

Siti web accelerati grazie a Google AMP Project

Non è una novità che Google voglia velocizzare il Web, e nemmeno che una delle sue iniziative al riguardo sia l'[AMP Project](#) (Accelerated Mobile Pages), un framework HTML in grado di generare pagine dalle funzionalità ridotte al minimo al fine di aumentare la velocità di visualizzazione.

Allo sviluppo stanno partecipando alcuni tra i Big dell'informazione, perciò non si può fare a meno di darci un'occhiata, soprattutto se state gestendo siti di news o testate giornalistiche. Tenete comunque conto che tutti i siti possono beneficiarne, perché se Google trova un modo più rapido di leggere i contenuti di un sito allora la SEO ne guadagna.

Leggendo la specifica, salta subito all'occhio che **sono vietati i tag *script*, *form* ed *iframe*** giusto per citare gli esempi più eclatanti. Altre (apparenti) esclusioni "eccellenti" riguardano i tag *img*, *video*, *audio*. Inoltre, un solo tag *style* è ammesso. Ci sono alcune eccezioni ma anche ulteriori importanti restrizioni sia HTML che CSS, per conoscere le quali vi rimando alla [documentazione ufficiale](#).

Per cercare di sopperire a tali mancanze, AMP HTML fornisce un set di [custom element](#) che sono gli unici a poter far uso di JavaScript o includere risorse esterne.

Niente jQuery, quindi, né tantomeno codice JavaScript custom. Il motivo di questa scelta così drastica è semplice: da quando Internet esiste, **le pagine web non sono mai state così pesanti** come quelle di oggi. Non è strano imbattersi in pagine che pesino più di 1MB con almeno tre banner, il che nell'era del mobile è una bella zavorra. Tramite JavaScript si caricano risorse, si fanno preload, si impegna il processore per eseguire slideshow. L'AMP Project vuole dar modo di pubblicare gli stessi contenuti in maniera più rapida, scremando le pagine da tutti i fronzoli e razionalizzandone il codice.

Per capire cosa comporta sviluppare in AMP-HTML non c'è niente di meglio che testare! Ho provato, quindi, a realizzare delle semplici pagine HTML poco più che testuali, durante lo sviluppo delle quali ho tratto delle conclusioni davvero interessanti.

Il codice nel tag HEAD

Questa è la prima questione che si affronta, ma per fortuna la documentazione è molto chiara.

A tag *html* (lo so, lo so, non si trova "nel tag *head*"...) va aggiunta la dicitura "amp" o, in alternativa, il simbolo "?".

Per quanto riguarda il contenuto dell'head, l'uso di molti dei tag obbligatori è già diventato la prassi, come i *canonical*, il *charset* ed il *viewport*. Ad essi si aggiungono uno *script*, uno *style* ed un *noscript*. C'è anche la possibilità di inserire un [JSON-LD](#).

Vediamo quindi come si presenta una pagina “hello word”.

È interessante notare che il tag *title* non è obbligatorio, e che la versione dello script di AMP inclusa è addirittura la... zero. Se ne deduce che il progetto è ancora nelle prime fasi di sviluppo.

Scrivere testo e link

Le uniche limitazioni sono che l'*href* dei link non deve iniziare con “javascript:” e che il *target*, se usato, deve sempre essere “_blank”.

Creare dei contenuti

Nessun problema quando si parla di inserire testo, ovviamente. Ma **com'è possibile inserire un'immagine visto che non possiamo usare il tag *img***, giusto per dirne una? Ebbene, attualmente Google AMP dispone di una serie di custom element per gestire varie funzionalità, e altri sono in fase di sviluppo. Vediamone alcuni.

Per inserire un'immagine, dobbiamo scrivere:

Come potete vedere, il tag è molto simile a quello al quale siamo abituati, e **deve contenere gli attributi *width* ed *height*, così come ogni altro tipo di risorsa esterna**. È consigliato anche l'uso di un placeholder in caso di errore di caricamento (un background grigio va benissimo). Impostando l'attributo *layout* a “*responsive*” l'immagine diventerà responsiva.

Il tag *amp-img* non supporta le gif animate, che vanno inserite usando il tag *amp-anim*.

Come potete notare, il tag *amp-anim* possiede anche l'utilissima capacità di mostrare un placeholder durante il caricamento! Per far sì che la pagina supporti le gif animate bisogna anche aggiungere nel tag *head* il seguente script:

Per incorporare un video da YouTube possiamo usare questo tag:

Come per le immagini, è possibile aggiungere l'attributo *layout=“responsive”* anche se funziona correttamente soltanto per i video con proporzioni 16:9.

Come anticipato, vi sono diversi altri custom element interessanti, tra i quali *amp.lightbox*, *amp-carousel*, *amp-audio* ed *amp-video*.

Personalizzare i CSS

Nel tag *head* possiamo inserire **un solo tag style** contenente codice CSS custom.

Come HTML, anche CSS ha una serie di restrizioni. Non è possibile, ad esempio, usare *@import*, *:not()*, ***, *!important*, *behaviour*, *-moz-binding*, *filter*, *overflow auto* e *scroll*. Transizioni e animazioni sono concesse soltanto per quelle proprietà che vengono accelerate dalla GPU (attualmente AMP supporta *opacity* e *transform*).

I font si possono inserire tramite *@font-face* o tramite il tag *link*, ma in questo caso soltanto verso fornitori accreditati (attualmente fonts.googleapis.com e fast.fonts.net). Consiglio, comunque, il secondo metodo.

Inserire un banner

E l'advertising? Com'è possibile includere un circuito banner se non possiamo usare il tag *iframe*? Google è ovviamente sensibile alla questione, e per questo ha creato due appositi custom element. Il primo è *amp-ad*, che però supporta soltanto i circuiti A9, AdReactor, AdSense, AdTech e Doubleclick. Il secondo è, guarda un po' chi si rivede, *amp-iframe*, che però ha delle limitazioni molto particolari: deve trovarsi molto più in basso del dal top della pagina, può contenere soltanto risorse https e infine viene sempre applicato il nuovo attributo HTML5 [sandbox](#).

Integrare Google Analytics

Qui la questione è stata un po' complessa, perché l'integrazione dei sistemi di tracciamento avanzati nel framework è ancora in fase di [sviluppo](#). Fino a quando non sarà completata, per integrare Analytics ho usato la sua versione "noscript", ovvero il pixel di tracciamento.

Dando uno sguardo ai parametri, notiamo che vengono usate alcune delle [variabili](#) fornite dal framework AMP, ovvero TITLE, CANONICAL_URL e RANDOM. Salta anche all'occhio che il [Client ID](#) (*cid*, obbligatorio) corrisponde all'id di una sessione, la quale durata, per restare in linea con le pratiche di Google, è stata impostata a due anni.

Questa soluzione però funziona male nel caso in cui le pagine vengano cacheate lato server, in quanto il *cid* sarebbe uguale per tutti i visitatori. Piuttosto, in un caso del genere potrebbe essere meglio valorizzarlo con RANDOM, anche se così facendo non si riuscirebbero ad individuare i visitatori unici.

Cookie Law europea

Dato che ho integrato Analytics, il sito deve rispettare la normativa europea in materia di cookie. Purtroppo non esiste ancora un componente in grado di mostrare un popup la cui chiusura può essere ricordata tramite cookie, ma prima o poi [arriverà](#), proprio per far fronte a situazioni come questa. Nel frattempo dovremo usare PHP per gestire i cookie e mostrare il messaggio in un *div*, anche se in questo modo chiudere il banner significa ricaricare la pagina.

Anche questa soluzione, purtroppo, è incompatibile con sistemi di cache lato server.

Il CDN di Google AMP

È possibile vedere le vostre pagine attraverso il servizio di cache lato server del progetto AMP componendo l'url in questo modo:

La `"/s/` indica di usare il protocollo HTTPS.

In questo modo però i font inclusi con `@font-face` non vengono trovati (`*.woff` - error 404), motivo in più per usare i fornitori autorizzati di cui sopra.

Validare il codice

È possibile validare le proprie pagine aggiungendo `#development=1` all'url e andando a guardare la console. Se avete intenzione di usare il CDN di AMP, sappiate che quest'ultimo controlla anche che l'html sia valido.

Un plugin per Wordpress

Immane come le zanzare d'estate arriva questo [plugin](#) che consente di visualizzare qualsiasi pagina in formato amp aggiungendo semplicemente `"/amp/` alla fine di ogni url.

Conclusioni

Google AMP è un subset HTML ancora in fase di sviluppo che implementa custom elements che razionalizzano il codice. L'aumento di velocità è per forza di cose notevole, in quanto è vietato l'uso indiscriminato di `javascript` ed `iframe`. Aumentando la velocità di fruizione dei contenuti migliora di conseguenza anche la SEO. Google AMP ha ancora alcune problematiche importanti per le quali ne sconsiglio l'utilizzo in siti di produzione (come la scarsa compatibilità con Google Analytics e la normativa europea in materia di cookie) ma lì si sta gradualmente risolvendo e a breve sarà possibile integrarlo efficacemente. Vale la pena tenerlo d'occhio!