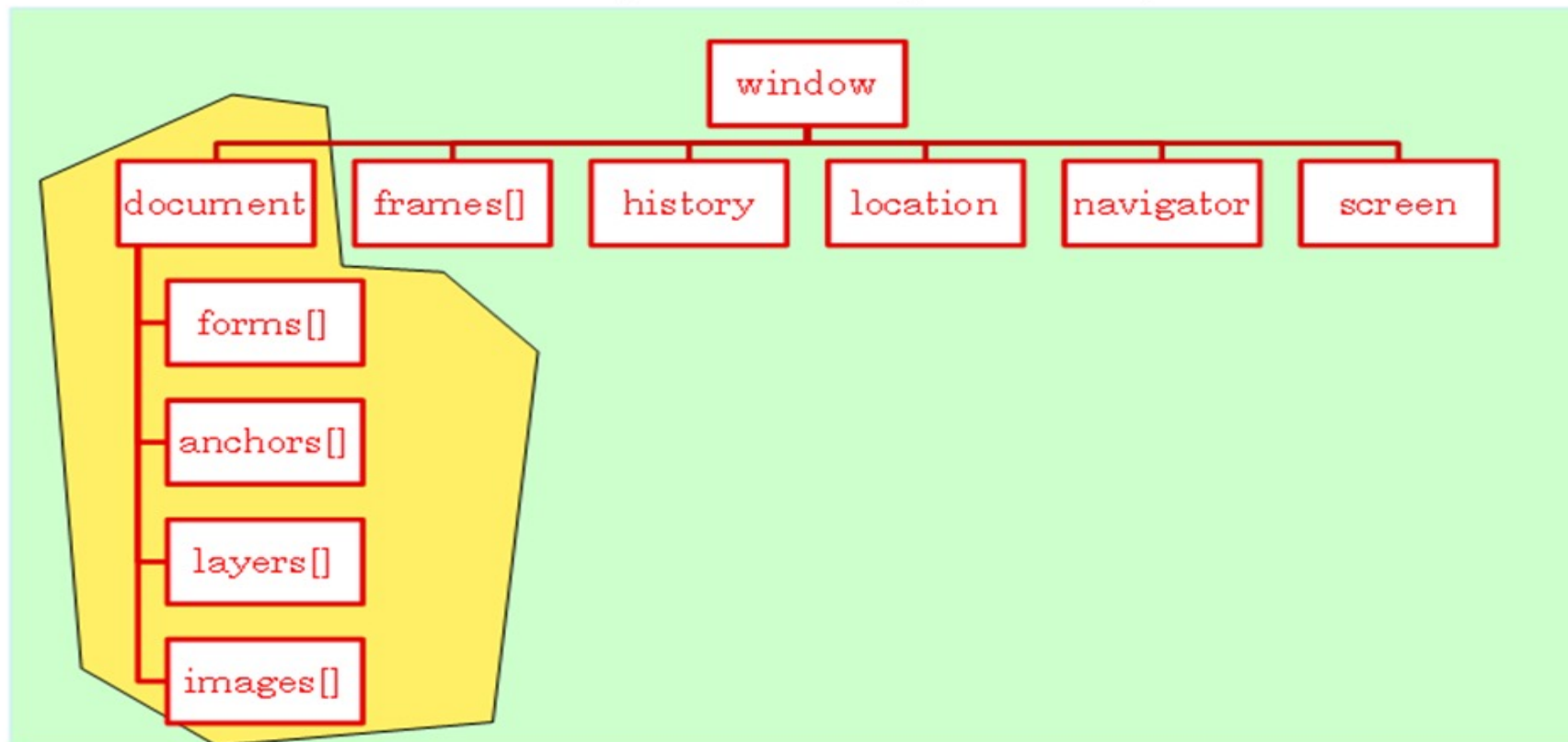


10 BOM과 DOM

DOM/BOM

- HTML 문서를 객체로 표현한 것을 DOM
- 웹 브라우저를 객체로 표현한 것을 BOM(Browser Object Model)

BOM(Browser Object Model)



DOM(Document Object Model)

Window 객체

메소드	설명
open()	새로운 창을 연다
close()	열려진 창을 닫는다
alert()	내용을 나타내는 경고 창이 뜬다
confirm()	사용자의 대답을 확인하는 창이 뜬다
prompt()	메시지와 초기값을 나타내고 새로운 값을 입력할 수 있는 창이 뜬다.
setTimeout(fn, millisecond)	주어진 시간이 경과하면 지정된 함수가 호출되어 실행된다
setInterval(fn, millisecond)	주어진 시간이 경과할 때마다 지정된 함수가 호출되어 실행된다
clearInterval(id 변수)	setInterval() 메소드를 종료시킨다

Window 객체

속성	설명
opener	open()을 통해 새로운 창을 열었을 때 그 창을 자식창이라 한다면 자식창에서 부모창을 가리킬 때 opener라 한다

새로운 윈도우 오픈 예제

```
window.open(URL, name, specs);
```

- URL : 오픈할 페이지의 URL
- name : 타겟(target)을 지정하거나 윈도우의 이름
- specs : 여러가지 속성

새로운 윈도우 오픈 예제

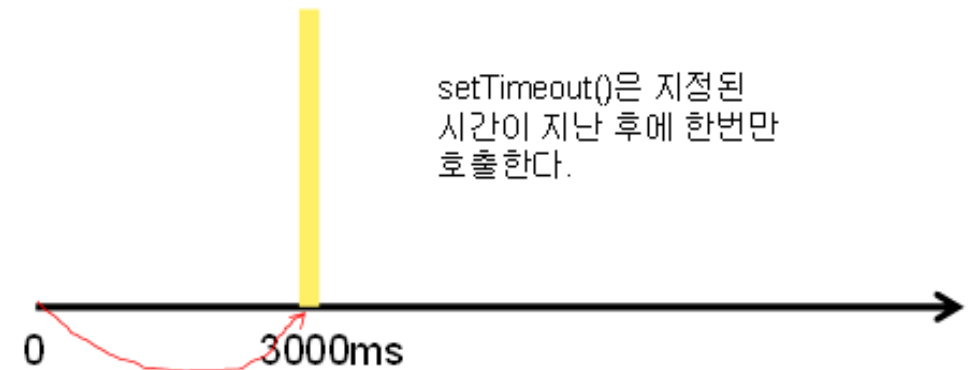
```
<!DOCTYPE html>
<html>
<head>
<script>
function openPopup() {
    open("http://www.google.co.kr", "_blank", "width=200,height=200");
}
</script>
</head>
<body>
<input type="button" value="구글창 열기" onclick="openPopup();" >
</body>
</html>
```

setTimeout()

setTimeout(code, millisec)

호출되는 함수의 이름, 호출되는 함수를 여기서 직접 정의할 수도 있다.

함수를 호출하기 전에 흘러야 하는 시간



예제

```
<script>  
function showAlert() {  
    setTimeout(function () { alert("Hello~~") }, 1000);  
}  
</script>  
  
<body>  
    <button onclick="showAlert()">눌러보세요</button>  
</body>
```



```
function sayHi(who, phrase) {  
    alert( who + ' 님, ' + phrase );  
}  
setTimeout(sayHi(), 1000, "홍길동", "안녕하세요.");
```

setTimeout은 함수의 참조 값을 받도록 정의되어 있는데 sayHi()를 인수로 전달하면 *함수 실행 결과*가 전달되어 버립니다. 그런데 sayHi()엔 반환문이 없습니다. 호출 결과는 undefined가 되겠죠. 따라서 setTimeout은 스케줄링할 대상을 찾지 못해, 원하는 대로 코드가 동작하지 않습니다.

setInterval()

setInterval()은
주기적으로
호출한다.



문제

- test_setInterval, 1초마다 브라우저 배경색이 랜덤으로 바뀌는 프로그램을 작성하시오. --
- 참고 : HTML에서 색 지정 방법
 - 색 이름을 영문으로 지정(black, pink, orange, yellow, blue, red...)
 - #RRGGBB : 16진수로 표현
 - 10진수를 16진수로 표현하는 방법
 - `var a = 230; a.toString(16);`
 - 만약에 파랑을 만들고 싶으면
`var red = 0, green = 0, blue = 255;`
`"#" + red.toString(16) + green.toString(16) + blue.toString(16);`

location 객체

속성	설명
hash	URL 중에서 앵커 부분을 반환한다.
host	URL 중에서 hostname과 port를 반환
hostname	URL 중에서 hostname을 반환
href	전체 URL을 반환
pathname	URL 중에서 경로(path)를 반환
port	URL 중에서 port를 반환
protocol	URL 중에서 protocol 부분을 반환
search	URL 중에서 쿼리(query) 부분을 반환

메서드	설명
assign()	새로운 문서를 로드한다. 히스토리를 남김
reload()	현재 문서를 다시 로드한다.
replace()	현재 문서를 새로운 문서로 대체한다. 히스토리 남기지 않음. 뒤로 가기 안됨

Location 객체

Location 객체

- URL 정보를 얻어오는 객체
- href : 전체 URL
- port : port
- host : hostname과 port
- hostname : hostname
- pathname : path(경로)

Location

Location 객체

location.href

- 자동으로 특정 페이지로 이동할 때 사용 **페이지 이동**
- a 태그 이용 [공지사항목록href=javascript:goPage\(\)](#) [공지사항목록onclick="goPage\(\)](#) [공지사항목록 onclick=location.href](#)
- get 방식에서 **공지사항 목록**

location.replace(URL)

- 새로운 문서로 대체: **공지사항 목록replace**

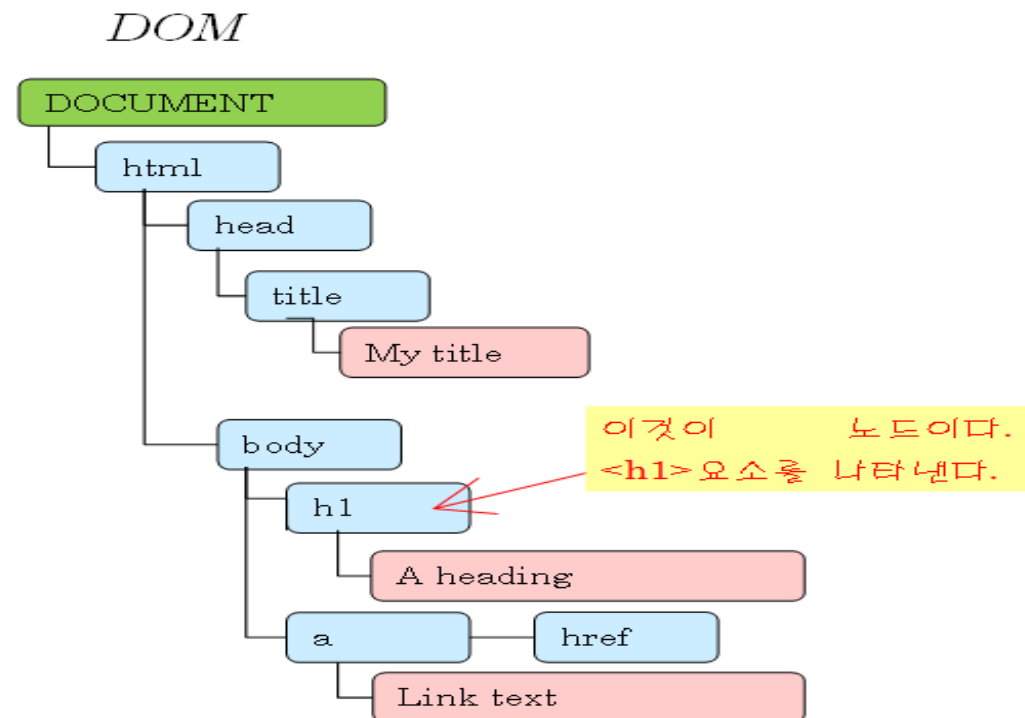
location.reload()

- 현재 문서를 다시 로드: **페이지 재호출**

문서 객체 모델(DOM)

- DOM은 W3C (World Wide Web Consortium)의 표준입니다.
- DOM은 문서를 액세스하기위한 표준을 정의한다 :
- *"W3C 문서 객체 모델 (DOM)은 프로그램 및 스크립트를 동적으로 액세스하여 문서의 콘텐츠, 구조 및 스타일을 갱신 할 수 있도록 플랫폼 및 언어 중립 인터페이스 이다."*
- The HTML DOM is a standard for how to get, change, add, or delete HTML elements.

```
<html>
  <head>
    <title>My title</title>
  </head>
  <body>
    <h1>A heading</h1>
    <a href="#" >Link text</a>
  </body>
</html>
```



HTML 요소 찾기

- 동적인 웹페이지를 작성하려면 원하는 요소를 찾아야 한다.

```
document.getElementById("result1");  
document.getElementsByName("comImg")[0]  
document.getElementsByTagName("input")[0];  
document.getElementsByClassName('out')[0]
```

```
document.querySelector('input')  
document.querySelector('#result1')  
document.querySelector('.out')  
  
document.querySelectorAll('input')  
document.querySelectorAll('.out')
```

DOM 트리 순회

속성	내용
childNodes	한 요소의 모든 자식 요소에 접근할 수 있다. 배열이 반환
firstChild	"childNodes" 배열의 첫번째 자식 노드가 반환된다. "childNodes[0]"와 같다.
lastChild	"childNodes" 배열의 마지막 자식 노드가 반환된다. "childNodes[childNodes.length - 1]"와 같다.
parentNode	현재 노드의 부모 노드를 반환한다.
nextSibling	현재 노드의 다음 형제 노드를 반환한다.
previousSibling	현재 노드의 이전 형제 노드를 반환한다.

DOM - firstChild

```
<!DOCTYPE html>
<html>
<body>
  <ul>
    <li>List item 1</li>
    <li>List item 2</li>
    <li>List item 3</li>
    <li>List item 4</li>
    <li>List item 5</li>
  </ul>
  <script>
    var list = document.getElementsByTagName('ul')[0];
    var allItems = list.getElementsByTagName('li');

    for (var i = 0, length = allItems.length; i < length; i++) {
      alert(allItems[i].firstChild.data);
    }
  </script>
</body>
</html>
```

새로운 요소 생성

- 텍스트 노드를 갖는 요소와 갖지 않는 요소로 구분
- 요소노드와 텍스트 노드를 생성한 후에 텍스트 노드를 요소 노드에 붙임

메소드	설명
<code>createElement(tagName)</code>	태그요소 생성
<code>createTextNode(text)</code>	텍스트 노드 생성
<code>appendChild(node)</code>	새로운 노드를 추가
<code>removeChild(node)</code>	노드를 삭제한다
<code>remove()</code>	노드삭제

새로운 HTML 요소 생성

랜덤으로 문자열 발생하여 div요소에 문자열 추가하기
문자열은 배열에 저장

```
<script>
```

```
function proc1() {  
    text = "문자열";
```

```
    var node = document.createTextNode(text);
```

```
    document.getElementById("target").appendChild(node);
```

```
}
```

```
</script>
```

```
<input type="button" value="확인" onclick="proc1()">
```

```
<div id="target"> </div>
```

새로운 HTML 요소 생성

- 새로운 이미지 태그 만들어 추가하기
- -createElement()
- - firstChild.data
- -appendChild()

1. 새로운 HTML 요소 생성(이미지 요소)

버튼을 클릭하면 이미지를 표시한다.

- 1. 각 list 의 문자를 가져온다 - firstChild.data;
- 2. 이미지 객체를 생성한다 - createElement('img')
- 3. 각 list의 자식으로 이미지를 추가한다 - appendChild()

실행

- 컴퓨터1
- 식품2
- 호랑이
- cloud03
- coffee1

새로운 HTML요소생성

- 실행결과
- li요소들을 가져온다
- li 요소만큼 반복문 -
 - li의 data 값을 가져온다
- img엘리먼트 생성 해서
변수에 대입
- img의 src, width, height
속성을 설정
- li요소에 자식요소로 img를 추가



예제

- 각 리스트 요소를 클릭하면 해당 이미지 표시

2. 새로운 HTML 요소 생성(이미지 요소)

각 리스트를 클릭하면 해당 이미지를 표시한다.

- 1. 메소드 호출시 매개변수로 this를 넘겨준다
- 2. createElement() : 이미지 요소를 생성해라
- 3. appendChild() : this 요소 안에 자식으로 넣어라

- 컴퓨터1
- 식품2
- 호랑이
- cloud03



- coffee1

HTML 요소 삭제

- **removeChild("node");**

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function removeNode() {
        var parent = document.getElementById("target");
        var child = document.getElementById("p1");
        parent.removeChild(child);
      }
    </script>
  </head>
  <body>
    <div id="target">
      <p id="p1">첫번째 단락</p>
      <p id="p2">두번째 단락</p>
    </div>
    <button onclick="removeNode()">누르세요!</button>
  </body>
</html>
```

추가/삭제(1)

추가/삭제

- createElement('p')
- createTextNode()
- appendChild()

추가

랜덤으로 발생하는 문자열로 textNode 생성
div에 textNode 추가

삭제

lastChild
removeChild(child)

좋은아침

행복한날

Hello

좋은아침

Happy

안녕!

추가/삭제(2)

추가삭제

parentNode
remove()

Good Morning

삭제

안녕!

삭제

행복한날

삭제

좋은아침

삭제

행복한날

삭제

Good Morning

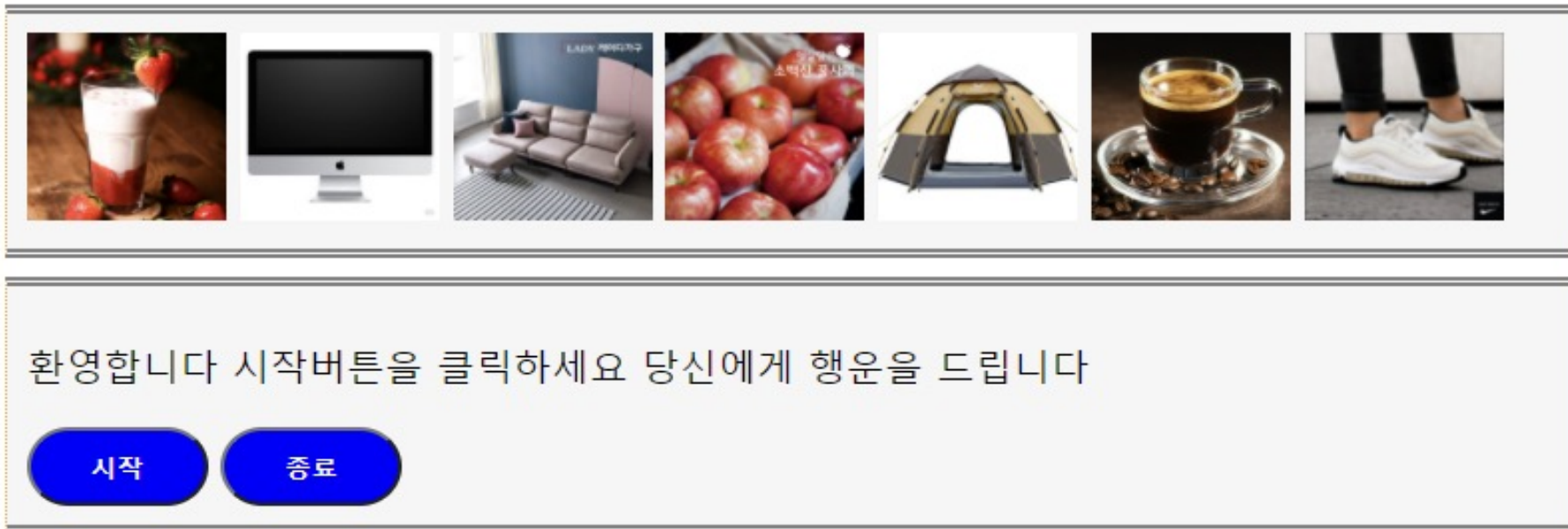
삭제

Hello

삭제

연습

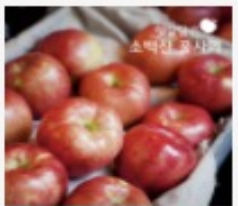
- 시작버튼 클릭하면 첫 번째 이미지(index: 0)를 div box의 뒤로 추가한다
- 첫 번째 이미지가 div box의 뒤로 추가될때 첫 번째 이미지는 삭제되어진다
- 1초마다 반복적으로 실행한다
- 시작버튼은 클릭하면 감춴다가 종료 버튼 클릭 시 다시 표시된다





환영합니다 시작버튼을 클릭하세요 당신에게 행운을 드립니다

종료



환영합니다 시작버튼을 클릭하세요 당신에게 행운을 드립니다

시작

종료