



Chapter 10

웹 스토리지 (Web Storage)



■ 웹 스토리지(web storage)

- 클라이언트에 간단한 정보를 저장하기 위한 저장 영역
 - 역할은 쿠키와 비슷하지만 크기에 제한이 없음
 - : 쿠키의 경우 4K에 불과함
 - : HTML5 관련 명세서에는 웹 스토리지는 용량을 5M의 크기로 권장하고 있음
 - 유효기간도 존재하지 않음
 - : 웹 스토리지 데이터는 세션에 의존할 수 있음(세션 스토리지)
 - : 경우에 따라 사용자가 제거하지 않는 한 지속적으로 유지될 수 있음 (로컬 스토리지)
- 저장되는 데이터는 키와 값으로 구성되어 있음





■ 세션 스토리지(session storage)와 로컬 스토리지(local storage)

- 세션 스토리지(session storage)

- 세션 기간만 데이터가 유지되는 스토리지
- 윈도우를 새로고침 하거나 동일한 웹사이트에서 새로운 페이지가 로드될 경우에도 정보 유지에 사용

- 로컬 스토리지(local storage)

- 저장된 데이터는 영구하게 보존되며 데이터를 생성한 어플리케이션은 언제든지 사용 가능

- 세션 스토리지나 로컬 스토리지는 동일한 인터페이스로 동작

- 거의 유사한 API를 사용





■ 세션 스토리지(session storage)

- 세션 기간만 데이터가 유지되는 스토리지
 - 저장한 데이터는 영구적으로 보관하지 않고 세션이 종료되면 자동 폐기됨
- 도메인마다 저장 영역이 따로 생성되며 각 세션마다 따로 생성됨
 - 같은 도메인이라도 다른 윈도우에서 생성되면 서로의 스토리지에 접근이 불가능 함
- 기능에 따른 세션 스토리지의 메서드

구분	메서드	설명
데이터의 생성(저장)	setItem(key, value)	데이터(value)을 생성 (모든 데이터는 key가 필요함)
데이터의 추출	getItem(key)	setItem()으로 생성한 데이터를 읽어 들임
데이터의 참조	length	스토리지에 존재하는 데이터들의 수를 반환
	key(item)	특정 key를 가지는 데이터를 반환
데이터의 삭제	removeItem(key)	특정 key를 가진 데이터를 제거
	clear()	스토리지 내의 모든 아이템을 제거





■ setItem() 메서드

- 웹 스토리지에 데이터를 저장하기 위한 메서드
 - 변수를 의미하는 key와 값을 의미하는 data를 인자로 가짐

```
sessionStorage.setItem(key, data)  
sessionStorage.key = data  
sessionStorage[key] = data
```

- key가 이미 존재하는 경우
 - : 새로 지정된 data로 대체됨 (데이터를 수정하는 방법)

```
sessionStorage.setItem("computer", "programming");  
sessionStorage.computer = "programming";  
sessionStorage["computer"] = "programming";
```





■ getItem() 메서드

- 웹 스토리지에 저장되어 있는 데이터를 추출하기 위한 메서드
 - Key만을 인자로 가지며, 생략불가능함

```
변수 = sessionStorage.getItem(key);  
변수 = sessionStorage.key  
변수 = sessionStorage[key]
```

```
var value = sessionStorage.getItem("computer");  
var value = sessionStorage.computer;  
var value = sessionStorage["computer"];
```





■ setItem() 메서드와 getItem() 메서드의 예

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">
    window.onload = function()
    {
      var count=sessionStorage.getItem("counter")
      if (count == null) {
        sessionStorage.setItem("counter", 1);
        alert("처음 방문해 주셨습니다.");
      } else {
        alert(count + "번째 방문해 주셨습니다.");
        count++;
        sessionStorage.setItem("counter", count);
      }
    };
  </script>
</head>
<body> </body>
</html>
```

sessionStorage → localStorage

모든 sessionStorage를 localStorage로 변경해 실행해 봄

이 페이지 내용:

처음 방문해 주셨습니다.

확인

이 페이지 내용:

1번째 방문해 주셨습니다.

☐ 이 페이지가 추가적인 대화를 생성하지 않도록 차단합니다.

확인





■ length 속성

- 웹 스토리지에 저장되어 있는 모든 데이터의 수를 반환
 - 자바스크립트 배열의 length 속성과 같은 동작
 - 주로 조건 지정에 사용됨

```
sessionStorage.length
```

■ Key() 메서드

- 인덱스를 사용해 특정 데이터에 접근하는 기능을 수행
 - 모든 데이터는 순차적으로 저장되며 0부터 시작하는 인덱스를 가짐

```
sessionStorage.key(index)
```





■ removeItem() 메서드

- 특정 데이터를 제거하는 기능을 수행
 - setItem()에서 지정한 key를 인자로 사용

```
sessionStorage.removeItem(key)
```

```
sessionStorage.removeItem("computer");
```

■ Clear() 메서드

- storage 내의 모든 데이터를 삭제함

```
sessionStorage.clear()
```





```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">

  <style>
    #formbox{ float:left; padding:20px; border:1px solid #999999; }
    #databox{
      float:left; width:400px;
      margin-left:20px; padding:20px;
      border:1px solid #999999;
    }
    #keyword, #text{ width: 200px; }
    #databox > div{ padding:5px; border-bottom:1px solid #999999; }
  </style>

  <script>

    function initiate(){
      var button=document.getElementById('save');
      button.addEventListener('click', newitem, false);
      show();
    }
  </script>
</head>
<body>
```





```
function newitem() {
    var keyword=document.getElementById('keyword').value;
    var value=document.getElementById('text').value;

    sessionStorage.setItem(keyword,value);
    show();

    document.getElementById('keyword').value="";
    document.getElementById('text').value="";
}

function show() {
    var databox=document.getElementById('databox');
    databox.innerHTML=<div><button onclick="removeAll()">erase everything</button></div>;

    for(var f=0;f<sessionStorage.length;f++){
        var keyword=sessionStorage.key(f);
        var value=sessionStorage.getItem(keyword);
        databox.innerHTML+='<div>'+keyword+' - '+value+'<br>
        <button onclick="remove(\"'+keyword+'\")">remove</button></div>';
    }
}
```





```
function remove(keyword){
    if(confirm('Are you sure?')){
        sessionStorage.removeItem(keyword);
        show();
    }
}

function removeAll(){
    if(confirm('Are you sure?')){
        sessionStorage.clear();
        show();
    }
}

window.addEventListener('load', initiate, false);

</script>
</head>
```



웹 스토리지의 예 [계속]



```
<body>

  <section id="formbox">
    <form name="form">
      <p>Keyword:<br><input type="text" name="keyword" id="keyword"></p>
      <p>Value:<br><input type="text" id="text"></p>
      <p><input type="button" name="save" id="save" value="Save"></p>
    </form>
  </section>
  <section id="databox">
    No Information available
  </section>
</body>
</html>
```





■ 로컬 스토리지 (local storage)


- 데이터 저장 시간에 제한이 없어 사용자가 지우지 않는 한 영구적으로 보관
 - 도메인마다 저장 영역이 따로 생성
 - 도메인마다 생성된 로컬 스토리지에 서로 접근할 수 없음
- 동일한 도메인에 속해 있는 웹 페이지들은 모두 접근이 가능
- 관련 메서드 및 속성은 sessionStorage와 동일함





■ 웹 스토리지의 데이터 확인 방법

- 브라우저가 마다 가지고 있는 개발자 도구를 통해 세션 스토리지나 로컬 스토리지에 존재하는 데이터들을 확인할 수 있음
- **chrome** 브라우저 경우의 데이터 확인 예
 - 설정/ 도구 더보기/ 개발자 도구/ Application에서 확인 가능



Application		Key	Value
Application		일본	동경
Manifest		중국	북경
Service Workers		한국	서울
Clear storage			
Storage			
▶ Local Storage			
▼ Session Storage			
file://			
IndexedDB			
Web SQL			
▶ Cookies			