

AMP - Accelerated Mobile Page

개요

- Instant Loading 이 장점인 AMP 를 소개 및 학습
- AMP 의 동작원리를 소개 및 학습
- 간단한 AMP 페이지 제작 실습
- AMP 구성요소 (이미지, 스타일, 컴포넌트) 소개
- PWA 와 AMP 를 결합하는 방법에 대해 학습

AMP (Accelerated Mobile Page)

- 모바일 기기에서 웹 사이트의 접근성을 높이기 위한 가속화 모바일 페이지
- 거의 즉시에 가까운 페이지 로딩을 위해 여러 기법들로 최적화
- AMP HTML / JS / [Cache](#) 로 구성되어 있다.
- 정적인 콘텐츠를 제공하는 페이지로 적합, 동적인 기능을 넣을 때는 AMP 컴포넌트 활용
- 구글 검색 엔진 알고리즘의 가점 및 상위 [노출](#), [SEO](#)

AMP 동작원리

- AMP 에 포함되는 모든 자바스크립트는 비동기 방식으로 실행된다.
 - 개발자는 AMP에 별도의 자바스크립트를 삽입할 수 없다.
 - 페이지 동작과 관련된 부분은 커스텀 AMP 요소를 통해 구현 가능
 - 별도로, iframe 안에 페이지의 렌더링과 관계 없는 외부 자바스크립트 라이브러리 포함 가능

- 이미지, 광고, iframe 과 같은 외부 리소스들의 사이즈와 위치를 지정해야함
 - 외부 리소스를 다운받기 전에 이미지 크기를 정하면 리소스 다운로드를 기다릴 필요 없이 페이지 레이아웃을 정할 수 있음
 - 폰트를 포함한 전체 문서의 레이아웃을 그리기 위해 단 한번의 HTTP 요청만 발생
 - 위와 같은 작업으로 Page Layout 재조정 및 Style Recalculation 과 같은 고비용 작업을 피함

- 커스텀 스크립트를 사용하는 경우 커스텀 태그 지정

```
<script async custom-element="amp-iframe"  
          src="https://cdn.ampproject.org/v0/amp-youtube-0.1.js"></script>
```

- 외부 라이브러리가 페이지 렌더링에 방해되지 않도록 제한
 - AMP 는 외부 js 라이브러리를 iframe 안에서만 허용하여 메인 페이지 렌더링에 방해되지 않음
 - iframes 에서 스타일 재계산이나 페이지 레이아웃 재조정시 DOM 크기가 작아 속도가 빠름

- 모든 CSS 스타일은 인라인만 허용, 사이즈는 제한
 - 인라인 스타일로 HTTP 요청을 최소화 -> 빠른 페이지 렌더링
 - 인라인 스타일 시트의 사이즈는 50KB 로 제한

- 효율적인 웹 폰트 요청

- 일반적으로 페이지 렌더링 시 외부 자바스크립트와 스타일 시트를 로딩하고 나서 폰트를 다운
- AMP에서는 폰트를 받기까지 외부 자바스크립트와 스타일 시트를 다운받는 HTTP 요청이 존재 X

- 스타일 재계산 최소화하기

- 페이지에서 요소를 변경할 때 스타일 재계산 발생 (전체 페이지 레이아웃 재조정 = 많은 비용 소모)
- AMP에서는 모든 DOM 을 화면에 그리기 전에 미리 요소의 위치를 다 정의
- 따라서, 한 프레임당 한번의 스타일 계산이 일어나 빠른 로딩이 가능

- 리소스 로딩 순서 재조정
 - AMP 는 리소스 다운로드를 모두 제어하여 중요도에 따라 순서를 재조정
 - 이미지나 광고는 필요한 경우 (스크롤 해서 보여지는 경우) 에만 미리 다운로드
 - Lazy loading 이 필요한 리소스는 pre-fetch 로 로딩 (로딩이 필요한 시점에 로딩)

- Instant Page Loading

- DNS Lookup, TCP Handshake 등을 미리 처리하는 **preconnect** 로 HTTP 요청을 빠르게 처리.
- pre-rendering 시 많은 양의 대역폭과 CPU 를 소비. AMP 자체 내부에서 이 부분을 최적화
- Instant Loading 이 될 수 있도록 초기 렌더링에 필요한 자원들에 집중한다는 점을 명심 (CPU 소비가 많은 리소스 일수록 뒤로 밀림)

AMP 실습 - Hello World

- 해당 [코드](#)로 간단한 AMP 페이지를 제작해본다

AMP 이미지 추가

- 아래와 같이 커스텀 태그로 이미지 삽입 가능

```
<amp-img src="welcome.jpg" alt="AMP Image" height="400" width="800"></amp-img>
```

- HTML 표준 태그들도 사용되지만 위와 같이 img 태그는 amp-img 로 변환하여 사용
- embed, param 등등 몇 개의 HTML 표준 태그들은 AMP 에서 사용할 수 없다. [참고](#)

amp-img 와 같은 별도 태그 사용 이유 : 해당 리소스가 로딩되기 전에 페이지 레이아웃을 정하고 레이저 로딩에 대한 네트워크 제어와 리소스 로딩 우선순위를 효율적으로 관리하기 위함

AMP 실습 - 이미지 추가

- 아래 태그를 실습 페이지에 추가해보자

```
<amp;img src="welcome.jpg" alt="AMP Image" height="400" width="800"></amp;img>
```

AMP CSS 스타일링

- AMP 의 요소에 대한 스타일링은 `<style amp-custom>` 태그를 이용한다.

```
<style amp-custom></style>
```


AMP 스타일링 주의사항

- 모든 AMP 페이지는 `<style amp-custom>` 태그 한 개만 포함할 수 있다. `< 50kb`
- HTML 요소 인라인 속성을 사용할 수 없다. 모든 스타일 규칙은 head 안에 선언되어야 함.
- `!important` 를 비롯하여 몇몇 표준 스타일 규칙을 사용할 수 없다.
- 외부 스타일 시트 참조 불가능 (커스텀 폰트 제외)

AMP 실습 - 스타일 조정

- 실습 자료에서 ul 의 리스트 속성 타입을 none 으로 변경

```
ul {  
  list-style-type: none;  
}
```

AMP 디버깅 및 유효성 검사

- 일반 HTML 문서와 동일하게 브라우저에서 파일을 보거나, 간단한 웹서버로 미리 보기가 가능하다.
- AMP 규칙 준수여부는 URL 끝에 `#development=1` 추가 후 개발자 도구 Console 에서 확인

AMP 검색 및 배포

- 같은 웹 콘텐츠에 대해 AMP, non AMP 페이지 2 개를 모두 갖고 있는 경우가 있다.
- 이러한 경우 `<link>` 를 이용해서 두 페이지를 연결한다.

- 일반 HTML 페이지

```
<link rel="amphtml" href="https://www.example.com/amp.html">
```

- AMP 페이지

```
<link rel="canonical" href="https://www.example.com/index.html">
```

- 만약 구분없이 AMP 페이지만 가지고 있을 경우

```
<link rel="canonical" href="https://www.example.com/amp.html">
```

Google 검색 엔진의 AMP 페이지 가이드 라인

AMP 컴포넌트

- 다양한 컴포넌트가 아직 없는 상황이라 목적에 맞는 컴포넌트가 있는지 선조사 필요
 - `<amp-tabs>` -> `<amp-selector>` .
- 없는 경우 대체할 수 있는 기존 컴포넌트를 활용할 수 있는지 확인
 - `<amp-img-lightbox>` : 사진 클릭하면 크게 확대됨

주의 : 컴포넌트가 다양하지 않으므로, 앱 설계 및 구현에 있어서 제공 여부부터 확인

AMP with PWA

- Web App Manifest 파일 추가 및 head 에 등록

```
<head>  
  <link rel="manifest" href="/manifest.json">  
</head>
```


- Service Worker 등록

```
<head>
  <script async custom-element="amp-install-serviceworker"
    src="https://cdn.ampproject.org/v0/amp-install-serviceworker-0.1.js"></script>
</head>

<body>
  <amp-install-serviceworker
    src="https://www.your-domain.com/serviceworker.js"
    layout="nodisplay">
  </amp-install-serviceworker>
</body>
```

amp-install-serviceworker spec

참고

- [AMP Official Site](#)
- [AMP Start](#)
- [AMP Slack](#)
- [AMP 는 어떻게 웹 페이지 성능을 높이는가?, Naver D2](#)

끝