

Design by Design pattern

Author : Rhio.kim

Date : 2008.07.01

Blog : blog.ecmas4.com

Mail : Rhio.kim@gmail.com

Version : 1.0.1 / Update : 2008.07.05

Pattern 이야기를 하기 전에 국내 인터넷 인프라에 대해 간단하게 언급하고 갑니다. 국내의 인터넷 인프라는 어느 국가보다 잘되고 있다고 해도 과언이 아닙니다. 메가패스의 FTTH를 시작으로 100Mbps 급 광랜이 일반 가정집까지 들어가고 광랜이 아닐지라도 10Mbps 급, ADSL급이 전국으로 뻗어있습니다.

하지만 국외의 경우에는 아직도 모뎀을 사용하는 지역이 매우 많습니다. 이런 특징에서인지 우리의 웹 개발의 분위기와 외국의 개발 분위기는 디자인에서부터 확실한 차이를 보인다는 것쯤은 대부분의 웹 개발자라면 한번쯤 생각해봤을 것이다.

외국은 최저 사양과 최고 사양의 폭이 너무 크고 그러한 조건을 모두 만족할 수 있는 웹 서비스를 구축하다 보니 이미지를 줄여두고 텍스트 기반의 콘텐츠 구성을 통한 서비스가 주가 되었다.

이처럼 인터넷 인프라에 따라 웹 서비스의 방향과 분위기도 차이가 생기기 마련입니다. 최근 Ajax UI, Rich UI가 웹 서비스에서 하나의 Trend로 자리잡아가고 그에 따라 다양한 Javascript framework이 개발되어지고 있습니다.

이러한 javascript framework에서도 그런 분위기를 느낄 수 있을 뿐 아니라. Javascript의 객체 지향 개발을 위한 부분, DOM 표준을 준수하여 Ajax UI, Rich UI를 위한 크로스 브라우징 유연하게 제공합니다. 이런 부분을 통해서 DOM, CSS 의 표준이 얼마나 중요한지도 이해할 수 있게 되었습니다.

그만큼 Dynamic UI를 위해서 DOM 설계 및 CSS 설계까지 매우 중요한 부분이 되었습니다. 여기에서 다음 이야기는 바로 Design 에 의한 설계 패턴입니다.

Design에 의한 Rich UI 설계는 HTTP를 사용하는 브라우저에서 사용되는 네트워크 리소스 사용률, 메모리 리소스를 최소화 하여 사용자 하여금 최적화된 UI를 개발하기 위함입니다.

여기서는 HTTP를 사용하는 브라우저의 네트워크 리소스 사용률, 메모리 리소스를 최소화라는 기술에 대한 내용은 언급하지 않습니다.

Design by Design Pattern 정의

모티브 : HTML 디자인을 개발자에 의해서 동적으로 생성되도록 처리하려면 개발자에 의해서 디자인을 javascript 의 정해진 변수에 넣고 프로세서에 맞춰 필요한 디자인을 Document 에 렌더링을 해야합니다. 유지보수 시 디자인적인 추가 요소가 발생할 경우 개발자가 디자인을 수정하거나 편집해야 하는 일이 발생한다.

목적 및 장점 :

1. HTML 로 구성되어진 레이아웃을 (HTML 파일 자체를) 개발자에 의해서 추가되거나 수정되지 않는데 목적을 둡니다. 즉 디자인과 개발의 완벽한 분리를 꾀하는데 목적을 둠
2. 유지보수에 있어 디자인과 개발의 업무량을 최소화 하기 위함
3. 초기 접속 시 불필요한 동적 렌더링을 피할 수 있다.

조건 :

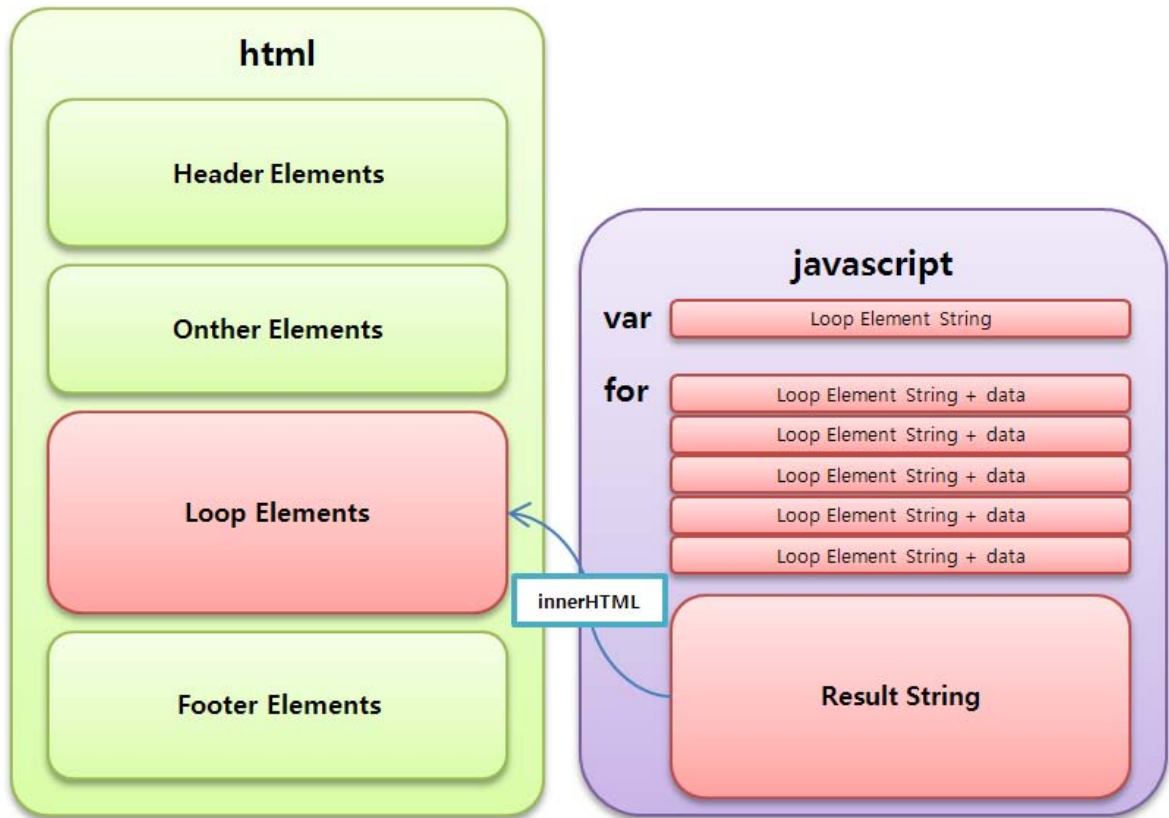
1. 디자인 시 거의 모든 부분이 Div 구조로 디자인 되어야 한다. Table 과 같이 다른 특성을 가지고 있는 영역끼리 하나의 Layout 으로 연관되어 질 경우 개발자에 의해서 처리되는 로직이 매우 복잡해진다.
2. 디자인 시 특징이 다른 Div Element 에는 유니크 한 ID 를 부여할 수 있어야 한다.
3. 개발자가 ID 를 부여할 경우에 개발 설계에 의한 최소한의 ID 만 부여해야 한다.
4. 부분적으로 Table 을 사용해야 할 경우 개발자의 협의가 필요할 수도 있다.

제약 :

1. 개발자뿐만 아니라 디자이너도 DOM 에 대한 기술적 이해가 필요하다.

단점 :

1. HTML 의 구조가 변경되면 구조에 맞게 javascript 개발 부분도 변경되어야 한다. 즉 HTML 버전관리와 Javascript 버전관리가 함께 이뤄져야 합니다.
2. 페이지 접속 시 디자인 레이아웃의 흐트러짐 현상이 있을 수 있다.



위의 그림은 일반적으로 사용할 수 있는 구조입니다.

UI 상에서 게시판의 게시물 리스팅과 같이 반복되는 HTML 구조를 javascript string value 로 지정하여 for 문을 통해 string 을 만든 후 해당 element 에 innerHTML 합니다.

또한 모든 페이지에서 사용되는 Header Elements, Footer Elements 등도 미리 정의된 javascript 템플릿으로 동적으로 사용하는 경우가 종종 있다.

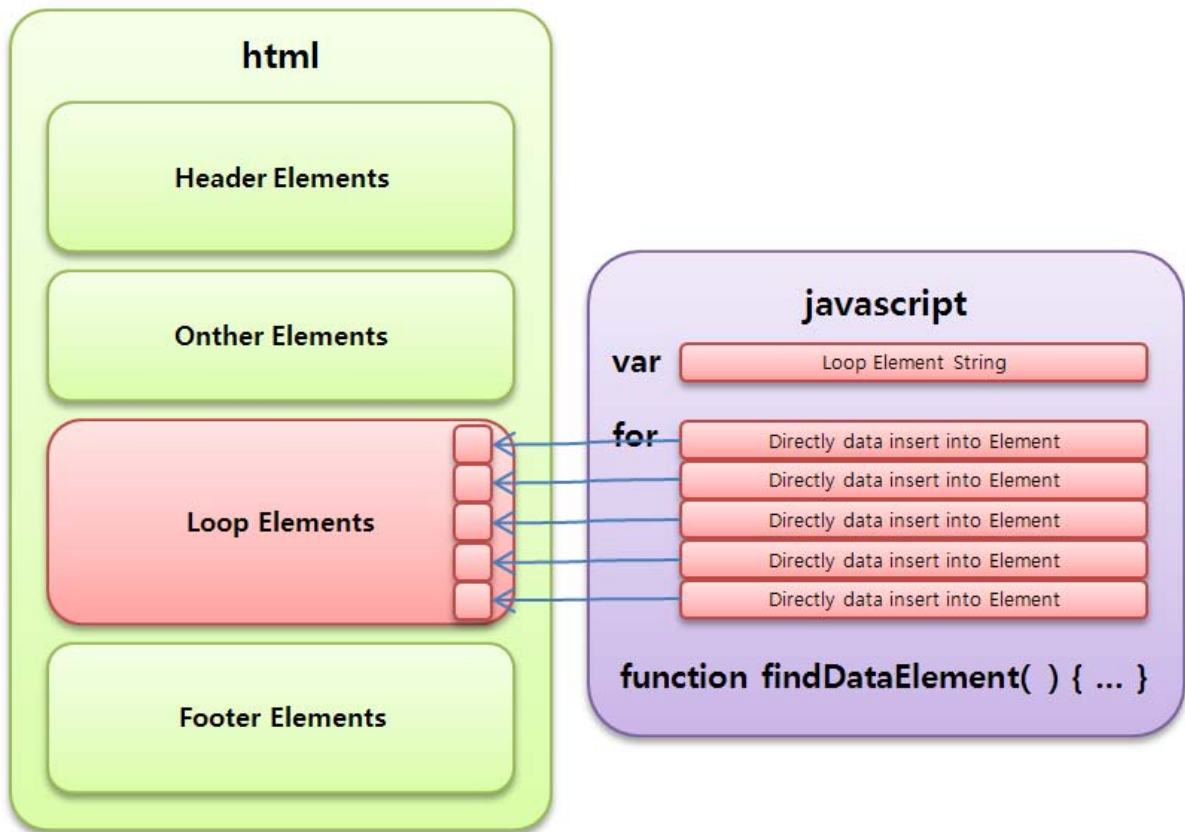
이런 경우를 경험했거나 글귀를 본적이 있나요?

DOM 을 생성하는 것보다 문자열로 innerHTML 하는 것이 빠르다. 그리고 문자열 합치기 보다 문자열을 배열에 쌓는 것이 더 빠르다.

그것보다 이미 렌더링 된 DOM Element 에 직접 입력하는 것이 더 빠르다.

몇만 건이라는 조건의 테스트가 아닌 실용적인 측면에서 가장 효율적인 방법이 이미 렌더링 된 DOM Element 에 입력하는 것이 더 낫지 않을까 싶다.

그 이유는 다음 예를 살펴보면서 설명한다.



위의 그림은 Loop Elements 를 구성하고 있는 이미 렌더링된 DOM Element 에 for 문을 돌면서 직접적으로 데이터를 입력하는 것을 나타낸 것입니다.

위에서 언급한 실용적인 측면을 언급한 것은 일반적으로 웹 서비스에서는 일정 해상도 내에 다양한 콘텐츠를 비주얼하게 표현할 수 있도록 디자인(설계)됩니다. 경우에 따라 동적으로 추가되거나 늘어날 수 있으나 그 부분은 일단 배제합니다.

그렇기 때문에 그림에서 언급한 Header Element 나 Footer Elements 는 고정이고 Loop Elements 의 경우에도 일정한 개수로 고정됩니다. 그렇게 데이터 량이 많아질 경우에는 페이지를 합니다. 이것은 화면 UI 가 변하는 게 아닌 표현할 데이터가 변하는 것이기 때문에 HTML 을 새롭게 문자열로 생성을 한다거나 할 필요가 없게 됩니다.

새롭게 생성하게 되면 브라우저는 새로운 DOM Element 를 DOM Parser 에 의해서 분석되고 다시 렌더링하게 되는 불필요한 일련의 과정을 반복 하게 됩니다. 또한 경우에 따라 동적 이벤트 관리에도 메모리 리크 발생하고 여러 가지 복합적인 요소가 발생할 수 있는 단점을 가집니다.

문자열을 생성해 innerHTML 하려고 할 때 하지 말아야 할 이유 중에 한가지 웹 서비스들을 자세히 살펴보면 거의 모든 화면 즉 UI 는 동적이나 정적이나 변화하는 것은 거의 없습니다. 변하는 것은 단지 UI 에 들어가는 데이터만 변하는 것입니다. 디자이너가 창조한 디자인에 필요한 영역에 데이터를 뿌려주면 됩니다.

그리고 한가지 장점 아닌 장점이라면 프로토타이핑 용 Rich UI 에 가장 적합한 패턴이라는 것입니다. 기획 의도가 자주 바뀌고(그러면 안되겠지만) HTML 엘리먼트 구조가 종종 바뀌게 될 경우 이와 같은 패턴을 이용하는 것도 좋습니다.

Ajax UI, Rich UI 의 개발 설계 방법으로 짧은 경험에서 나온 패턴입니다.
경우에 따라 좀더 보완해야 할 부분이 있기는 합니다.

최근 개발했던 서비스에 적용한 패턴입니다. 이유인 즉 기획적인 요소에 의해 디자인의 레이아웃이 너무 자주 변경되면서 첫번째 언급했던 방식으로 개발할 경우 개발자가 생성해 놓은 javascript 변수의 디자인(HTML) 문자열에서 추가, 편집을 해야 하는 일이 자주 발생하여 개발자가 디자인까지 수정해야 되는 일이 발생, 레거시 코드가 될 때 코드에 대한 이해 및 유지보수에 어려움이 발생

적용한 후 대응이 상당히 빨랐고 디자인을 직접 수정할 필요가 없었음.

Ajax UI, Rich UI 개발자가 디자인을 손대는 일을 없애도록 합시다. - By rhio.kim -