연습 문제 이론 정답

연습문제 01

EXERCISE

이론문제

1. ③ 인터넷에 연결되는 PC는 210.1.1.2와 같이 4 개의 숫자로 구성되는 IP주소를 가진다.

2. ② 키보드, 마우스, 모니터가 없는 웹 서버 컴퓨터는 없다.

3. 웹 브라우저들이 세상에 나온 순서를 시간 순으로 나열하면 다음과 같다.

   WorldWideWeb 〉 Netscape Navigator 〉 Internet Explorer 〉 Opera, Filrefox 〉 Safari 〉 Chrome

4. ① Tim Berners-Lee

5. WorldWideWeb

6.

HTML	HTML 문서의 구조와 콘텐츠 표현
CSS	문서의 모양 표현
Javascript	웹 페이지의 동적 변경 및 응용프로그램 작성

7. ③ 하이퍼링크 개념은 전자 문서에서 시작되었다.

8. 과정 1) 웹 브라우저는 www.univ.ac.kr 컴퓨터에 접속한다.
   과정 2) 웹 브라우저는 score.html 파일을 보내 줄 것을 웹 서버에 요청한다.
   과정 3) 웹 브라우저는 받은 웹 페이지를 해독하여 화면에 출력한다.

9. ④ 웹 브라우저들이 저마다 플러그인을 만들어서 경쟁하기 때문이다.

10. ① HTML4가 지원하지 못하였던 이미지와 동영상을 지원할 필요가 있었다.

11. ③ PC의 웹 브라우저에서 연주되도록 작성되었으면 스마트폰에서는 들을 수 없다.

연습문제 02

EXERCISE

이론문제

1. ② <img>

2. ③ <!-- 이것은 주석문 -->

3. ④ src:shrek.png
   바르게 고치면 src="shrek.png"이다. 참고로 HTML 언어에서는 대소문자를 구분하지 않고 단일 인용 부호나 이중 인용 부호를 모두 사용할 수 있으며, 인용부호가 없어도 무관하다.

4. ① <iframe>

5. ① <iframe> ④ <img>

6. ④

7. ④ <img>

8. ② <meta>

9. <meta name="description" content="자바에 대해 알려주려고">

10.
```
<head>
    <base href="http://www.mysite.com/html/design">
</head>
```
```
<a href="dress.html">옷</a>
<a href="shoes.html">신발</a>
```

11.
(1) 유럽 여행에 필요한 물품을 나열할 때 : <ul> (물품에는 순서가 없기 때문)
(2) 공항에서 출국 수속 과정을 나열할 때 : <ol> (출국 수속은 순서가 있기 때문)
(3) 유럽 각 나라들의 특징을 나열할 때 : <dl> (나라와 특징을 묶어서 나열하기 때문)

12. 링크 텍스트에 대한 디폴트 색에 대해 답하라. 링크 텍스트는 브라우저가 처음으로 출력할 때, blue 색으로 출력한다. 하지만, 사용자가 일단 방문하게 된 후에는 purple 색으로 출력하고, 방문하기 위해 마우스를 누르고 있는 동안에는 red 색으로 출력한다.

13. ① <iframe>

14.

(1)

```
<!doctype html>
<html>
<head><title></title></head>
<body>
<h3>나의 이야기</h3>
<hr>
나는 자랑스러운 대한민국의 국민입니다.
</body>
</html>
```

(2)

```
<!doctype html>
<html>
<head><title></title>
</head>
<body>
<br>Merry Christmas!
Happy New Year!
</body>
</html>
```

15. ③ <frameset><frame src="1.html"><frame src="2.html"></frameset>

16.

```
<audio controls>
    <source src="hello.mp3" type="audio/mpeg">
    audio를 지원하지 않습니다.
</audio>
```

17. 웹 브라우저의 공간에 320×240의 크기의 영역을 할당받아 비디오를 출력한다. 비디오의 크기를 이 크기에 맞추어 조절한다. 물론 비디오의 크기는 가로 세로 비율을 일정하게 유지한다. 웹 브라우저가 로딩되면 바로 비디오를 재생하며 재생, 중단, 음량 조절, 음소거 등의 제어 버튼을 함께 출력한다. 현재 브라우저가 mp4 타입의 비디오를 재생할 bear.mp4 파일을 재생하고, 그럴 수 없는 경우 ogg 타입의 비디오를 재생할 수 있으면 bear.ogg 파일을 재생한다. 즉 bear.mp4나 bear.ogg 중 재생 가능한 비디오 파일을 재생하며 bear.mp4를 먼저 선택한다. 만일 브라우저가 <video> 태그를 지원하지 않는 경우, 비디오를 재생하는 대신 '브라우저가 video 태그를 지원하지 않습니다.'라고 출력한다.

연습문제 03

EXERCISE

이론문제

1. ② <img>

2. ①

3. ②

4. ① 웹 페이지 내 가치 있는 정보의 탐색이 원활하도록 하기 위해

5. ③ 검색 엔진이 좋아하는 웹 페이지로 만들기 위해

6. ② <footer>

7. ④ <nav>

8. <form> 태그의 (action) 속성은 웹 서버의 응용프로그램 이름을 지정하며, (method) 는 폼 데이터를 웹 서버로 전송할 때 데이터를 전송하는 방식을 결정한다. 이 방법에는 (get)와 (post) 가 있다. 그리고 (target) 속성에는 폼 데이터를 전송하고 웹 서버로부터 받은 결과를 출력할 윈도우의 이름을 지정한다.

9. (1) <input type="number" min="-5.0" max="5.0" step="0.5">
   (2) <input type="month" value="2017-05>
   (3) <label>자바스크립트 <input type="checkbox"></label>
   (4) <label>오후 7시 <input type="radio"></label>

10. ④ <button type="button"> 버튼은 <form> 없이 사용되어야 한다.

연습문제 04

EXERCISE

이론문제

1.

(1) =을 :로 수정. body { background-color : mistyrose; }
(2) <h3>을 h3으로 수정. h3 { color : purple /* purple은 보라색 */ }
(3) _를 -로 수정. HR { height : 5px; background-color : grey }
(4) ,를 ;으로 수정 span { color : blue; font-size : 20px } // 태그에 적용

2. 다음 CSS3 스타일 시트 작성이 잘못된 것을 바르게 고쳐라.

(1) ()를 {}로 수정. div { font-size : 30px; }
(2) ,를 ;로 수정. 안녕하세요
(3) 5를 5em으로 수정. p { text-indent : 5em } // em 대신 px 등 단위 사용 가능
(4) '' 삭제. p { color : blue }

3. ① div { color : rgb(55, 325, 128); }
설명) rgb 표현법은 색을 나타내는 r,g,b 요소의 숫자가 0~255만 가능하다. green 요소를 나타내는 수 325는 적절치 않다.

4. ④ width, height 프로퍼티로 박스 크기를 제한하면 박스의 콘텐츠가 비록 크더라도 박스 크기를 넘어가지 않고 잘려서 출력된다.
설명) overflow 프로퍼티로 박스 크기를 넘어 출력할지 결정할 수 있다.

5. ③ ##div

6. ① em

7.

```
<!DOCTYPE html>
<html>
<head><title>CSS3</title>
<style>
  p {
      color : blue;
      text-align : center;
  }
</style>
</head>
<body>
<p>test</p>
</body>
</html>
```

```
/* 스타일 시트를
style.css에 저장 */

p {
    color : blue;
    text-align : center;
}
```

```
<!-- 수정된 HTML파일 -->

<!DOCTYPE html>
<html>
<head><title>CSS3</title>
<style>
@import url(style.css);
</style>
</head>
<body>
<p>test</p>
</body>
</html>
```

8.

(1) color : red; text-align : right; font-size : 3em;
(2) color : green; text-align : right; font-size : 2em

9.

③ div > span { color : blue; }
설명) > 기호는 직계 자식의 관계를 나타내는 것인데, 은 의 직계 자식이 아니다.

10.

④ div.a { color : blue; }
설명) div > a로 해야 한다.

11.

(1)

```
a {
    text-decoration : none;
    color : blue;
}
```

(2)

```
a:hover {
    font-size : 2em;
}
```

(3)

```
a:visited {
    color : violet;
}
```

12.

① class 속성이 “menu”인 모든 <li class="menu"> 태그에 적용된다.

연습문제 05

EXERCISE

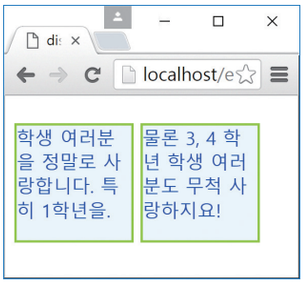
이론문제

1. ① 정적 배치

2. ② HTML 태그를 개발자가 원하는 위치에 임의로 배치할 수 있다는 의미

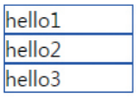
3.

```
p {
border : 2px solid yellowgreen;
color : blue; background : aliceblue;
}
...
<div>
<p style="display:inline-block; height:100px; width:100px">
학생 여러분을 정말로 사랑합니다. 특히 1학년을.</p>
<p style="display:inline-block; height:100px; width:100px">
물론 3, 4 학년 학생 여러분도 무척 사랑하지요!</p>
</div>
```



4. ② 는 DIV 와 동일하다.
설명) 는 디폴트로 블록 박스이므로 DIV 와 동일하다.

5. 태그는 디폴트가 블록 박스이다.
(1) div { border : 1px solid blue; width : 100px }

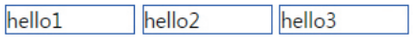


설명) 태그는 블록 박스이므로 하나의 태그가 한 줄을 완전히 차지한다. 그러므로 태그는 새로운 줄에서 시작하고 옆에는 다른 태그가 배치되지 못한다. 하지만 폭이 100px로 주어졌기 때문에 실제 차지하는 영역은 폭이 100 픽셀이다.

(2) div { display : inline; border : 1px solid blue; width : 100px }

설명) 태그를 인라인 박스로 바꾼 경우이다. 인라인 박스이므로, 하나의 태그는 콘텐츠의 크기만큼만 공간을 차지하며 width:100px의 스타일은 작동하지 않는다. 양 옆에는 다른 인라인 박스들이 나열 가능하다.

(3) div { display : inline-block; border : 1px solid blue; width : 100px }



설명) 태그를 인라인 블록 박스로 바꾼 경우이다. 인라인 박스의 속성이 반영되어 양옆에는 다른 인라인 박스나 인라인 블록 박스가 배치 가능하다. 또한 블록 박스의 속성이 반영되어 width:100px의 스타일에 따라 태그는 옆으로 100px의 공간을 차지한다.

6.

(1) img { visibility : hidden } 또는 img { display : none }
(2) img { display : block }
(3) img { width : 400px; height : 300px }

7. ② input[type=password] { background : yellow }

8.

(1) input[type=button] { color : blue }
(2) input[type=button]:hover { background : yellow }
(3) input[type=button]:focus { background : yellow }

9.

(1) span { transition : font-size 2s }
(2) img { transition : width 3s }

10.

(1) 180도 회전

```
#tran:hover {
    transform : rotate(180deg);
}
```

(2) y축으로 -20도 기울임

```
#tran:hover {
    transform : skew(0deg, -20deg);
}
```

(3) 90도 회전하고 1:3비율 확대

```
#tran:hover {
    transform : rotate(90deg) scale(1,3);
}
```

연습문제 06

EXERCISE

이론문제

1. ① 자바스크립트 프로그램은 컴파일 과정이 간단한다.

2. ③ 로컬 컴퓨터의 파일을 삭제할 수 있다.

3. ff, 4+5

4. ③ "1.5px".
   설명) 먼저 나온 산술 식 3/2를 계산하여 결과가 1.5이므로, 1.5+"px"를 처리하여 "1.5px"

5. ② loop

6. ④ case x>6:

7. (1) w=1, x=2, y=1, z=5
   (2) w=true, x=false, y=true, z=16

8. (1) s=3hello3, t=1hello23
   (2) s=false, t=false

9. (1) 지역 변수 : y, 전역 변수 : x, z
   (2) 지역 변수 : 없음, 전역 변수 : sum, x, n

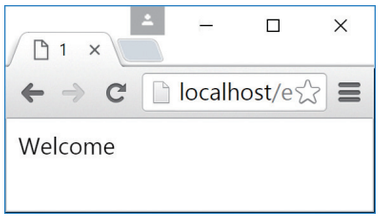
10. (1) '합 = 0' 이 출력된다.
    설명) acc() 함수는 1에서 10까지 더하기를 실행하여 acc() 함수 안에 선언된 로컬 변수 sum 에 값을 저장한다. 하지만 document.write()에서는 전역 변수 sum을 출력하므로 0이 출력된다.

    (2) '합 = 45'가 출력된다.
    설명) acc() 함수는 this.sum에 1에서 10까지 더하기를 실행한다. this.sum은 전역변수를 sum을 가리킨다. 그러므로 document.write()에서 전역 변수 sum을 출력하면 45가 출력된다.

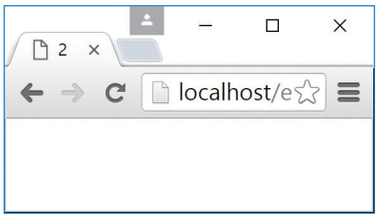
11. ② 53/10 = 5
    설명) 자바스크립트에서 나누기는 실수 연산이므로 연산의 결과 값은 5.3

12.

(1) 다음과 같이 브라우저에서는 Welcome이 출력된다.



(2) 다음과 같이 브라우저에서는 아무것도 출력되지 않는다.



그 이유는 print() 함수가 호출되고 있지 않기 때문이다. 함수는 호출되어야만 실행되는 코드 블록이다.

13.

```
var a=20, b=2;
```

식	a	b
(1) a += b;	22	2
(2) a = a << b;	80	2
(3) a = (a>100)?a:b;	2	2
(4) b = (a>10)&&(a<30);	20	true

14.

```
var a="123", b="45";
```

식	a	b
(1) a += b;	"12345"	"45"
(2) a = parseInt(a) + parseInt(b);	168	"45"
(3) a = 10 + a + b;	"1012345"	"45"
(4) b = 10 + parseInt(a) + "ab";	"123"	"133ab"

연습문제 07

EXERCISE

이론문제

1. ③ new

2. ④ var value = [1, 2, 3];

3. ④ months.length = 5로 지정하여 배열의 크기를 5개로 늘일 수 없다.
   설명) Array의 length는 쓰기가능한(writable) 프로퍼티로서, months.length = 5로 지정하면 배열의 크기를 5로 늘일 수 있으며, 늘어난 공간은 undefined 상태로 비어 있다.

4. ② grades[3] = 70;은 잘못된 코드이다. 왜냐하면 문자열 배열에 정수를 넣기 때문이다.
   설명) 자바스크립트에는 변수나 배열에 저장되는 값에 특별한 제약이 없어 아무값이나 들어갈 수 있다.

5.
```
var money = new Array(3);  // Array를 이용하여 크기가 3인 배열 money 생성
money[0] = 5;      // money의 첫 번째 원소에 5 삽입
money[1] = 7;      // money의 두 번째 원소에 7 삽입
money[2] = -3;     // money의 세 번째 원소에 -3 삽입
var sum = 0;
for(i=0; i< money.length ; i++) sum += money[i];  // 배열 합 구하기
document.write( sum/money.length );  // 평균 출력
```

6. ④ new Array()로 생성한 객체는 []로 생성한 배열과 약간 다르다.

7. (1) 15
   (2) "b"
   (3) 2
   (4) "HTML5 Programming"
   (5) "P"

8. Math.random()은 0에서 1보다 작은 임의의 실수를 리턴하므로, Math.random()*10+1;은 1.0에서 10.9999... 사이의 임의의 실수를 만들기 때문이다. 바르게 수정하면 다음과 같다.

```
var x = Math.floor(Math.random()*10)+1;
```

Math.random()*10의 결과 값에서 소숫점 이하를 제거한 정수를 만들기 위해 Math.floor() 함수를 활용한다. 그 결과 Math.floor(Math.random()*10)는 0에서 9를 포함하는 임의의 정수가 되고 여기에 1을 더하면 1에서 10을 포함하는 임의의 정수가 된다.
다음과 같이 해도 된다.

```
var x = Math.floor(Math.random()*10+1);
```

9.

```
var student = {
    id : 1,
    name : "kitae",
    grade : 3.9
};
```

10.

(1)

```
function fill() {
    this.amount += 2;
}

function consume() {
    this.amount -= 2;
}

var box = new Object();
box.color = "red";
box.size = 10;
box.amount = 0;
box.fill = fill;
box.consume = consume;
```

(2)

```
box.fill();
document.write(box.amount);
box.consume();
document.write(box.amount);
```

(3) box.fill(); 후 document.write(box.amount);의 실행 결과 2가 출력되고, box.consume(); 후 document.write(box.amount);의 실행 결과 0이 출력된다.

연습문제 08

EXERCISE

이론문제

1. ④ html 객체

2. ② document 객체

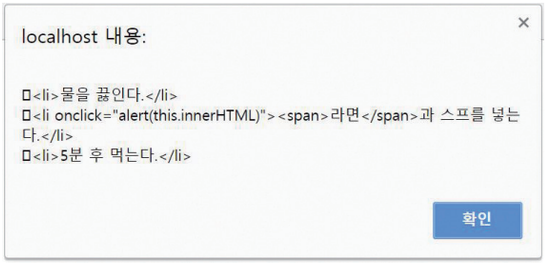
3. ④ DOM 트리에서 DOM 객체를 찾는 기능은 window.getElementById()이다.

4. HTML DOM 트리를 만드는 이유는 HTML 문서의 각 요소를 객체화하고 이 객체들을 계층관계로 연결하여 유지함으로써, HTML 페이지가 로드된 이후 자바스크립트 코드로 HTML 태그 요소를 접근하고 동적인 변화를 줄 수 있도록 하기 위해서이다. 또한 HTML 페이지가 이미 로드된 이후에도 DOM 트리에 새로운 HTML 요소를 삽입하거나 삭제하여 출력된 웹 페이지의 모양을 제어하기 위해서이다.

5. ② obj.style.backgroundcolor = "red";
설명) 바르게 고치면 obj.style.backgroundColor = "red"; 이다.

6. ② getElementByTagName()

7. (1) 라면과 스프를 넣는다.
(2) 다음과 같이 출력된다. 즉 와 사이의 모든 HTML 콘텐츠가 문자열로 출력된다.



8. alert() 함수가 실행되고 경고창이 출력된다. 그 이유는 HTML 문서가 로딩되는 초기에 <script> 내의 var elem = document.getElementById("myBody");코드가 실행된다, 하지만 아직 브라우저는 <body>...</body> 태그를 읽지 못한 상태이며 따라서 body 객체가 생성되지 않았다. 그러므로 document.getElementById("myBody") 메소드가 null을 리턴하여 if(elem == null)의 조건문이 true가 되고 alert() 함수가 실행된다.

9.

```
var div = document.getElementById("myDiv");   // id가 myDiv인 DOM 객체를 알아냄
var span = document.createElement("SPAN");    // <span> 태그를 나타내는 DOM 객체 동적 생성
span.innerHTML = "hello";                     // "hello"를 <span> 태그의 텍스트로 삽입
div.appendChild(span);                        // span을 div의 자식으로 삽입
```

연습문제 09

EXERCISE

이론문제

1. ③ 이벤트는 반드시 이벤트 타겟 객체에서만 처리된다.

2. ③ ONE, TWO, THREE 문자열을 가진 3 개의 경고창이 각각 순서대로 출력된다.
설명) 하나의 DOM 객체에 동일한 이벤트에 대해 여러 개의 리스너를 등록할 수 있다. 그들은 등록한 순서대로 실행된다.

3. ① div1.addEventListener("onclick", function() { this.innerHTML ="hello"; });
설명) onclick을 click으로 수정해야 한다.

4.
(1) var obj = document.getElementById("div1");
(2) obj.addEventListener("click", f);
(3) obj.onclick = f;
(4) obj.onclick = function (e) { obj.style.backgroundColor="orchid"; };

5. ① window 객체에서부터 타겟 객체까지

6. ③ 이벤트 버블이나 이벤트 캡쳐 단계 중 한 단계에서만 이벤트 리스너 작성 가능

7. window.addEventListener("click", function(e) { e.stopPropagation(); }, true);는 캡쳐 단계에서 function(e) { e.stopPropagation(); }가 실행되도록 window 객체에 onclick 리스너를 등록하는 코드이다. HTML 페이지의 어떤 요소에서 click 이벤트가 발생하든 캡쳐 단계를 거치게 되어 있고, 무조건 window 객체에 click 이벤트가 도착한다. 그런데 여기 window 객체에 등록된 onclick 리스너에서는 e.stopPropagation(); 함수를 호출하기 때문에 더 이상 이벤트가 전파되지 못하고 사라진다.

8. ① window.onclick = function(e) { alert(e.type); }
③ window.addEventListener("click", function(e) { alert(e.type) }, false);
④ window.onclick = f; function f(e) { alert(e.type); }
설명) addEventListener() 함수의 마지막 인자가 생략되거나 false이면 버블단계에서 실행되는 리스너를 등록한다. addEventListener() 함수를 이용하지 않고 리스너를 등록하는 나머지 경우는 모두 자동으로 버블단계에서 실행되도록 리스너가 등록된다.

9.

(1)

```
strong = document.getElementById("strong");
strong.onclick = function (e) {
    alert(this.tagName + "+" + e.target.innerHTML);
};
```

(2)

```
p = document.getElementById("p");
p.addEventListener("click", function (e) {
    alert(this.tagName + "+" + e.target.innerHTML); }, true);
```

(3) ② 문제 (2)의 경고창이 출력되고 문제 (1)의 경고창이 출력된다.

10. p.addEventListener("click", function (e) { alert("hello"); }, false);

연습문제 10

EXERCISE

이론문제

1. ② window

2. ② navigator

3.
(1) window.open("test.html", "win");
(2) window.open("http://www.google.com", "_self");
(3) window.open(“http://www.naver.com", "_blank", "width=400,height=500");
(4) window.open("test.html", "_blank", "left=0,top=0");

4.
(1) window.open("", "_blank", "left=100,top=100,width=500,height=600");
(2) window.open("http://www.google.com", "google");

5. ③ window.location.reload("http://www.naver.com");

6. ① navigator

7. ④ window.go(0);

8. ① 위치 정보

9.
(1) ③ 2밀리초 간격으로 f() 함수가 10번 호출된다.

10.

```
var sum=0;
function f() {
    sum++;
    if(sum != 10)
        id = setTimeout("f()", 2); // sum이 10이 아니면 다시 타이머 작동
    // sum이 10이면 타이머 작동시키지 않음
}
var id = setTimeout("f()", 2);
```

연습문제 11

EXERCISE

이론문제

1. ② <canvas>

2. ③ 캔버스는 2차원 그래픽만 지원하고, 3차원은 지원하지 않는다.

3.

```
var canvas = document.getElementById("can");    // (1) 캔버스 DOM 객체를 알아낸다.
var context = canvas.getContext("2d");          // (2) 그래픽 컨텍스트 객체를 알아낸다.
context.beginPath();                            // (3) 경로를 새로 구성한다.
context.moveTo(10, 10);                         // (4) (10, 10)의 시작점을 입력한다.
context.lineTo(50, 50);                         // (5) (50, 50)까지 직선을 만든다.
context.stroke();                               // (6) 경로에 있는 직선을 캔버스에 그린다.
```

4. ④ stroke() 메소드는 경로에 들어 있는 도형을 모두 캔버스에 그리고, 그린 도형은 경로에서 삭제한다.

5. rect() 메소드는 경로에 사각형을 삽입하며 실제 캔버스에 그리는 것은 아니다. 추후 stroke() 메소드가 불려져야 그려진다. 하지만 strokeRect()은 경로에 삽입하지 않고 캔버스에 바로 사각형을 그린다.

6. ② context.clearRect(0, 0, canvas.width, canvas.height);

7. img.src = "elas.png"; 라인은 이미지의 로딩을 시작시키고 이미지의 로딩이 완료된 것을 확인하지 않고 바로 리턴한다. 만일 이미지가 로딩되지 않은 채로 다음 라인 context.drawImage(img, 10, 10);이 실행되면, 이미지를 그릴 수 없게 된다. 이미지가 로딩된 것을 통보받은 시점에 이미지를 그려야 한다. 다음과 같이 수정하여야 한다.

```
var img = new Image();
img.onload = function f() {
    context.drawImage(img, 10, 10);
}
img.src = "elsa.png";
```

8. 문제에 주어진 코드는 다음과 같다.

```
1   var img = new Image();
2       img.src = "elsa.png";
3       img.onload = function f() {
4           context.drawImage(img, 10, 10);
5       }
```

라인 3~5의 코드(img.onload에 리스너를 등록하는 코드)가 라인 2의 img.src="elsa.png"; 의 뒤에서 실행되는 것이 문제이다. 라인 3에서 이미지의 로딩을 통보받는 리스너를 등록하기 전에, 라인 2에 의해 이미지의 로딩이 끝나고 load 이벤트가 발생해버린다면, 이미지를 그리는 라인 4의 코드가 실행되지 않아 이미지가 그려지지 않게 된다. 문제를 수정하여 재작성하면 다음과 같다.

```
var img = new Image();
img.onload = function f() {
    context.drawImage(img, 10, 10);
}
img.src = "elsa.png";
```

연습문제 12

EXERCISE

이론문제

1. ① 사용자 컴퓨터의 고성능화 때문

2. ③ 웹하드

3. ① 브라우저가 웹 서버로부터 이미지나 웹 페이지 등 하나의 웹 자원을 가져 오는 과정

4. HTML 파일, mystyle.css, 1.png, 2.png, 3.png을 각각 읽어오기 위해 한 번씩 요청을 보내기 때문에 브라우저는 웹 서버에 총 5번 요청을 보낸다.

5. ③ 200

6. ① 사용자 로컬 컴퓨터의 파일

7. ③ 사용자가 주로 보는 동영상 파일.
설명) 이유는 하나의 쿠키는 4KB이하의 작은 정보만 저장하는데 동영상 파일은 보통 이보다 훨씬 크기 때문이다.

8. ① 한 사이트의 여러 웹 페이지 사이의 정보 공유를 위해

9. ③ 다양한 형태의 정보 저장

10.
(1) 사용자의 입력 패턴을 분석하여 저장할 때 - 로컬 스토리지.
브라우저 윈도우를 종료하거나 심지어 브라우저를 종료해도 정보가 계속 남아 있어야 오랜 동안 누적된 사용자 입력 패턴을 분석할 수 있기 때문. 또한 여러 윈도우에 걸쳐 사용자의 입력 패턴이 공유되어야 되기 때문
(2) 게임에서 현재 사용자의 이름과 점수를 저장할 때 - 세션 스토리지.
윈도우가 다르면 서로 다른 사용자로 다루어져야 되기 때문.
(3) 게임에서 최고 점수 10명의 이름과 점수를 저장할 때 - 로컬 스토리지.
윈도우에 관계없이 모든 윈도우에서 공유되어야 하는 정보이므로

11. ② Storage 이벤트

연습문제 13

EXERCISE

이론문제

1.

(1) 예

(2) happy.mp3는 1회 재생된다.

(3) 출력되지 않는다.

(4) 주석문과 일치하도록 빈 칸에 자바스크립트 코드를 채워라.

```
var song = document.getElementById("pop");  // audio DOM 객체 알아내기
song.pause();                               // 오디오 일시 중지
song.volume -= 0.1;                         // 음량 0.1 만큼 줄이기
song.src = "media/birthday.mp3";            // birthday.mp3로 오디오 바꾸기
song.play();                                // 바꾼 오디오 재생 시작
song.muted = true;                          // 음 소거하여 들리지 않게 하기
```

3. ② onended

4.

```
var song = document.getElementById("pop");        // audio DOM 객체 알아내기
song.onended = function (e) {                     // 현재 음악 재생이 끝나면
    song.src = "media/birthday.mp3";              // birthday.mp3로 오디오 바꾸기
    song.play();
    song.onended = null;                          // 이벤트 리스너 제거
}
```

5. ① 모바일 장치에서는 작동하지 않는다.

6. HTML5에서는 (geolocation) 를 통해 위치 정보 서비스를 제공하며, 현재 위치를 알아내기 위해 (getCurrentPosition()) 메소드를 호출하고, 이동 중인 경우 위치가 바뀔 때마다 위치를 통보받고자 하면 (watchPosition()) 메소드를 호출한다.

7. ① 미분이나 적분 등 계산 중심적인 자바스크립트 코드를 실행할 때

8. ② onmessage

9. ① 워커 객체

10.

(1)

```
var w = new Worker("increment.js");        // 워커 태스크를 만든다.
w.postMessage("100");                      // 워커 태스크에게 숫자 100을 보낸다.
w.terminate();                             // 워커 태스크를 강제로 종료시킨다.
```

(2)

```
onmessage = function (e) {
    var n = parseInt(e.data);
    n++;
    postMessage(n);
}
```

(3) 워커 태스크로부터 오는 계산 결과를 받아 alert()로 출력하는 자바스크립트 코드를 작성하라. w는 워커 객체이다.

```
w.onmessage = function (e) {
    var m = e.data;
    alert(m);
}
```