



Markup Language
Content



Style sheet Language
Presentation



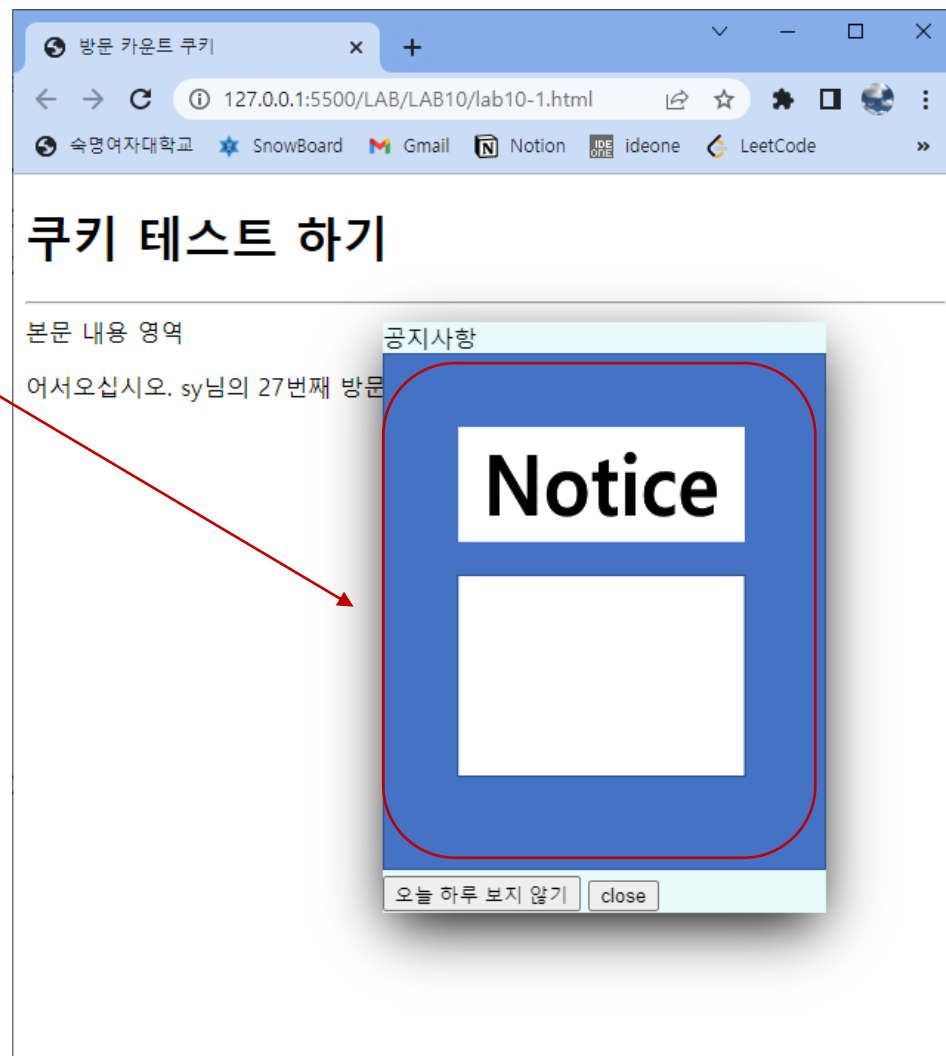
Programming Language
Behavior

Lab 7. 쿠키&스토리지 실습

각 실습 파일들은 압축하지 않고 개별 파일로 첨부해주세요.

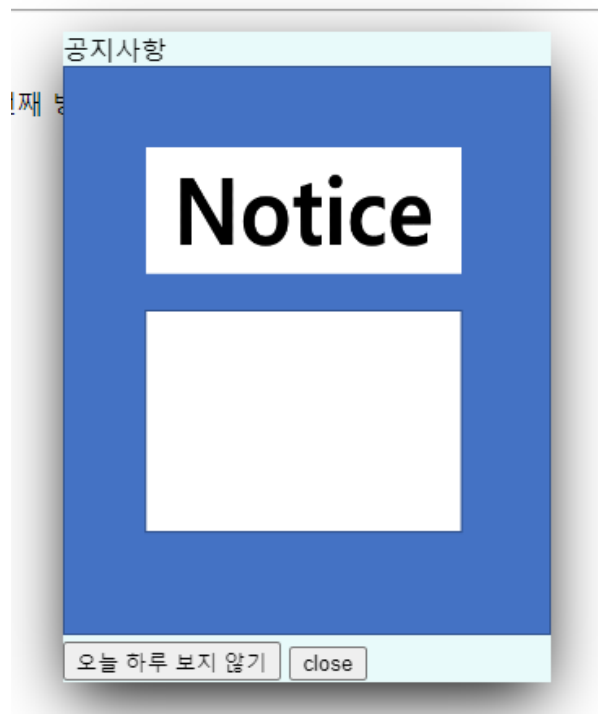
실습 1> 쿠키 실습

- 기존 lab7-1.html에 추가로 코드 작성하기
- div 로 팝업 레이어 구성하기
 - 내용 구성
 - “공지사항” 텍스트
 - Notice 관련 이미지(임의 설정)
 - 버튼 2개
 - 오늘 하루 보지 않기
 - close
 - 필요한 css 작성하기
 - 크기, 위치, 등
 - 버튼 2개에 대한 javascript 작성하기



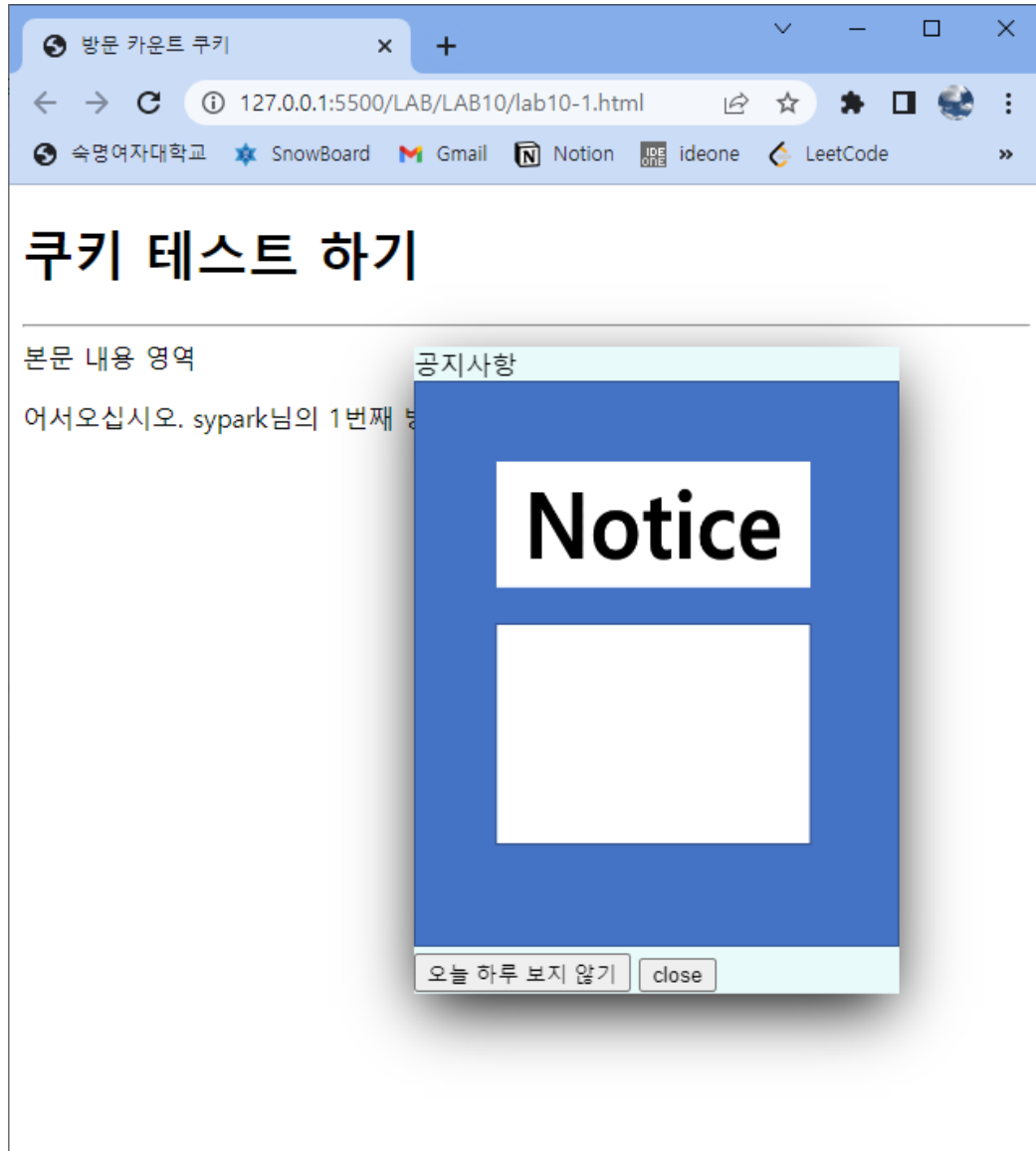
css 파일 내용 추가

- 서식 추가 가능
- 각자가 원하는 값으로 설정



```
.popup{  
    position: fixed;  
    width:      ;  
    height:     ;  
    z-index:    ;  
    box-shadow:      ;  
    display: block;  
    top:        ;  
    left:       ;  
    background-color: ;  
}  
img{  
    width:      ;  
    height:     ;  
}
```

처음 화면 실행 했을 때



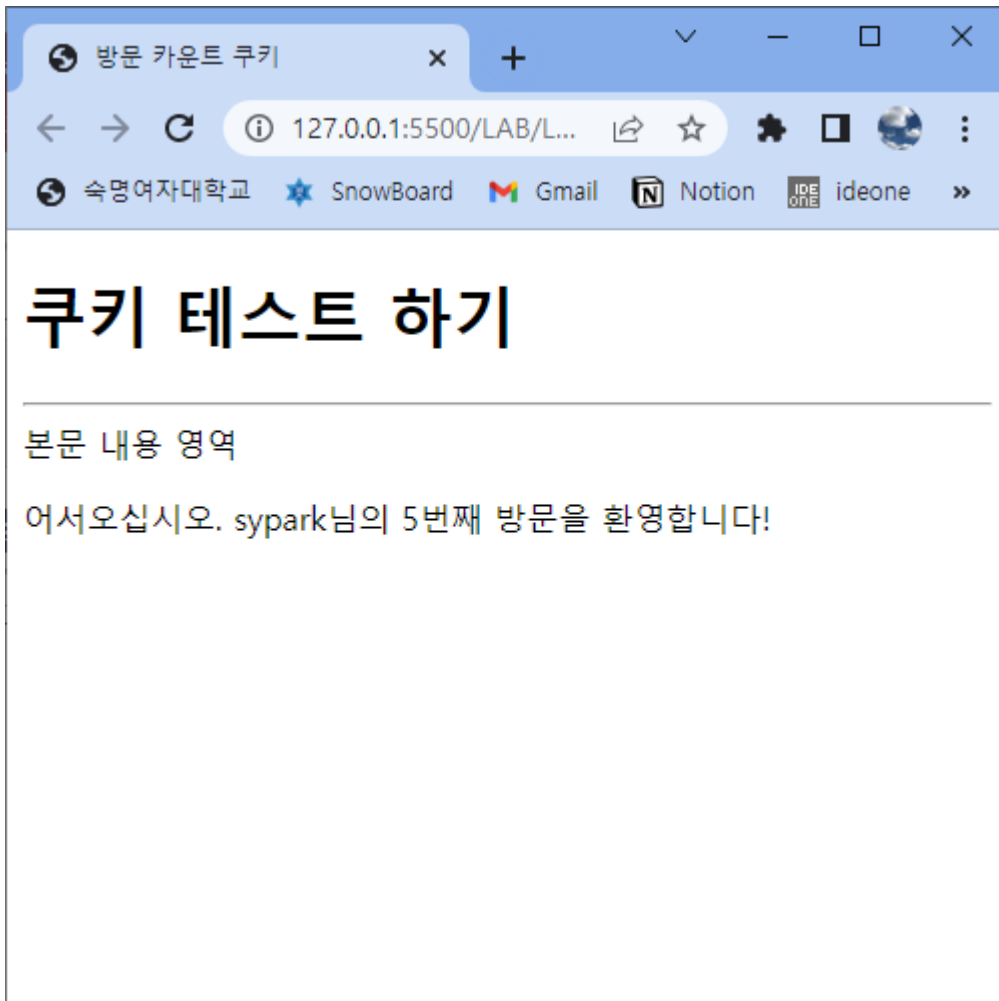
Name	Value	Domain	Path
count	1	127.0.0.1	/LA...
username	sypark	127.0.0.1	/LA...

■ close 버튼 클릭

- 해당 팝업 창이 사라짐.
 - display 를 "none" 으로 설정

“오늘 하루 보지 않기” 버튼 클릭

- 팝업창이 사라짐
- 이 후 화면을 계속 새로고침 하더라도 공지사항 팝업창이 뜨지 않음.



Name	Value	Domain	Path	E...	S...
count	5	127.0.0.1	/LA...	2...	6
today	y	127.0.0.1	/LA...	2...	6
username	sypark	127.0.0.1	/LA...	2...	14

- 해당 쿠키 값(today == “y”)이 설정되어 있으면
 - 팝업창이 보이지 않도록 함.
- 해당 쿠키 값이 설정되어 있지 않으면
 - 해당 쿠키 값을 설정 (SetCookie)
 - 팝업창이 보이도록 함.
- 쿠키 이름과 값은 변경해도 됨

```

<style>
    /* 팝업창 스타일 작성 */
</style>
</head>
<body>
    <h1>쿠키 테스트 하기</h1>
    <hr>
    <div>본문 내용 영역</div>
    <div class="popup">
        <div class="popup_contents">
            <!--공지사항 내용 추가-->
        </div>
        <div class="btn">
            <!-- "오늘 하루 보지 않기 " 버튼 추가 + 함수 연결-->
            <!-- "Close" 버튼 추가 + 함수 연결-->
        </div>
    </div>

```

```
<script>
```

```

    let username = GetCookie("username");
    let count = GetCookie("count");
    // 오늘 하루 보지 않기 기능을 위한 쿠키 추가
    let expire = new Date (); // 현재 시간
    // 오늘 하루 보지 않기 기능을 위한 쿠키 만료 시간 추가
    if (username == null) {
        count = 0;
        username = prompt("이름을 입력해 주십시오.", "");
        if (username == null) {
            alert("이름을 입력하시면 보다 나은 서비스를 제공받을 수 있습니다.");
            username = "아무개";
        } else {
            expire.setTime(expire.getTime() + (365 * 24 * 3600 * 1000)); // 1년
            SetCookie("username", username, expire);
        }
    }
    count++;
    expire.setTime(expire.getTime() + (365 * 24 * 3600 * 1000)); // 1년후
    SetCookie("count", count, expire);
    document.write("<p>어서오십시오. "+username+"님의 "+count+"번째 방문을 환영합니다!");

```

```
// 오늘 하루 보지 않기 기능 실행용 함수 작성
```

```
// 팝업창 닫기 기능 실행용 함수 작성
```

```
// 오늘 하루 보지 않기 기능 실행
```

```
</script>
```

(선택)

- 쿠키를 이용한 count 증가하는 기존 코드를 수정해서 session 이 만료되었을 때만 카운트를 증가하도록 변경해 보시오.
 - 창을 닫고 다시 열었을 때는 count 증가
 - 해당 창에서 refresh만 할 때는 count 증가되지 않도록 함.

실습 2> 스토리지 실습

■ 키와 값을 스토리지에 저장

- 기존 실습 코드 참고

■ 키와 객체를 스토리지에 저장

- 다음 슬라이드에 나오는 실습 코드 작성

Application

- Manifest
- Service Workers
- Storage

Storage

- Local Storage

Filter

Key	Value
사과	5
바나나	10

1 5

Storage

- Local Storage
 - http://127.0.0.1:5500/
- Session Storage
- IndexedDB
- Web SQL
- Cookies
 - http://127.0.0.1:5500
- Private State Tokens
- Interest Groups

Filter

Key	Value
favorites	["pancakes","cupcakes","icecream"]

▼ ["pancakes", "cupcakes", "icecream"]
0: "pancakes"
1: "cupcakes"
2: "icecream"

코드 작성

```
<head>
<style>
.list li {
    cursor: pointer;
}

.list li:hover:after, .list li.fav:after {
    content: ' \2605';
    color: rgb(155, 203, 125);
}

.list li.fav:hover:after {
    content: ' \2606';
}
</style>
</head>
```

```
<body>
    <ul class="list">
        <li id="pancakes">Pancakes</li>
        <li id="donuts">Donuts</li>
        <li id="cupcakes">Cupcakes</li>
        <li id="icecream">Icecream</li>
        <li id="cookies">Cookies</li>
        <li id="chocolate">Chocolate</li>
    </ul>
```

```
<script>
  var favorites = JSON.parse(localStorage.getItem('favorites')) || [];

  favorites.forEach(function(favorite) {
    document.getElementById(favorite).className = 'fav';
  });

  document.querySelector('.list').addEventListener('click', function(e) {
    var id = e.target.id;
    var item = e.target;
    var index = favorites.indexOf(id);

    if (index == -1) {
      favorites.push(id);
      item.className = 'fav';
    } else {
      favorites.splice(index, 1);
      item.className = '';
    }
    localStorage.setItem('favorites', JSON.stringify(favorites));
  });
</script>
```

JSON(JavaScript Object Notation)

값이나 객체를 나타내주는 범용 포맷

JSON 관련 메소드

JSON.stringify

객체를 JSON으로 바꿔준다.

JSON.parse

JSON을 객체로 바꿔준다.

실행 결과

The screenshot shows a web browser window with the address bar at `127.0.0.1:5500/LAB/LAB10/lab`. The browser's address bar and tabs are visible at the top. Below the browser window, there are two panels. The left panel displays a list of desserts: Pancakes, Donuts, Cupcakes, Icecream, Cookies (marked with a green star), and Chocolate. The right panel shows a 'Storage' inspection tool. The 'Local Storage' section is expanded, showing a single entry for `http://127.0.0.1:5500/`. The 'Cookies' section is also expanded, showing a single entry for `http://127.0.0.1:5500/`. The 'Storage' tool's 'Filter' input is empty. The 'Key' and 'Value' columns are visible. The 'favorites' key has a value of `["pancakes","cupcakes","icecream"]`. Below this, the array is expanded, showing the following values:

- 0: "pancakes"
- 1: "cupcakes"
- 2: "icecream"