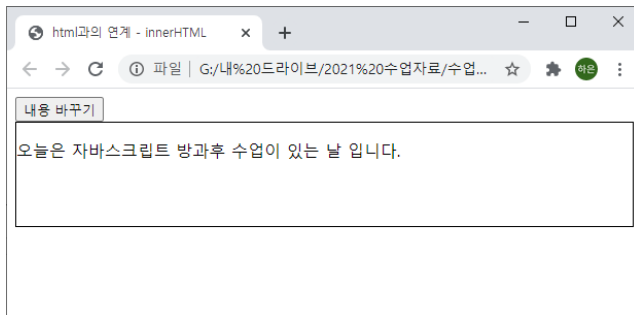
Node 다루기2

- 각 버튼을 아래와 같이 실행되도록 만들기
 - click 이벤트 적용(HTML 태그 리스너 or addEventListener)
 - createElement, createTextNode, insertBefore, removeBefore, appendChild 사용



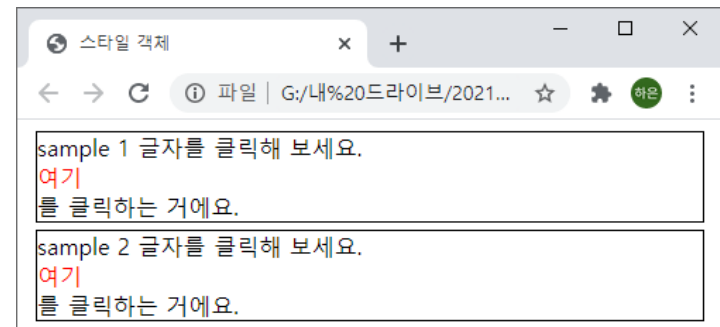
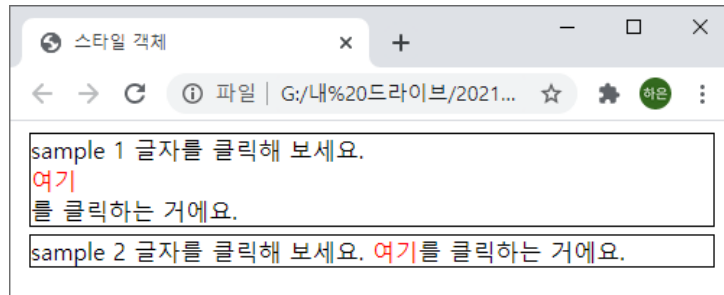
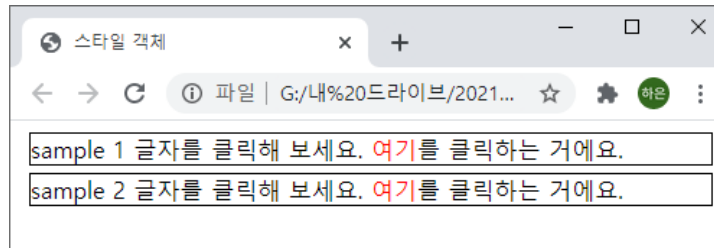
이벤트 다루기1

- 내용바꾸기 버튼 클릭 후 사용자가 입력한 문장으로 바꾸기
 - getElementById 사용
 - 이벤트 등록(html태그에 등록 or addEventListener사용)
 - innerHTML 사용



이벤트 다루기2

- “여기”라는 단어를 클릭하면 줄바꿈 처리, 다시 클릭 시 원래대로 한 줄에 표시되도록 만들기
 - ▣ 이벤트 등록(html태그에 등록 or addEventListener사용)
 - ▣ style.display=“block”, style.display=“inline” 활용



배열 객체 메서드(Array Method)

□ Array 객체

- 데이터 요소 여러 개를 묶어서 처리하고자 할때 사용
- 배열의 각 요소가 동일한 데이터 타입이 아니어도 되며, 배열의 크기는 언제라도 변경이 가능

분류	멤버	개요
가공	concat(arr)	지정 배열을 현재의 배열에 연결
	join(item)	배열 내의 요소를 구분문자 del로 연결
	slice(start [,end])	start ~ end-1 번째의 요소를 빼냄
	splice(start, cnt [,rep [, ...]])	배열 내의 start+1 ~ start+cnt+1 번째 요소를 re, ... 치환
추가/삭제	pop()	배열 끝의 요소를 취득하여 삭제
	push(item)	배열 끝에 요소를 추가
	shift()	배열 선두의 요소를 취득하여 삭제
	unshift(data1, [, data2, ...])	배열 선두에 지정 요소를 추가
정렬	reverse()	역순으로 정렬 (반전)
	sort([fnc])	요소를 오름차순으로 정렬
기타	length	배열의 길이
	toString()	[element, element...] 형식의 문자열로 치환

배열 객체 메서드(Array Method)

- `slice()`: 배열 객체에 데이터 중 원하는 인덱스 구간만큼 잘라서 배열 객체로 가져옴
- `splice()`: 배열 객체에 지정 데이터를 삭제하고 그 구간에 새 데이터를 삽입할 수 있음
- `join(연결문자)`: 배열 객체에 데이터를 '연결문자' 기준으로 1개의 문자형 데이터로 반환
- `reverse()`: 배열 객체의 데이터의 순서를 거꾸로 바꾼 후 반환

classList

- 클래스를 조작하는 다양한 메서드들을 쓸 수 있음

- 구문

```
element.classList.메소드('클래스명');
```

- 메소드(Method)

- add : Element에 클래스를 추가

- ex) element.classList.add('open')

- remove : Element에 클래스를 제거

- ex) element.classList.remove('open')

- toggle : Element에 클래스가 존재하면 제거하고 없으면 추가

- ex) element.classList.toggle('open')

- contain : Element에 클래스가 존재하는지 확인

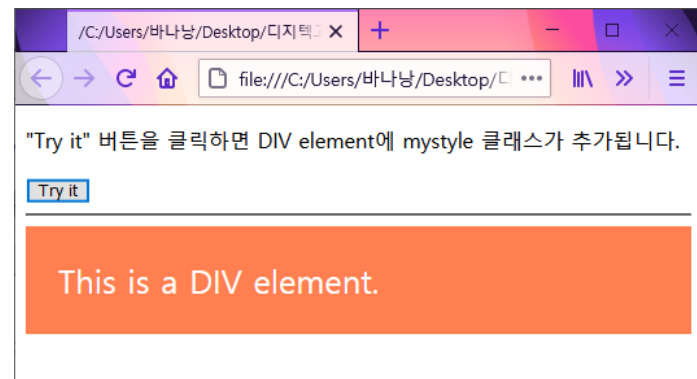
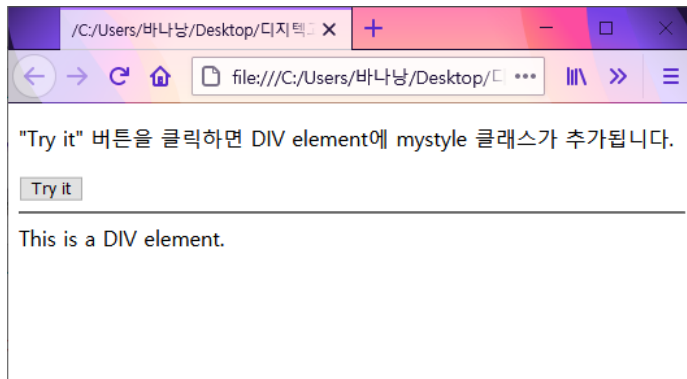
- ex) element.classList.contains('open')

- replace : Element에 클래스를 새로운 클래스로 교체

- ex) element.classList.replace('open', 'close') open클래스를 close로 교체

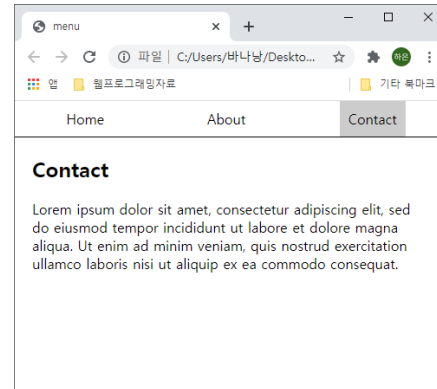
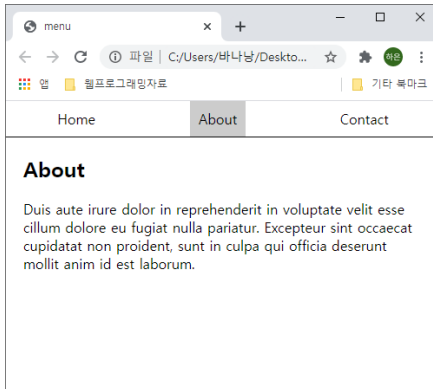
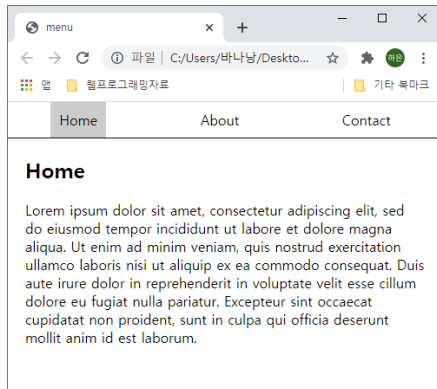
classList & split

- 버튼을 클릭하면 하단의 내용에 스타일이 적용되는 페이지
 - ▣ classList or className 사용: div태그에 myStyle class추가
 - ▣ split() 사용: split을 사용해 배열을 만듦
 - ▣ indexOf 사용: 해당하는 구문에 myStyle클래스가 있는지 확인



간단한 메뉴 만들기

- 각 메뉴 탭을 클릭하면 해당하는 내용이 출력되는 페이지
 - ▣ classList 사용하기(add, remove)
 - ▣ querySelector사용



Thank You!

