

이미지 파일을 Base64로 인코딩 하기

IE8이 정식으로 출시되고나면 가장먼저 해 보고 싶었던 것은 [Base64 Data URI](#)를 이용한 파일 호출이었습니다. 사실상 모든 메이저급 브라우저에서 Data URI를 사용할 수 있게 되는 중요한 시점이기 때문입니다. Data URI를 이용하면 브라우저가 이미지, 자바스크립트, 스타일시트 등 정적인 파일들을 호출하는데 있어서 개별적으로 네트워크 커넥션을 생성하지 않기 때문에 페이지 로딩 속도를 대폭 향상시킬 수 있게 됩니다. 실험적으로 스타일시트에서 호출하는 이미지 파일들들 대상으로 Base64 Data URI를 적용했습니다. 물론 IE6, 7을 위해 .background-image 핵으로 기존 이미지 경로를 보존 했지요. 부작용이 없다면 사용영역을 계속해서 확대할 예정입니다. 아래의 Base64 인코더는 위 작업과정에서 파생된 녀석인데요. 애용해 주세요.(업로드된 파일은 서버에 저장하지 않습니다.)

덧붙임.

역시나 부작용이 있네요. 과하면 독이 된다더니, 사이트에서 사용되는 모든 이미지를 인코딩할 경우 css 파일의 용량이 무려 400kb에 달하면서 페이지 로딩 속도가 오히려 감소하는 현상이 발생하였습니다. 아래는 Base64 Data URI를 사용할 때 나타나는 현상들과 주의 해야할 점들에

대하여 정리해 본 것입니다.

- IE7이하 브라우저에서는 불필요한 Data URI정보를 들고 있게 되어 더욱 느려진다. 재법 규모가 큰 프로젝트는 Data URI가 정의된 클래스들을 별도의 파일로 분리하여, 서버-사이드에서 브라우저의 Base64 인코딩 지원 여부를 알아내고 유효한 경우에만 호출되도록 하면 더욱 시맨틱한 웹 환경을 만들수 있겠다.
- CSS로 정의한 모든 이미지들을 Data URI로 사용하는 경우 용량이 매우 커지게 되므로 오히려 성능이 떨어진다. 재사용 빈도가 높고 비교적 작은 용량을 가진 이미지들을 Data URI로 사용하는 것이 바람직하다.
- 이미지 캐시 효과가 발생한다. 로딩 인디케이터 출력과 같이 돌발적인 상황에 즉시 반응하는 최상의 솔루션으로 자리잡을 것이다.(엑박과는 이별이란 말이다.)
- CSS 파일을 Gzip으로 압축 전송하고 Expires 헤더를 설정하는 것은 선택이 아닌 필수다.
- 그래픽 디자인 유지/보수가 어렵다. 변동사항이 생길때 마다 인코딩해야 하며, 파일명이나 경로명을 알 수 없어서 추가적인(소스파일, 용도 등을 주석으로 처리) 관리를 필요로 한다.
- 파이어버그에서 스크립트 응답 없음 현상이 발생한다.