pushstate

Introduction

pushstate

pushstae란?

HTML5 에서 새로 추가된 history객체의 method로 session history 에 주어진 타이틀로, 주어진 데이터를 쓰며 url이 주어진 경우 적용합니다.

사양참조 : http://www.w3.org/TR/html5/browsers.html#dom-history-pushstate

왜 pushstae를 사용하는가?

먼저 전통적인 웹 개발방식에 의해 만들어진 웹페이지를 생각해보겠습니다.

A 화면에서 B 화면으로 이동하기 위해 우리는 anchor tag나 form submit 등을 이용하여 페이지를 이동합니다.

또한 현재 화면의 데이터를 저장하려면 hidden tag 나, 세션, 쿠키등을 사용하여야 하고

일반적으로 사용자가 데이터에 대한 처리를 하고자 할 때 페이지를 호출하는 것 (URL을 호출하는 것) 이외에 다른 방법은 없습니다.

그렇다면 url에 데이터를 저장하려면 어떻게 해야 할까요?

가장 간단한 방법은 queryString (질의문자열) 과 #를 사용하는 것입니다.

ex) http://www.slipp.net/wiki/pages/editpage.action?pageId=19530134&userid=gogo

http://www.slipp.net/wiki/pages/writepage#pushstateinfo

관련 객체

window.location

window.location.search

#를 쓰면 데이터를 편하게 저장할 수 있습니다.

그런데 #를 사용했을 때의 문제는 #뒤의 내용을 변경하면 히스토리도 같이 변경됩니다.

따라서 뒤로가기버튼 클릭과 같은 상황에 대해서 일반적으로 대응할 수가 없습니다.

물론 타이머를 이용해서 주기적으로 #값을 체크해서 변경되면 해당페이지를 복원하는 방법도 있지만 예외처리에 대한 비용이 지나치게 많이듭니다.

그래서 HTML5에 추가된 history.pushstate, history.replaceState를 이용해서 보다 편리하게 history를 관리하게 되었고

이를 통해서 사용자가 의도하지 않은 행동 back 버튼등을 클릭하거나 다시 앞으로 가기 버튼을 눌렀을 때

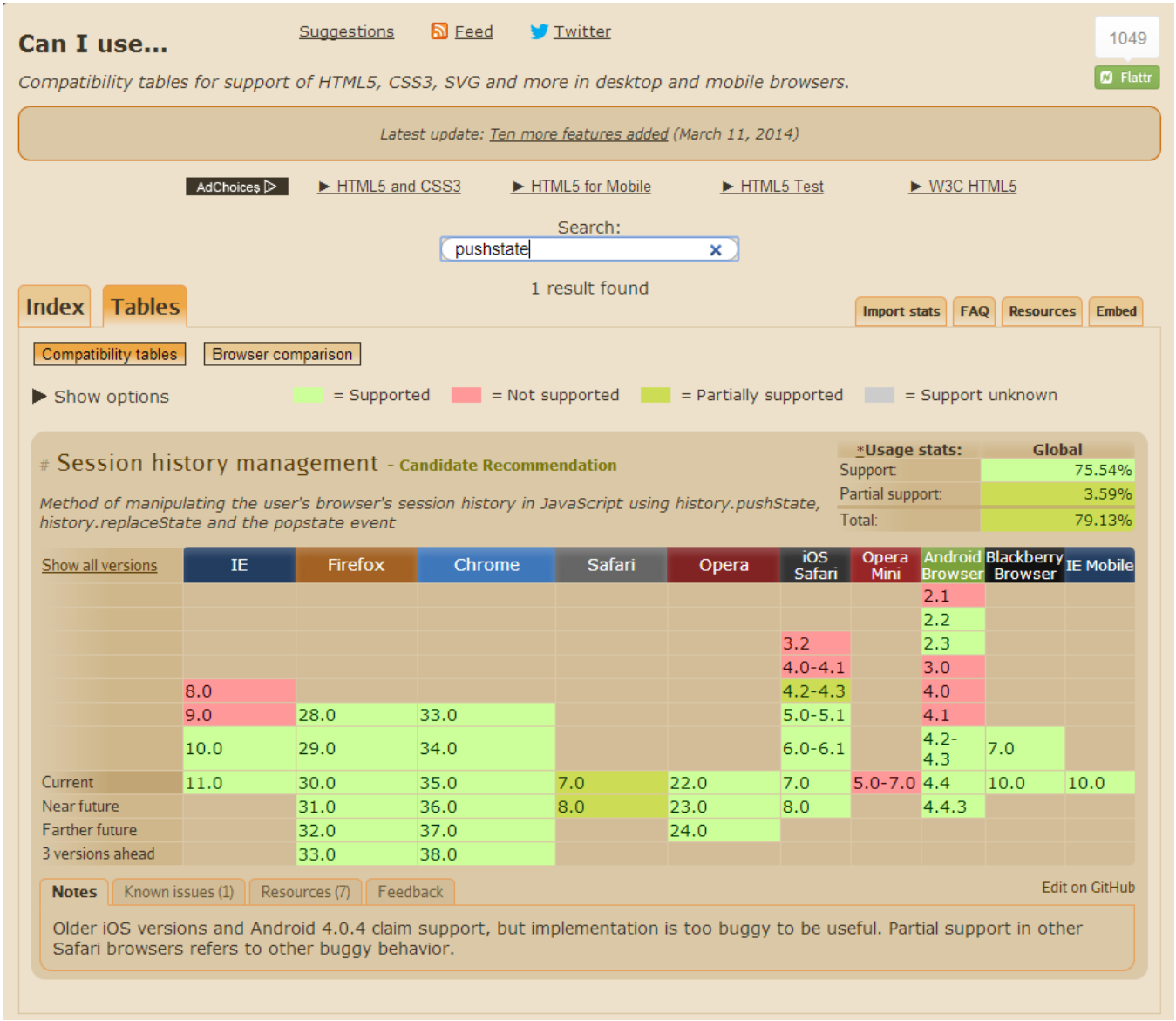
(url해시 또는 이벤트를 가로채서 처리할 수도 있겠지만..)
pushstate를 사용하면 상태가 변경되었을 때 URL자체를 업데이트해서 url에 상태를 반영할 수 있습니다.

pushstate API

- window.history.pushState(data, title [, url])

- window.history.replaceState(data, title [, url]

pushstate 지원 브라우저



url 참고 : http://caniuse.com/#search=pushstate

간단한 예제로 알아보는 차이점

말로만 설명을 하는 것보다는 간단한 예제를 통해서 기존방식과의 차이와 사용법을 익혀보겠습니다.

전통적인 html 방식의 화면 전환

- 다음과 비슷하게 html 파일을 여러개 작성하고 anchor 태그를 통해서 화면의 전환이 일어난다.

basic_0.html

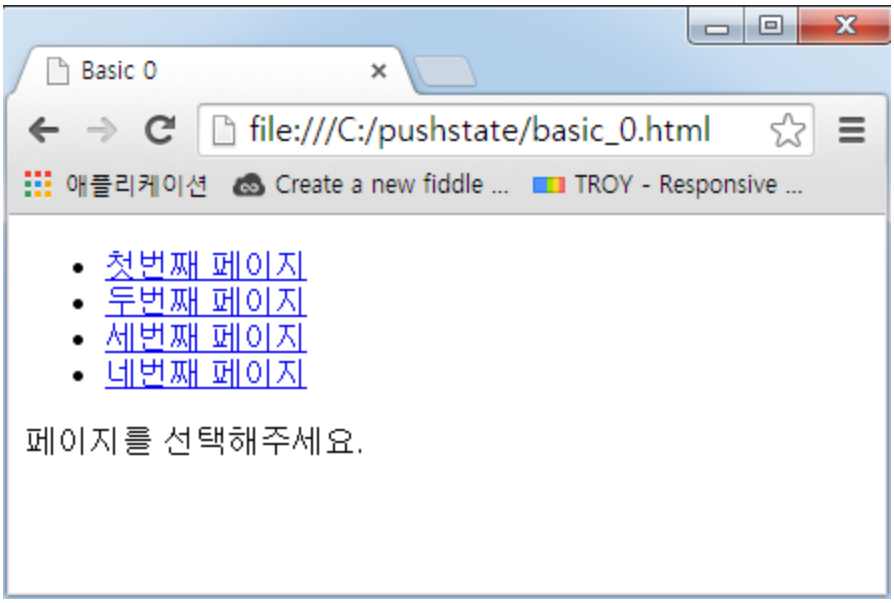
```
<!DOCTYPE html>
<html lang="ko">
<head>
<title>Basic 0</title>
</head>
<body>
<ul>
<li><a href="basic_1.html"> </a></li>
<li><a href="basic_2.html"> </a></li>
<li><a href="basic_3.html"> </a></li>
<li><a href="basic_4.html"> </a></li>
</ul>
<div>
 .
</div>
</body>
</html>
```

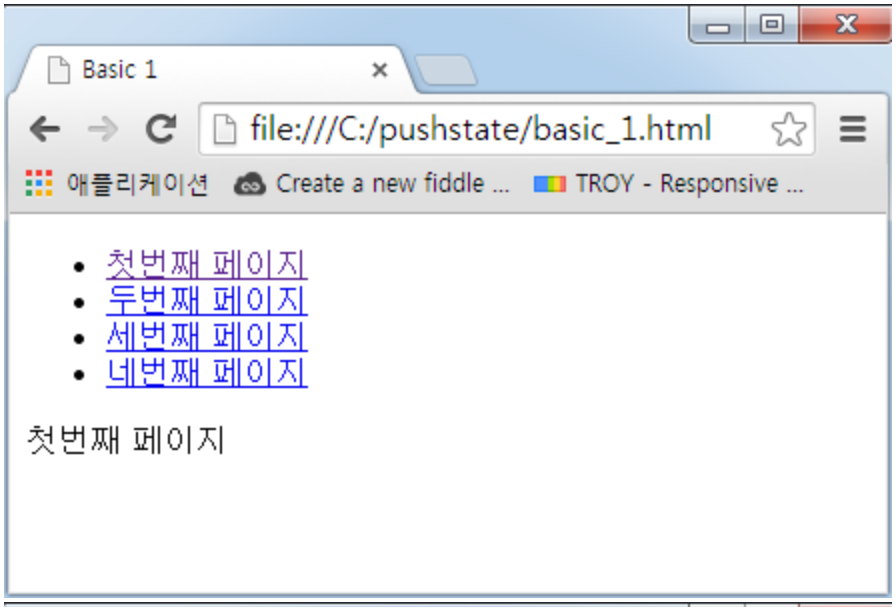
샘플 소스

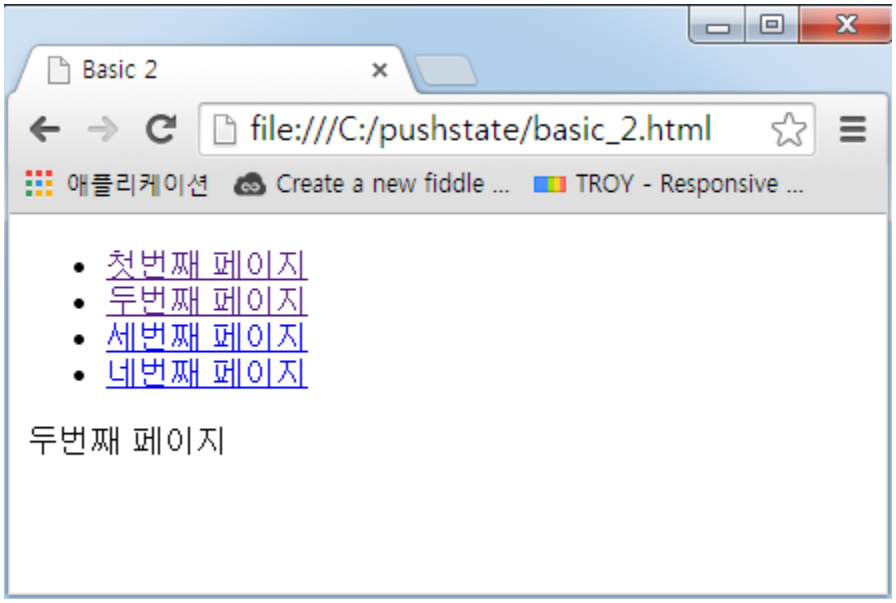
- basic_0.html
- basic_1.html
- basic_2.html
- basic_3.html
- basic_4.html

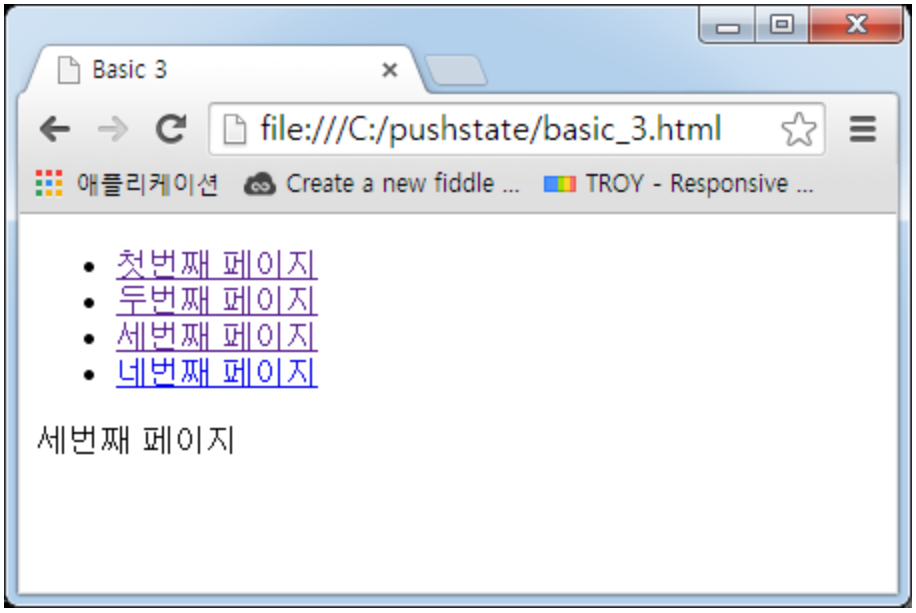
실행결과

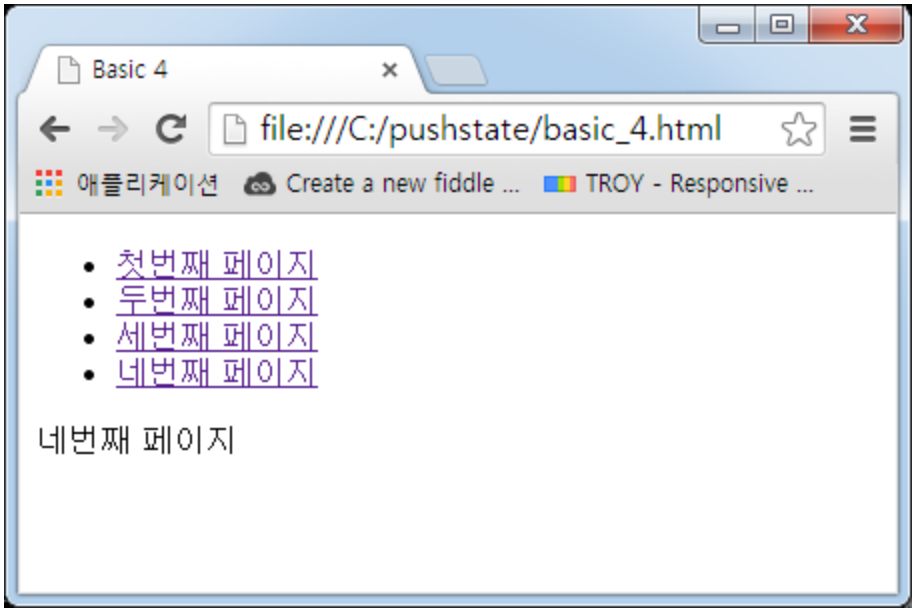
- anchor 태그를 클릭하면 url이 변경되며 화면 이동이 일어난다.
- 백스페이스 버튼등을 클릭하면 history 를 이용하여 전화면으로 돌아간다.




Basic 1
file:///C:/pushstate/basic_1.html
애플리케이션
Create a new fiddle ...
TROY - Responsive ...
첫번째 페이지
두번째 페이지
세번째 페이지
네번째 페이지
첫번째 페이지


Basic 2
file:///C:/pushstate/basic_2.html
애플리케이션
Create a new fiddle ...
TROY - Responsive ...
첫번째 페이지
두번째 페이지
세번째 페이지
네번째 페이지
두번째 페이지


Basic 3
file:///C:/pushstate/basic_3.html
애플리케이션
Create a new fiddle ...
TROY - Responsive ...
첫번째 페이지
두번째 페이지
세번째 페이지
네번째 페이지
세번째 페이지



Single Page 어플리케이션 방식인 경우

- javascript를 이용하여 화면상의 컨텐츠를 변경한다.

single_page.html

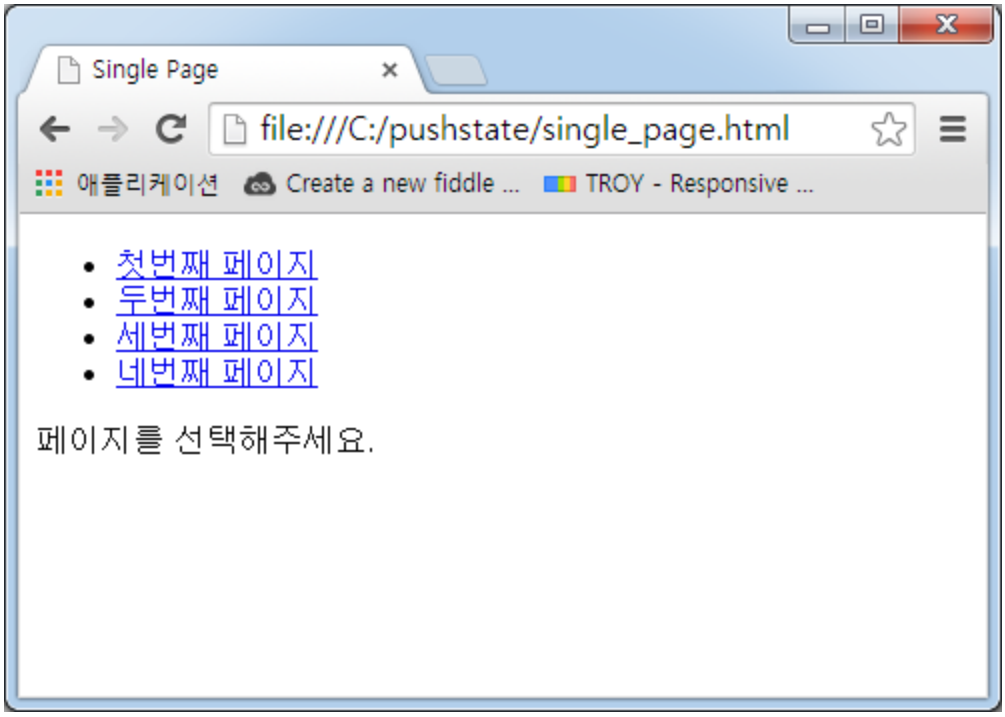
```
<!DOCTYPE html>
<html lang="ko">
<head>
    <title>Single Page</title>
    <script type="text/javascript">
        function viewContents(idx) {

            event.preventDefault();

            var con = document.getElementById("contents");

            switch (idx) {
                case 1:
                    document.title = 'Single Page - 1';
                    con.innerHTML = " ";
                    break;
                case 2:
                    document.title = 'Single Page - 2';
                    con.innerHTML = " ";
                    break;
                case 3:
                    document.title = 'Single Page - 3';
                    con.innerHTML = " ";
                    break;
                case 4:
                    document.title = 'Single Page - 4';
                    con.innerHTML = " ";
                    break;
                default:
                    break;
            }
        }
    </script>
</head>
<body>
    <ul>
        <li><a href="#" onClick="viewContents(1);"> </a></li>
        <li><a href="#" onClick="viewContents(2);"> </a></li>
        <li><a href="#" onClick="viewContents(3);"> </a></li>
        <li><a href="#" onClick="viewContents(4);"> </a></li>
    </ul>
    <div id="contents">
         .
    </div>
</body>
</html>
```

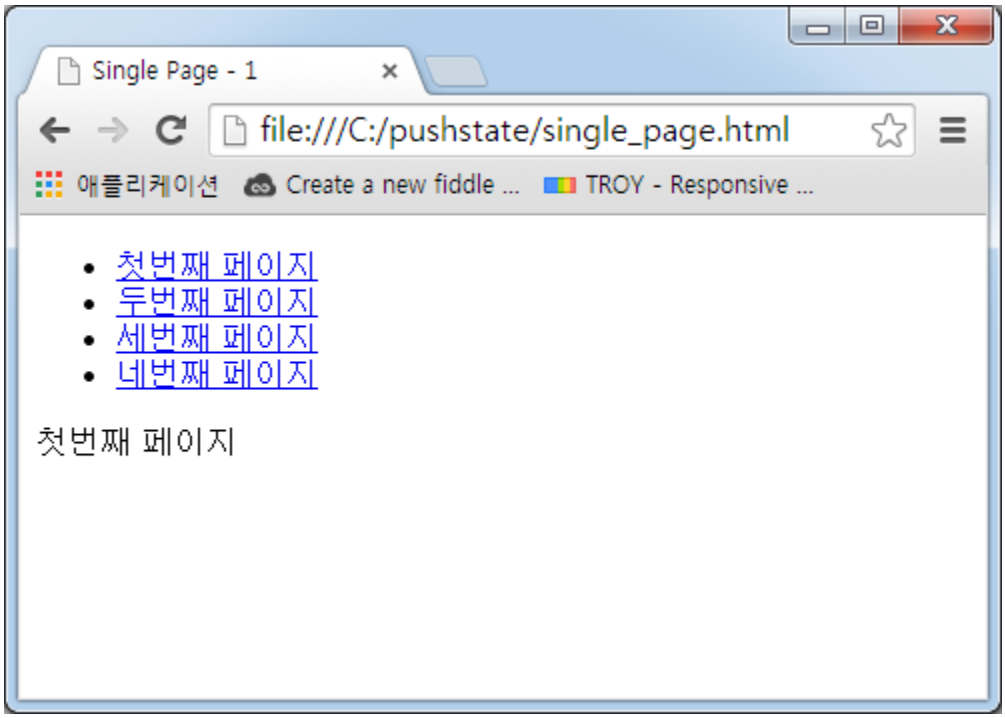
샘플소스

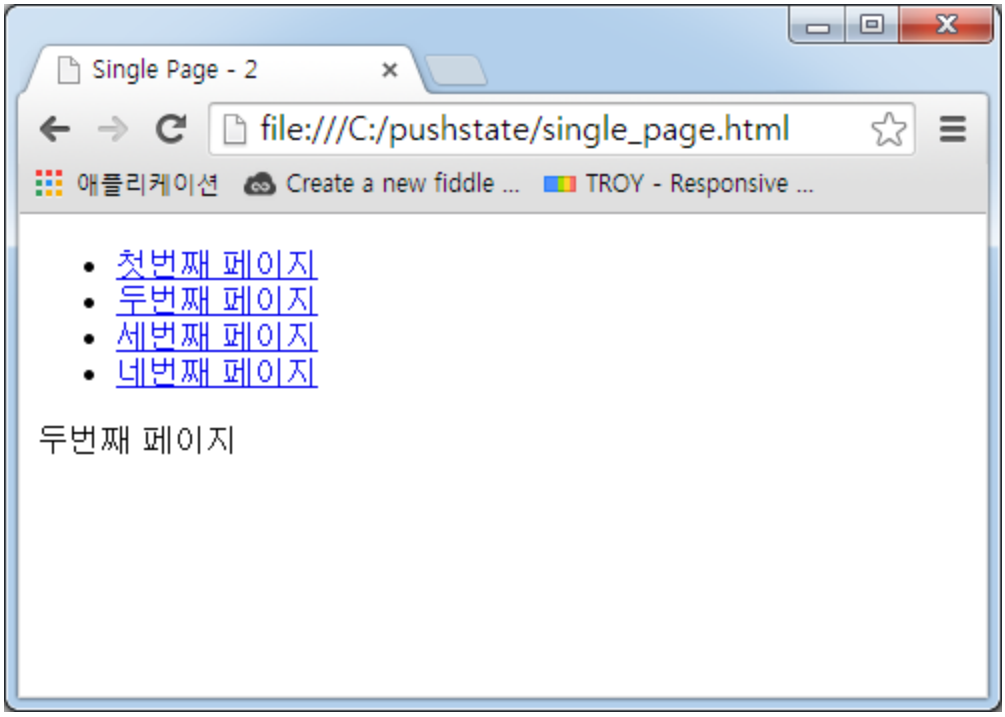
- single_page.html

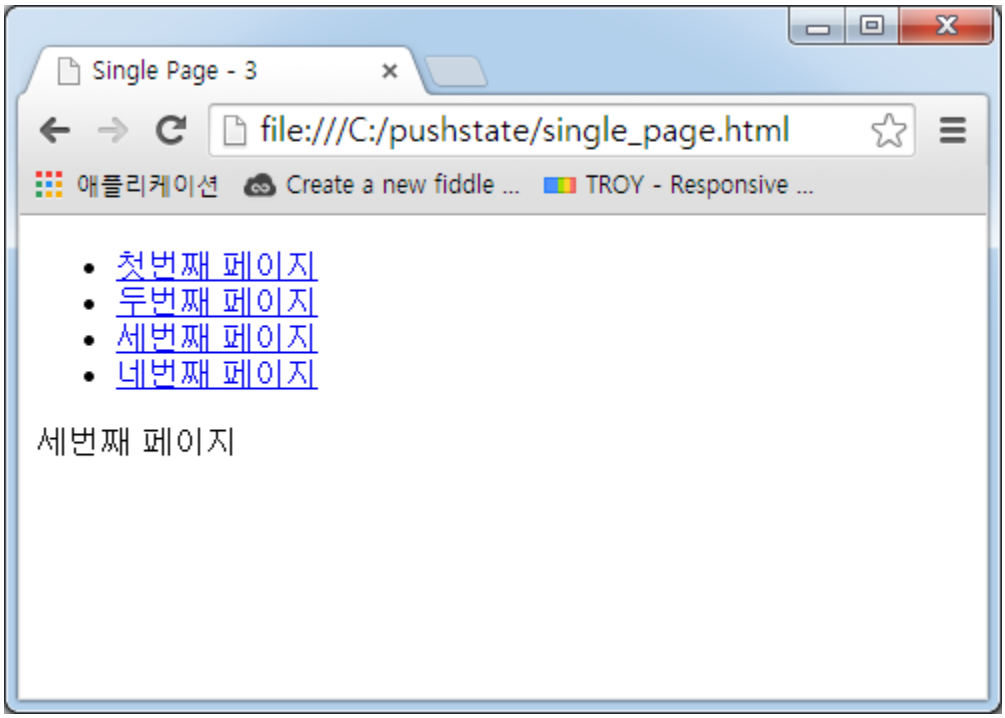
실행결과

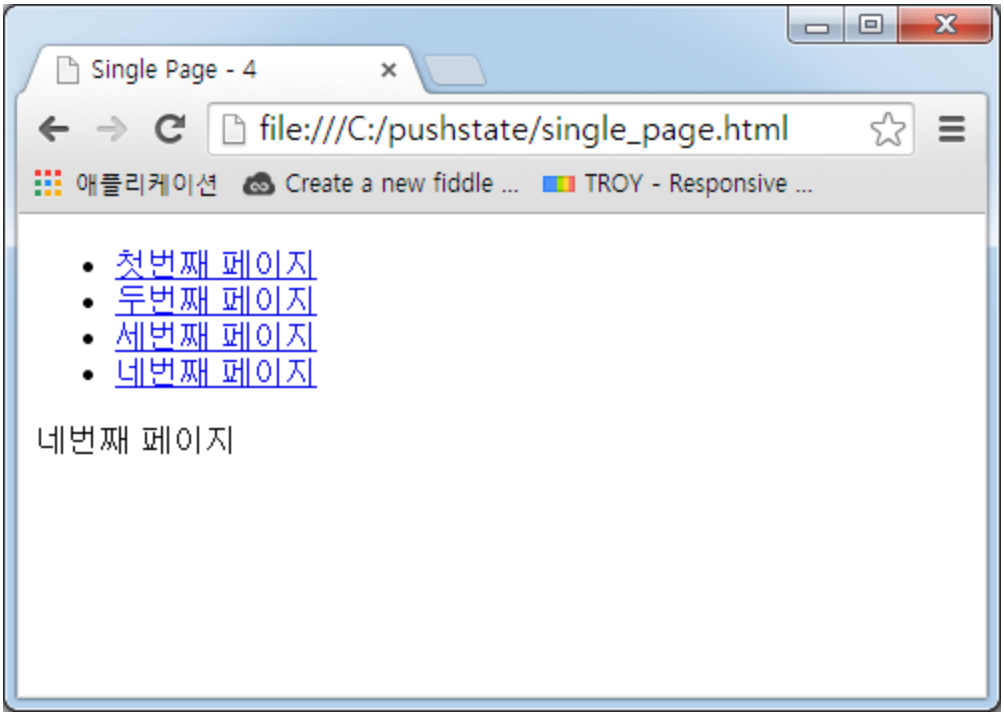
- javascript를 이용하여 타이틀과 페이지의 내용을 변경
- url의 변경은 없음
- backspace 버튼를 통한 history back 지원하지 않음.


Single Page
file:///C:/pushstate/single_page.html
애플리케이션
Create a new fiddle ...
TROY - Responsive ...
첫번째 페이지
두번째 페이지
세번째 페이지
네번째 페이지
페이지를 선택해주세요.


Single Page - 1
file:///C:/pushstate/single_page.html
애플리케이션
Create a new fiddle ...
TROY - Responsive ...
첫번째 페이지
두번째 페이지
세번째 페이지
네번째 페이지
첫번째 페이지


Single Page - 2
file:///C:/pushstate/single_page.html
애플리케이션
Create a new fiddle ...
TROY - Responsive ...
첫번째 페이지
두번째 페이지
세번째 페이지
네번째 페이지
두번째 페이지


Single Page - 3
file:///C:/pushstate/single_page.html
애플리케이션
Create a new fiddle ...
TROY - Responsive ...
첫번째 페이지
두번째 페이지
세번째 페이지
네번째 페이지
세번째 페이지



pushstate 를 이용하는 경우

- ancher 태그를 클릭하는 경우 pushstate 메소드를 이용해 상태를 저장한다.
- onpopstate 이벤트를 선언한다.

pushstate.html

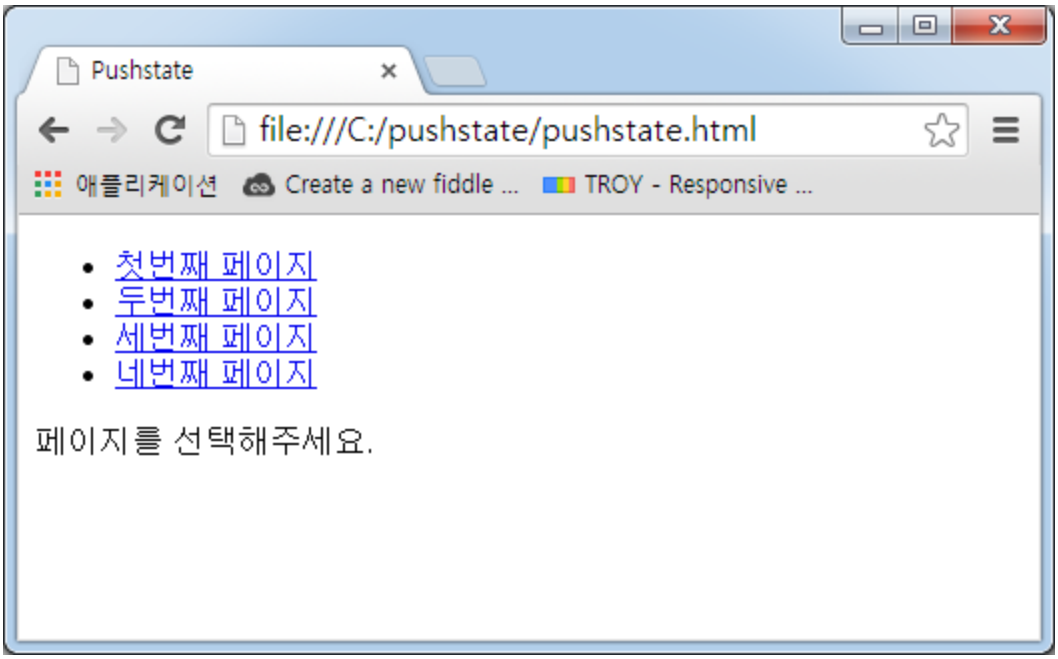
```
<!DOCTYPE html>
<html lang="ko">
<head>
    <title>Pushstate</title>
    <script type="text/javascript">

        window.onpopstate = function(event) {
          //alert("location: " + document.location + ", state: " + JSON.stringify(event.state));
            loadStateContent(event.state);
        };

        function viewContents(idx) {

            event.preventDefault();

            var con = document.getElementById("contents");

            switch (idx) {
                case 1:
                    history.pushState({page: 1, name:''}, 'PushState - 1', '?page=first')
                    con.innerHTML = " ";
                    break;
                case 2:
                    history.pushState({page: 2, name:''}, 'PushState - 2', '?page=second')
                    con.innerHTML = " ";
                    break;
                case 3:
                    history.pushState({page: 3, name:''}, 'PushState - 3', '?page=third')
                    con.innerHTML = " ";
                    break;
                case 4:
                    history.pushState({page: 4, name:''}, 'PushState - 4', '?page=fourth')
                    con.innerHTML = " ";
                    break;
                default:
                    break;
            }
        }

        function loadStateContent(state) {
            var con = document.getElementById("contents");
            con.innerHTML = " " + state.page + "  " + "hello " + state.name;
        }
    </script>
</head>
<body>
    <ul>
        <li><a href="#" onClick="viewContents(1);"> </a></li>
        <li><a href="#" onClick="viewContents(2);"> </a></li>
        <li><a href="#" onClick="viewContents(3);"> </a></li>
        <li><a href="#" onClick="viewContents(4);"> </a></li>
    </ul>
    <div id="contents">
         .
    </div>
</body>
</html>
```

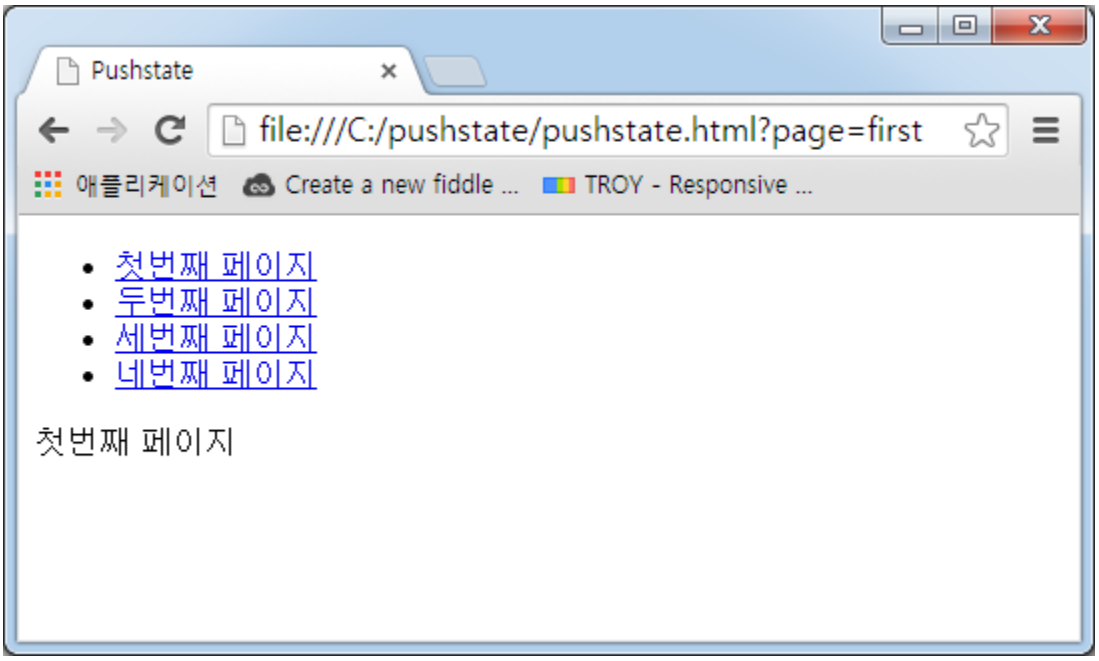
샘플소스

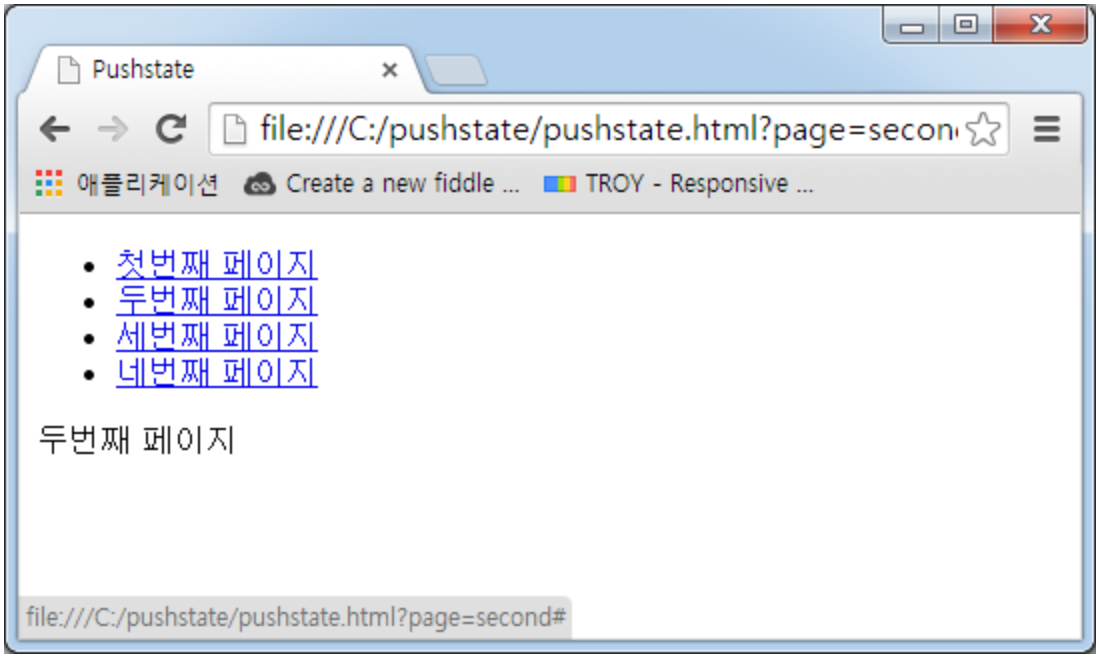
- pushstate.html

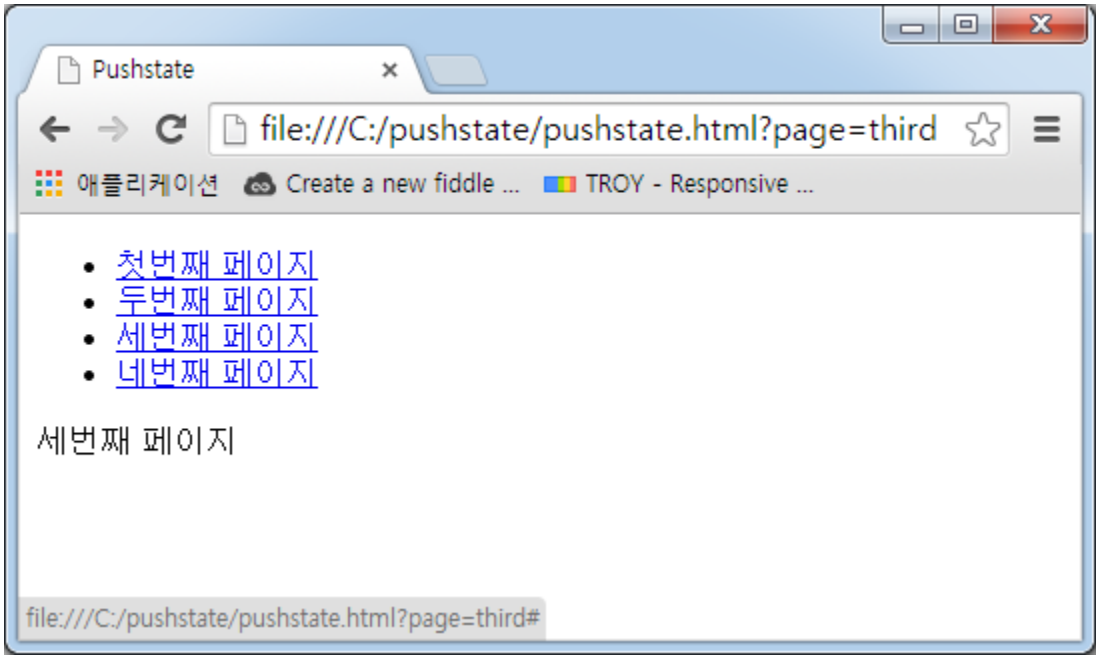
실행결과

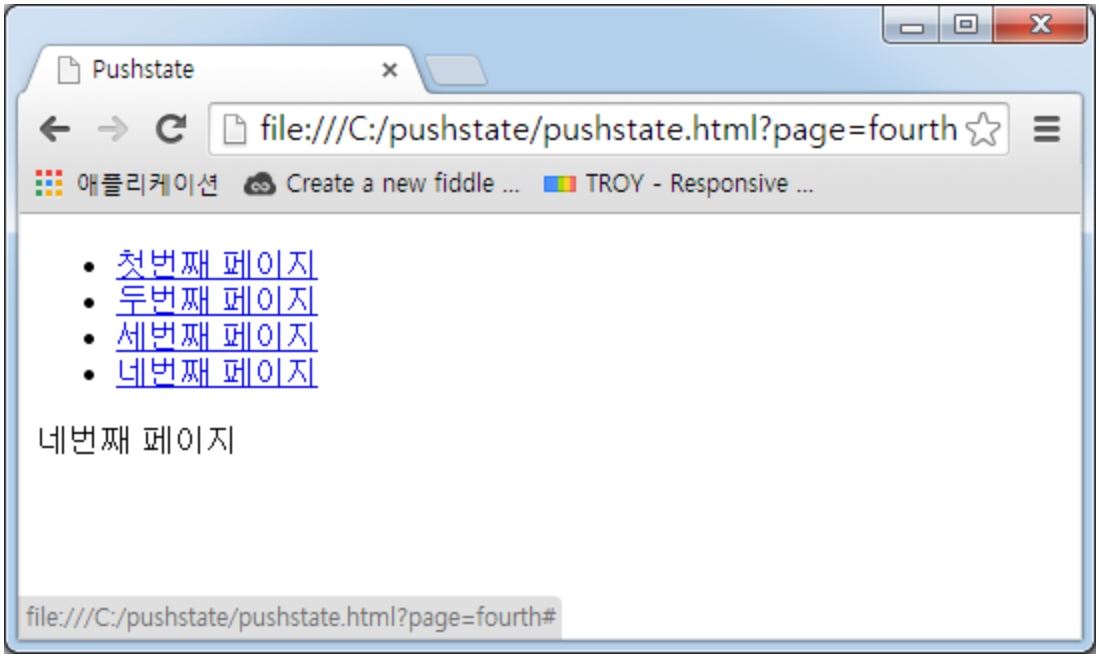
- pushstate를 이용하며 화면의 변화를 준 이후에는 backspae 버튼 등을 이용하여 history back, forward 등의 자유로운 이동이 가능
- 화면 전환이 일어나지 않았음에도 url에 변화를 줄수 있음
- pushstate 저장시 다양한 데이터(예제: 이름)를 저장하고 이를 나중에 호출할 수 있음.

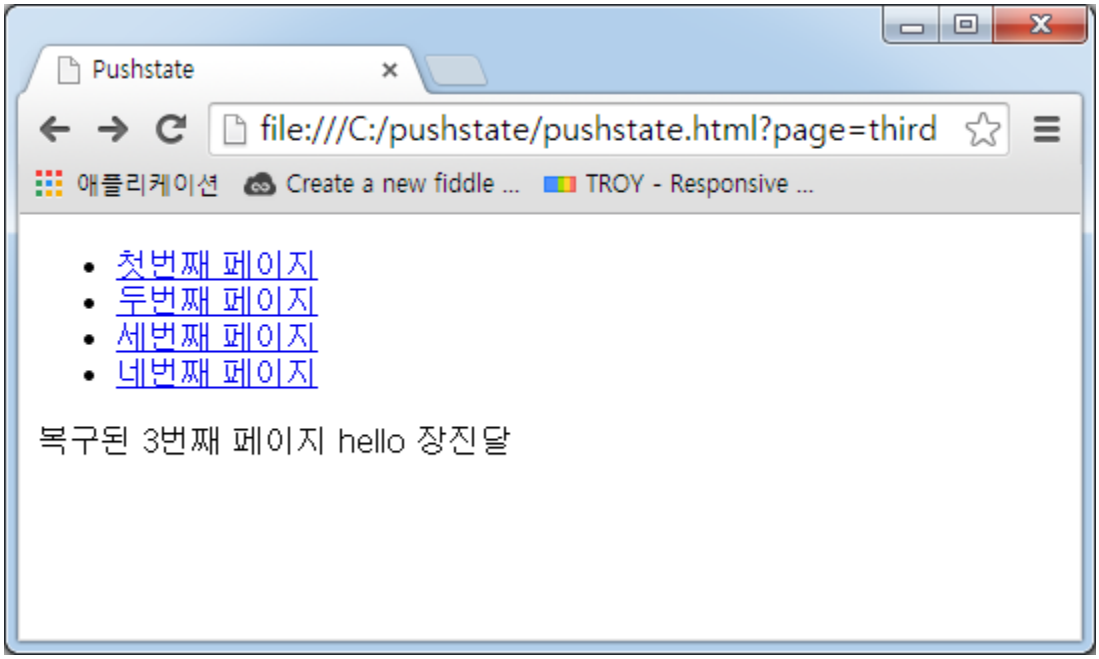


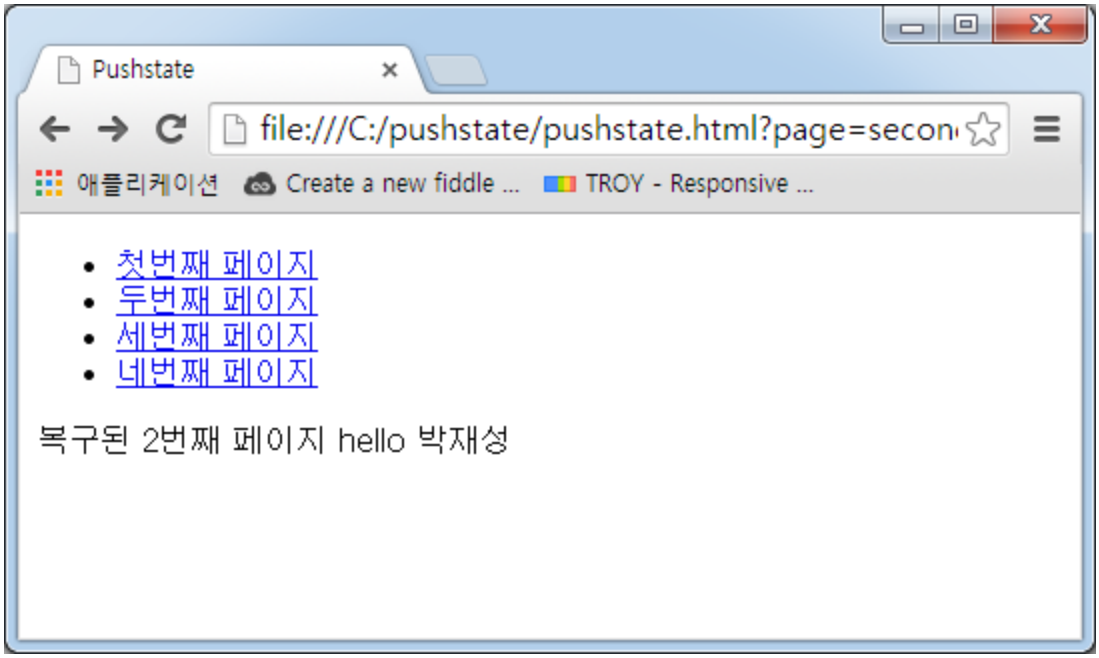


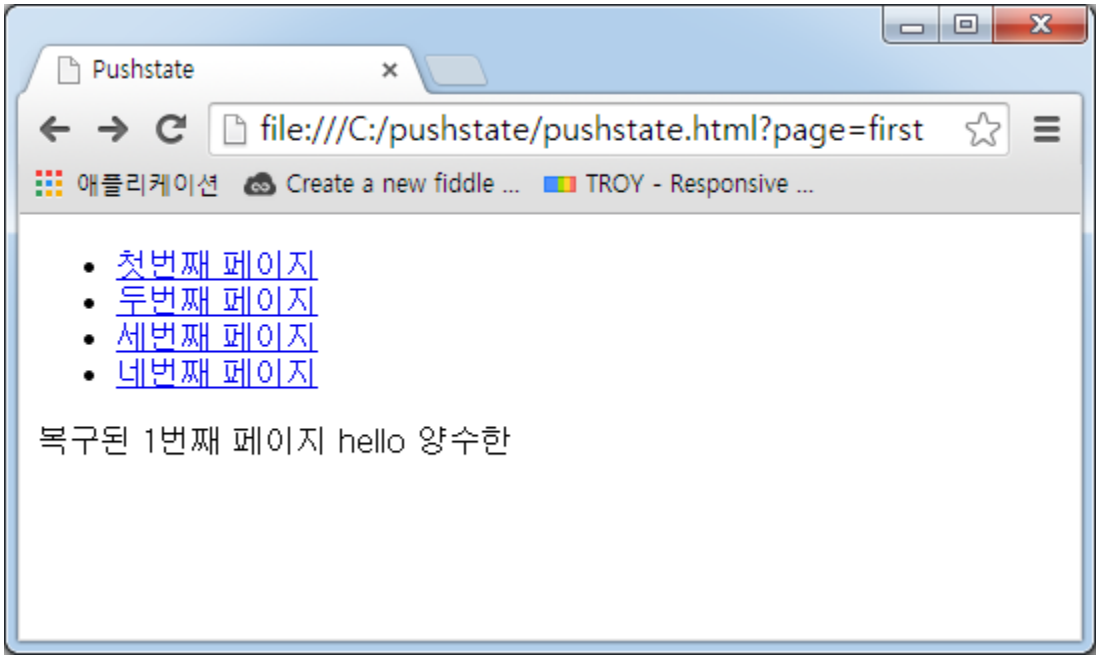

Pushstate
file:///C:/pushstate/pushstate.html?page=secon
애플리케이션
Create a new fiddle ...
TROY - Responsive ...
첫번째 페이지
두번째 페이지
세번째 페이지
네번째 페이지
두번째 페이지
file:///C:/pushstate/pushstate.html?page=second#

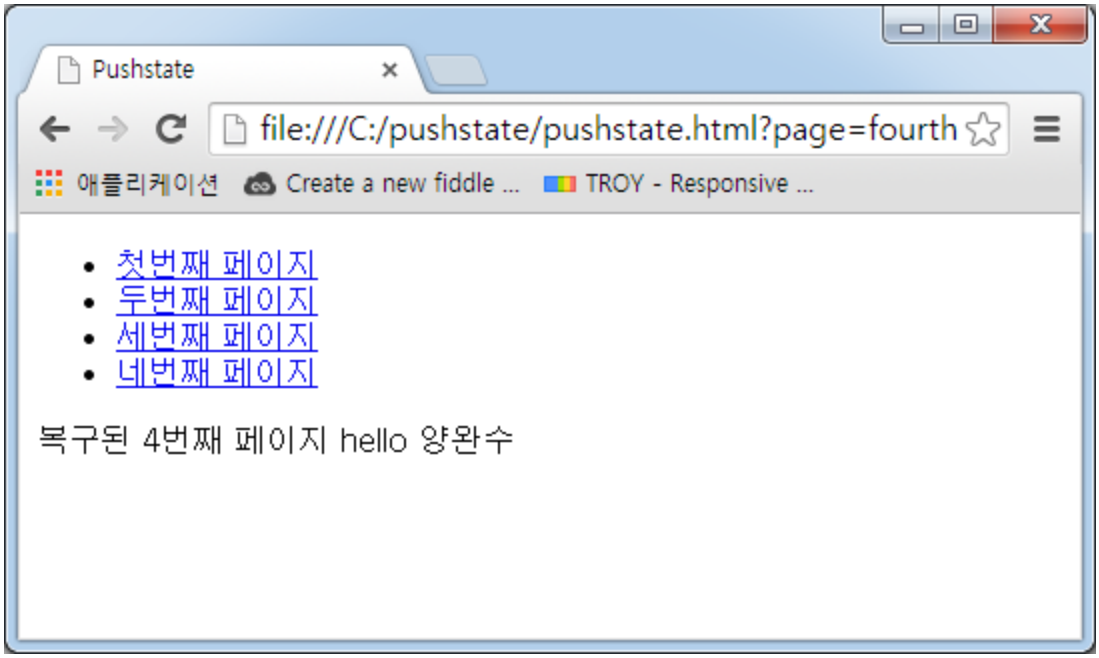

Pushstate
file:///C:/pushstate/pushstate.html?page=third
애플리케이션
Create a new fiddle ...
TROY - Responsive ...
첫번째 페이지
두번째 페이지
세번째 페이지
네번째 페이지
세번째 페이지
file:///C:/pushstate/pushstate.html?page=third#


Pushstate
file:///C:/pushstate/pushstate.html?page=fourth
애플리케이션
Create a new fiddle ...
TROY - Responsive ...
첫번째 페이지
두번째 페이지
세번째 페이지
네번째 페이지
네번째 페이지
file:///C:/pushstate/pushstate.html?page=fourth#


Pushstate
file:///C:/pushstate/pushstate.html?page=third
애플리케이션
Create a new fiddle ...
TROY - Responsive ...
첫번째 페이지
두번째 페이지
세번째 페이지
네번째 페이지
복구된 3번째 페이지 hello 장진달


Pushstate
file:///C:/pushstate/pushstate.html?page=secon
애플리케이션
Create a new fiddle ...
TROY - Responsive ...
첫번째 페이지
두번째 페이지
세번째 페이지
네번째 페이지
복구된 2번째 페이지 hello 박재성


Pushstate
file:///C:/pushstate/pushstate.html?page=first
애플리케이션
Create a new fiddle ...
TROY - Responsive ...
첫번째 페이지
두번째 페이지
세번째 페이지
네번째 페이지
복구된 1번째 페이지 hello 양수한


Pushstate
file:///C:/pushstate/pushstate.html?page=fourth
애플리케이션
Create a new fiddle ...
TROY - Responsive ...
첫번째 페이지
두번째 페이지
세번째 페이지
네번째 페이지
복구된 4번째 페이지 hello 양완수