|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2021\_1\_WEB\_11\_과제 | 학번 : | 20165164 | 이름 : | 지현한 |

* 개념 확인

1. 빈 괄호를 채워 넣으시오
2. 태그를 자바스크립트에서 사용할 수 있는 객체로 만든 것을 ( DOM객체 )라고 한다.
3. ( 인라인 이벤트 핸들러 방식 )은 HTML 태그 내부에 자바스크립트 코드를 넣어 이벤트를 연결하는 방식이다
4. 이벤트 모델로 연결된 함수를 ( 이벤트 핸들러 ) 또는 ( 이벤트 리스너 )라고 한다.
5. ( 디폴트 행동 )는 특정 태그가 가진 기본적인 이벤트를 의미한다.
6. ( document 객체 )는 웹 브라우저에서 보여주는 문서와 관련된 정보를 접근할 때 사용하는 객체 이다
7. ( className ) 프로퍼티는 클래스 속성값을 가져오거나 설정하며 ( id ) 프로퍼티는 id 속성의 값을 가져오거나 설정한다
8. ( textContent ) 프로퍼티는 마크업이 포함되면 문자열로 인식하며 ( innerHTML ) 프로퍼티는 마크업이 포함된 새로운 콘텐츠를 DOM에 추가할 수 있다
9. 다음 질문에 OX를 표시하시오.
10. 웹 페이지에 적혀 있는 HTML 태그들은 위쪽에서 아래쪽으로 차례대로 실행한다.( X )
11. script 태그는 반드시 head 태그 안에 위치해야 한다. ( X )
12. querySelector() 메서드는 선택자로 요소를 여러 개 선택할 가능성이 있으므로, 배열로 문서 객체를 리턴 한다. ( O )
13. 이 벤트 리스너 내부에서 this 키워드를 사용하면 이벤트를 발생한 자기 자신을 의미한다 ( O )
14. 이벤트 객체를 사용하면 이벤트와 관련한 정보를 알아낼 수 있다. ( O )
15. a 태그를 클릭했을 때 발생하는 웹 페이지 이동 등 기본 이벤트는 제거할 수 없다. ( X )
16. 다음 태그 중에서 텍스트 노드를 갖지 않는 태그는? 2
17. h1 태그
18. title 태그
19. br 태그
20. a 태그
21. 다음 중 문서 객체를 하나만 리턴하는 함수는? 1
22. document.getElementById()
23. document.getElementsByName()
24. document.getElementsByClassName()
25. document.querySelectorAll()
26. 다음 어떤 문서 객체의 background-image 스타일 속성을 선택하는 코드 중 오류가 발생하는 코드는? 3
27. tag.style[‘background-image’]
28. tag.style[‘backgroundImage’]
29. tag.style.background-image
30. tag.style.backgroundImage
31. 다음 코드의 문제와 해결 방법을 제시하시오.

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Document Object Model</title>  <script>  let element = document.getElementById('element');  element.style.color = 'red';  element.innerHTML = 'From JavaScript';  </script>  </head>  <body>  <h1 id="element">Lorem ipsum dolor amet</h1>  </body>  </html> |

태그 아래에 <script>삽입

1. 다음 코드에서 h1 태그를 선택할 수 있는 자바스크립트 코드를 작성하시오.

|  |
| --- |
| <!DOCTYPE html>  <html>  <body>  <h1 id="element" class="header">Lorem ipsum dolor amet</h1>  </body> |

let element = document.getElementById(“element”);

1. 다음 코드의 실행 결과를 설명하시오.

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Query Selector</title>  </head>  <body>  <h1>Header - 1</h1>  <h1>Header - 2</h1>  <h1>Header - 3</h1>  <script>  let header = document.querySelector('h1');  header.style.color = 'white';  header.style.backgroundColor = 'black';  </script>  </body>  </html> |

첫번째 h1태그에 글자색을 흰색으로 변경, 배경을 검은색으로 변경

1. 질문에 해당하는 스크립트를 작성하시오
2. p 태그 노드를 생성하여 nodeP에 저장한다

const newnode = document.createElement(‘p’);

1. ①에서 생성된 노드에 "텍스트 추가" 텍스트 노드를 생성하여 nodeT에 저장한다

const nodeT = newnode.createTextNode(‘텍스트 추가’);

1. ①에서 생성된 노드에 ②에서 생성된 텍스트 노드를 자식 노드로 추가한다

newnode.appendChild(nodeT);

1. ①에서 생성된 노드에 자식 노드가 있는지를 확인한다

newnode.hsaChildNodes();

1. 이벤트 용어를 설명하시오

이벤트 이름: 이벤트가 발생하는 이벤트 타입

이벤트 속성: 이벤트를 연결할 때 활용하는 속성, 이름 앞에 on을 붙여 사용

이벤트 리스너(이벤트 핸들러): 이벤트 속성에 할당된 함수

1. DOM 트리 노드 종류에 대하여 설명하시오

문서노드: 트리의 최상위에 존재, DOM tree에 접근하기 위한 시작점(entry point)

요소 노드: HTML 요소 표현, 어트리뷰트, 텍스트 노드에 접근하려면 먼저 요소 노드를 찾아 접근

어트리뷰트 노드: HTML 요소의 어트리뷰트, 해당 요소 노드를 찾아 접근하면 참조, 수정 가능

텍스트 노드: HTML 요소의 텍스트표현, 요소 노드의 자식이며 자식 노드를 가질 수 없다

1. 이벤트 흐름에 대하여 설명하시오

캡쳐 단계: 이벤트가 window 객체에서 중간의 모든 DOM 객체를 거쳐 타겟 객체에 전달되는 과정, 이벤트가 거쳐가는 모든 DOM 객체(window포함)의 이벤트 리스너 실행

버블 단계: 이벤트가 타겟에서 중간의 모든 DOM 객체를 거쳐 window 객체에 전달되는 과정, 이벤트가 거쳐가는 모든 DOM 객체(window포함)의 이벤트 리스너 실행

* 개념 활용

1. 실행 결과를 제시하시오.



|  |
| --- |
| [실행결과] |

1. ‘show’버튼을 클릭하면 type 속성값이 text <-> password로 토글 되도록 한다.

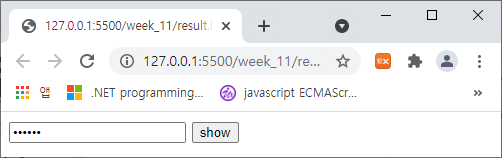
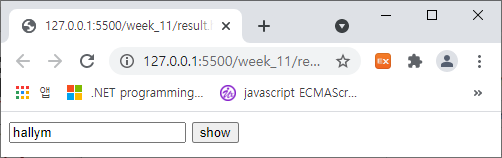
힌트) 1번과 클로저 함수 참조

<body>

   <input class="password" type="password" value="hallym">

   <input type="button" class="button" value="show" onclick="makeClickHandler()">

</body>

|  |
| --- |
| [소스]  <!DOCTYPE html>  <html>  <head>    </head>  <body>      <input class="password" type="password" value="hallym">      <input type="button" class="button" value="show" onclick="makeClickHandler()">      <script>           const pass = document.querySelector(".password");           const button = document.querySelector(".button");           function makeClickHandler(){               let isShow = false;               return function(){                   pass.setAttribute('type',isShow ? 'password' : 'text');                   isShow = !isShow;                   button.innerHTML = isShow ? 'hide' : 'show';               };           }           button.onclick = makeClickHandler();      </script>  </body>  </html> |
| [실행결과] |

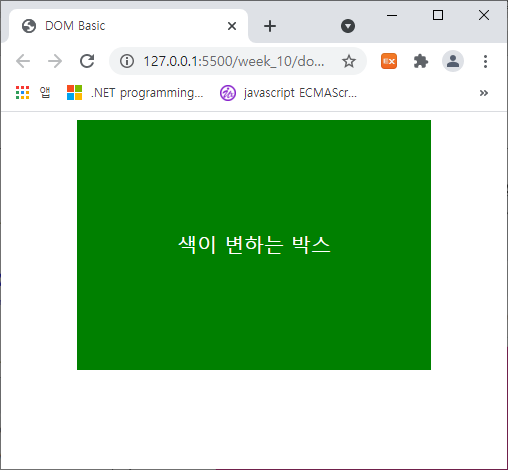
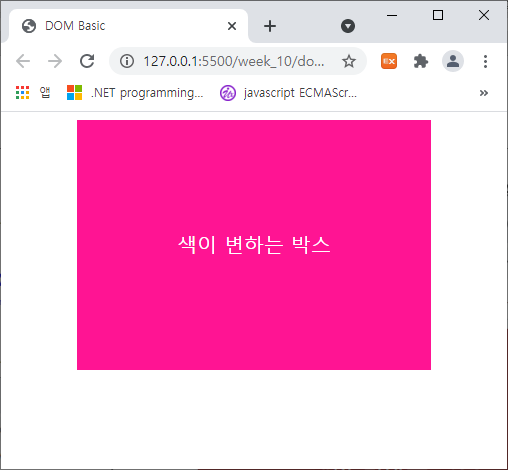
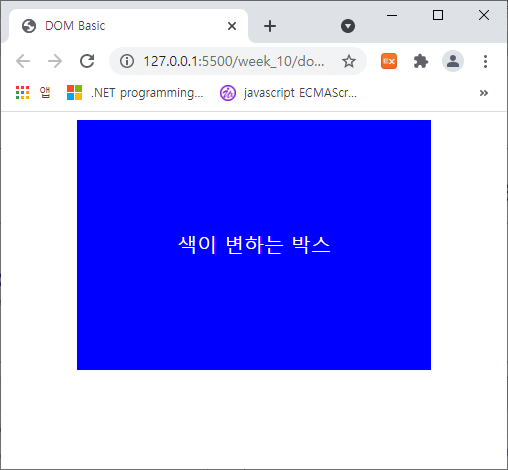
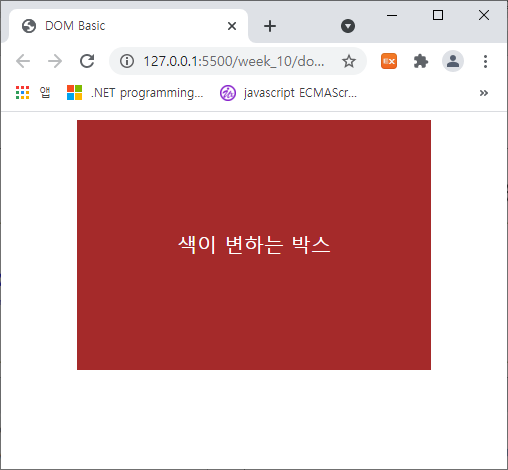
1. 1초 마다 배경색이 변하는 박스를 출력하는 웹 문서를 작성하시오. 단 박스(div) 관련 속성은 스크립트에서 지정한다. 색상은 7개이상.

힌트) setInterval(function(){  console.log('first()'); }, 1000); //1초 간격으로 첫번째 인수 실행

<body>

   <div>색이 변하는 박스</div>

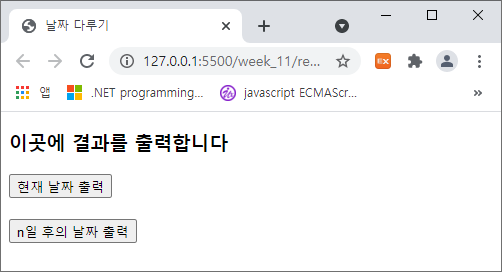
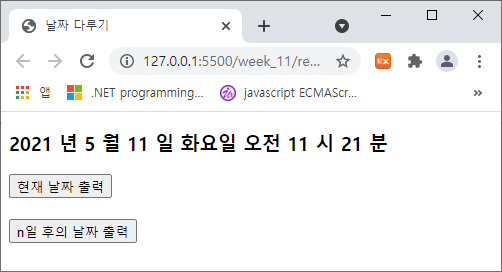
</body>

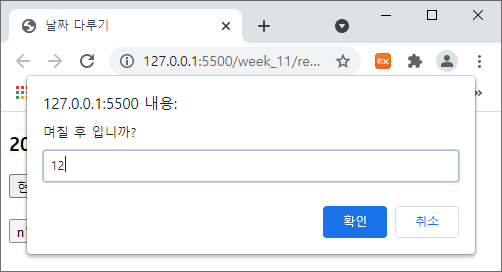
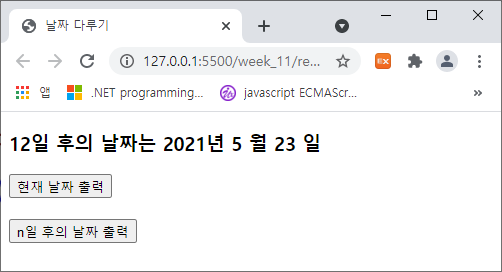


|  |
| --- |
| [소스]  <!DOCTYPE html>  <html>  <head>      <script>          window.onload = function(){              let div = document.querySelector('div');              div.style.height = '70vh';              div.style.width = '70vw';              div.style.margin = '0 auto';              div.style.fontSize = '20px';              div.style.color = 'white';              div.style.textAlign = 'center';              div.style.lineHeight = '70vh';              let color = new Array('blue','red','brown','green','black','deeppink','orange');              let index=0;              setInterval(function(){                  div.style.backgroundColor=color[index++ % color.length];          },1000);      };      </script>  </head>  <body>      <div>색이 변하는 박스</div>  </body>  </html> |
| [실행결과] |

1. 첫번째 버튼을 클릭하면 현재 날짜를, 두번째 버튼을 클릭하면 입력 받은 값만큼 지난 후의 날짜를 h3 태그 텍스트로 출력한다.

* 현재 날짜 출력을 위한 toDay() 메소드 구현
* n일 후의 날짜 출력을 위한 fuDay()메소드 구현 – 입력창으로 n일 후의 값을 입력받는다

<body>

  <h3>이곳에 결과를 출력합니다</h3>

  <input type="button" value="현재 날짜 출력" onclick="toDay()" ><br><br>

  <input type="button" value="n일 후의 날짜 출력" onclick="fuDay()" >

</body>

|  |
| --- |
| [소스]  <!DOCTYPE html>  <html>      <head>      </head>      <body>          <h2>이곳에 결과를 출력합니다</h2>          <input type="button" value="현재 날짜 출력" onclick="toDay()"><br><br>          <input type="button" value="n일 후의 날짜 출력" onclick="fuDay()">            <script>              const head = document.querySelector('h2');              let toDay = () => {                  let date = new Date();                  let day = new Array('일', '월', '화', '수', '목', '금', '토');                  let result = new String();                  result += date.getFullYear() + '년 ';                  result += (date.getMonth() + 1) + '월 ';                  result += date.getDate() + '일 ';                  result += day[date.getDay()] + '요일 ';                  if(date.getHours() < 12 ) {                      result += '오전 ' + date.getHours() + '시 ';                  } else {                      result += '오후 ' + (date.getHours() -12) + '시 ';                  }                  result += date.getMinutes() + '분 ';                    head.innerHTML = result;              }              let fuDay = ()=>{                  let date = new Date();                  let inputnumber = prompt(`며칠 후 입니까?`);                  let temp = new Date(date.setDate(date.getDate() + parseInt(inputnumber)) );                    let result = new String();                  result += inputnumber + `일 후의 날짜는 `;                  result += temp.getFullYear() + '년 ';                  result += (temp.getMonth()+ 1) + '월 ';                  result += temp.getDate() + '일';                  head.innerHTML = result;              }          </script>      </body>  </html> |
| [실행결과] |

1. 텍스트를 클릭하면 입력창에서 데이터를 입력한다. f를 입력하면 전체 데이터를 제시된 결과처럼 출력한다

<head>

  <meta charset="UTF-8">

  <title>여행지</title>

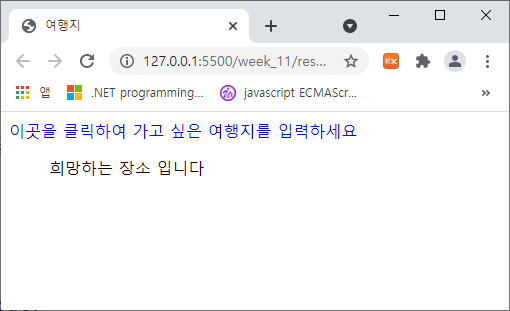
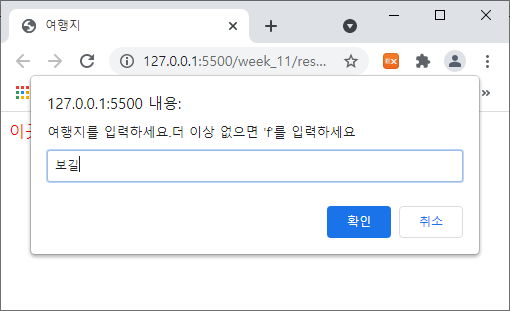
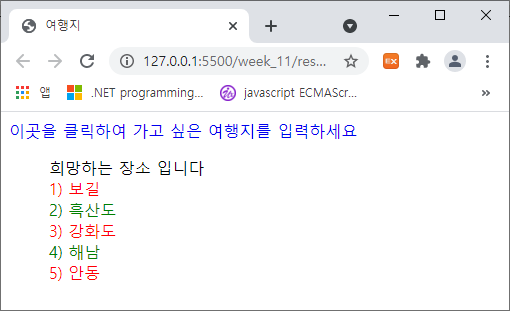
</head>

<body>

  <a href="javascript:insert()">이곳을 클릭하여 가고 싶은 여행지를 입력하세요</a>

  <ul>희망하는 장소 입니다</ul>

</body>

|  |
| --- |
| [소스] |
| [실행결과] |