

웹응용프로그래밍

김태운

부산대학교 정보의생명융합대학 정보컴퓨터공학부

목차

- Q1 : PHP Classes
- Q2 : Cookie
- Q3 : jQuery
- 데드라인 :
 - 1차 데드라인(수업시간 중) : 12/12(화) 1:30~2:45
 - 2차 데드라인 : 12/17(일) 23:55



Q1. PHP Classes

Q1. PHP Classes

- XAMPP에서 Apache 서비스를 시작하세요
- 배포된 Q1/sampleCode/artist-class-tester.php를 웹 브라우저에서 접속하여 화면을 표시하세요 →

Tester for Artist class	
Artists	
Pablo Picasso	
Birth:	25 Oct 1881(Malaga)
Death:	08 Apr 1973
Salvador Dali	
Birth:	11 May 1904(Figures)
Death:	23 Jan 1989
Number of Instantiated artists: 2	

- 해당 페이지는 Artist 클래스를 사용하는 코드입니다.
 - Artist 클래스 정의는 Q1/sampleCode/artist.php 코드를 확인하세요

```
class Artist {  
    const EARLIEST_DATE = 'January 1, 1200';  
    private static $artistCount = 0;  
    private $firstName;  
    private $lastName;  
    private $birthDate;  
    private $deathDate;  
    private $birthCity;  
    // notice constructor is using setters instead  
    // of accessing properties  
    function __construct($firstName, $lastName, $birthDate, $deathDate) {  
        $this->setFirstName($firstName);  
        $this->setLastName($lastName);  
    }  
}
```

Q1. PHP Classes

- 다음으로, Q1/yourCode 폴더로 이동하세요
- Q1/artist-class-tester-extended.php를 웹 브라우저에 표시하여, 우측과 같이 출력되도록 artist-extended.php 파일을 수정하는 과제입니다.

Tester for Artist, PaintingArtist, SculptureArtist class	
Artists	
Pablo Picasso Birth: 25 Oct 1881(Malaga) Death: 08 Apr 1973	
Salvador Dali Birth: 11 May 1904(Figures) Death: 23 Jan 1989	
Painting Artist	
Pablo Picasso Birth: 25 Oct 1881(Malaga) Death: 08 Apr 1973 Painting work: my_painting_1	
Sculpture Artist	
Salvador Dali Birth: 11 May 1904(Figures) Death: 23 Jan 1989 Sculpture work: my_sculpture_1	
Number of Instantiated artists: 4	

Q1. PHP Classes

- Q1/yourCode/artist-class-tester-extended.php 파일은 수정하지 마세요.
- Q1/yourCode/artist-extended.php 파일의 경우,
 - Q1/sampleCode/artist.php 파일에서 Artist 클래스 정의 부분을 그대로 복사하여 붙여넣기 하세요.
 - Artist 클래스를 상속받아 PaintingArtist 및 SculptureArtist 클래스를 정의하세요
 - 필요한 경우, 복사해 온 Artist 클래스를 수정할 수 있습니다.



Q2. Cookies

Q2. Cookies

- 쿠키를 사용하여, 가장 최근에 웹 페이지에 접근한 시간을 저장하고 화면에 출력하는 문제입니다.
- 아래와 같이 동작하도록 Q2/cookie-LastAccessTime.php 파일을 수정하십시오.

- 페이지에 처음 접근할 때:

Checking the last access time using cookie

Current time: 2023/12/11 01:28:02 am

Last access time: unknown

- 페이지에 다시 접근할 때:

Checking the last access time using cookie

Current time: 2023/12/11 01:29:17 am

Last access time: 2023/12/11 01:28:02 am ←

Q2. Cookies

- 다음으로, LastAccessTime을 저장하는 쿠키를 삭제하기 위해(= \$_COOKIE['LastAccess']를 삭제하기 위해) Q2/cookie-Delete.php 파일을 완성 하시오.

cookie-LastAccessTime.php 실행

Checking the last access time using cookie

Current time: 2023/12/11 01:29:17 am
Last access time: 2023/12/11 01:28:02 am



cookie-Delete.php 실행

Cookie delete for \$_COOKIE[LastAccess] is done



cookie-LastAccessTime.php 실행

Checking the last access time using cookie

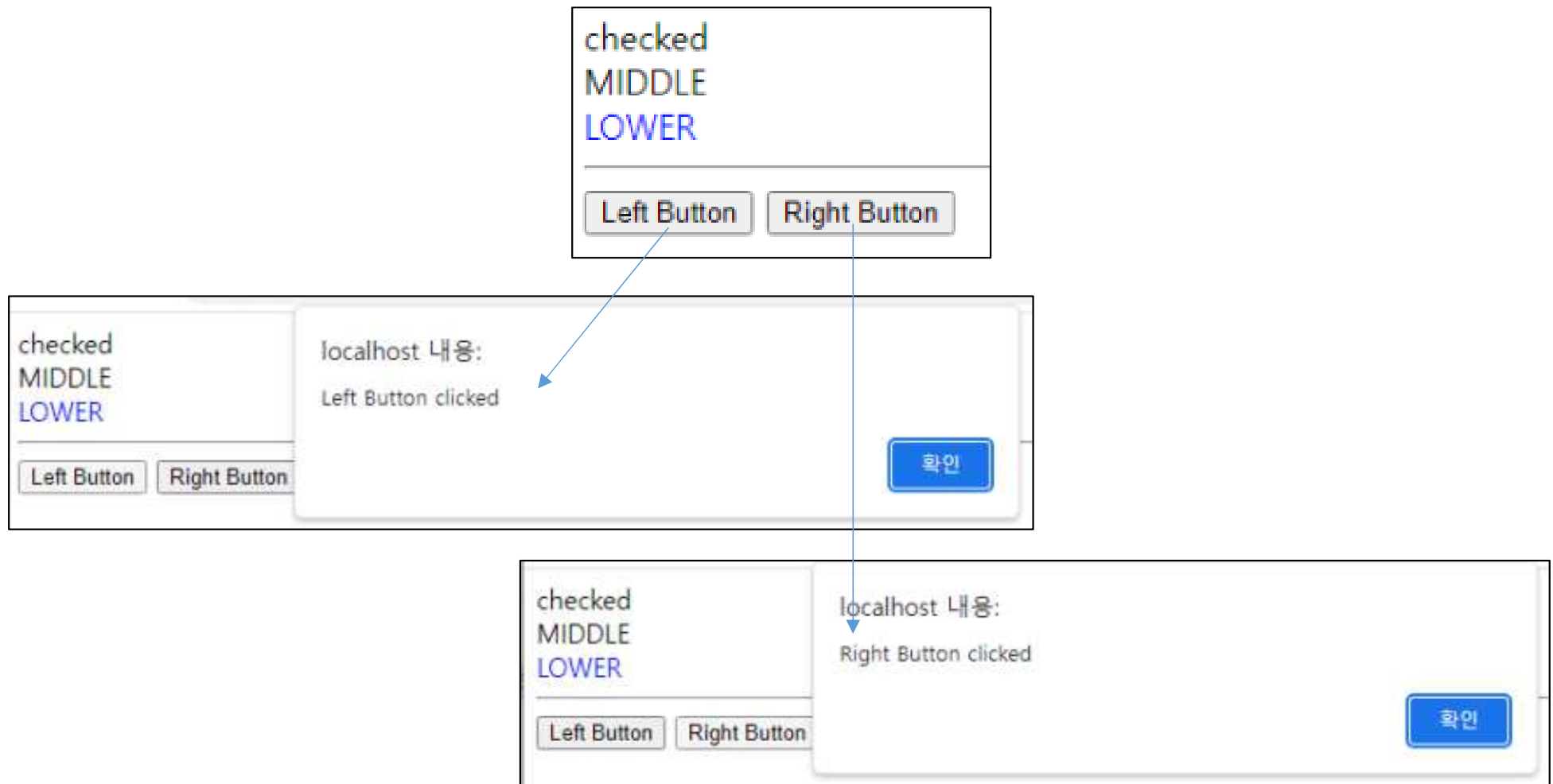
Current time: 2023/12/11 01:32:22 am
Last access time: unknown



Q3. jQuery

Q3. jQuery

- 웹 브라우저에서 Q3/index.html 파일에 접근할 때, 아래와 같이 표시되도록 index.html 파일을 수정하십시오.



Q3. jQuery

- Q3/index.html 파일을 아래와 같이 두 곳을 수정하시오.
 - jQuery 라이브러리 include 하기
 - 라이브러리는 <https://code.jquery.com>에서 다운 받을 수 있습니다.
 - 라이브러리를 다운받은 후 로컬에 저장된 라이브러리를 사용하도록 코딩하거나, 3rd-party CDN에서 라이브러리를 직접 다운받아 사용하도록 코딩할 수 있습니다.
 - 두 개의 버튼에 onclick 이벤트 핸들러를 등록하고, 클릭 시 alert 윈도우가 팝업되도록 코딩하세요.
 - 이 때, 이벤트 핸들러는 웹 페이지의 로딩이 완료된 이 후 등록하도록 코딩하세요.
 - 즉, `$(document).ready` 이벤트 핸들러 함수에서 버튼 이벤트 핸들러를 등록하세요.

THE END