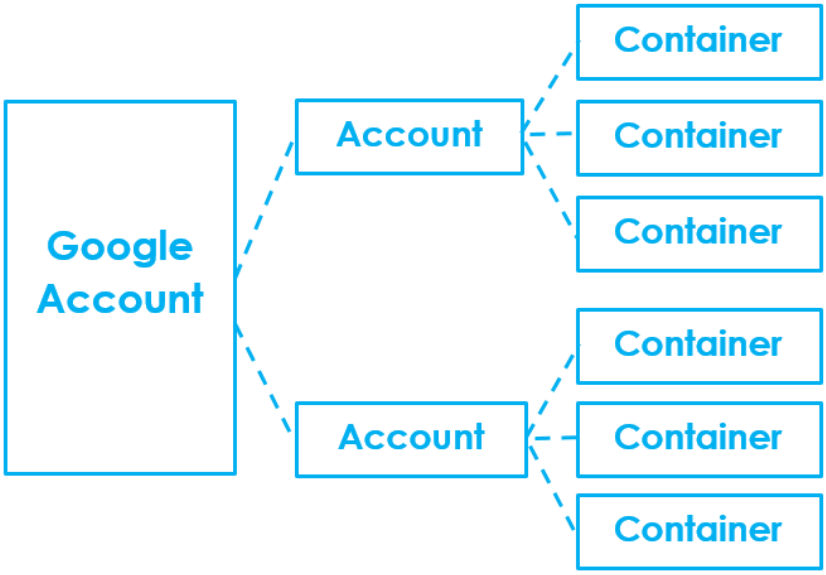
ULTIMATE

Google Tag Manager

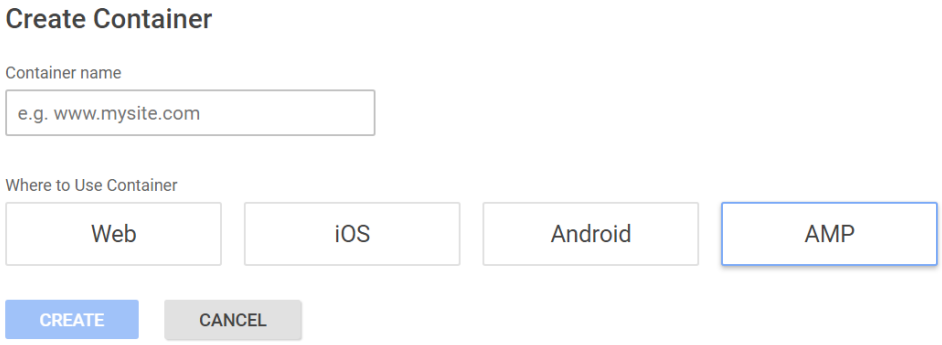
GLOSSARY

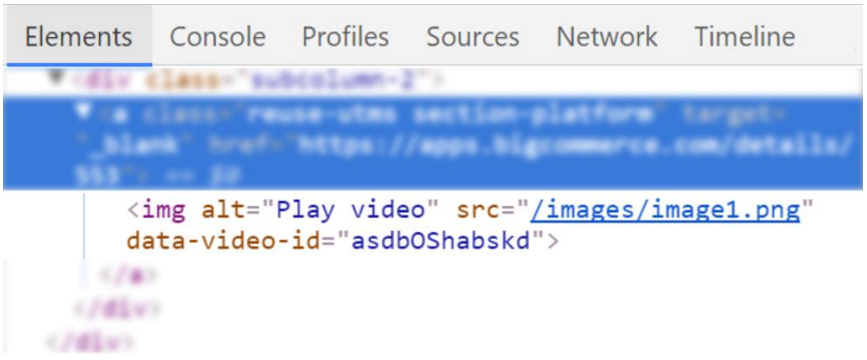
Analytics mania

By Julius Fedorovicius
Learn more about Google Tag Manager at analyticsmania.com
Linked In and Twitter

Term	Definition
1st Party Cookie	A variable which returns the value for the first party browser cookie with the name you specify in the Cookie Name field. For example, if you have a cookie called "name", you can use the 1st Party Cookie Variable to retrieve the value for this particular cookie. Learn more about how to set and read cookies with Google Tag Manager.
Account	First, let's not confuse *Google Tag Manager account* with Google account. Google account lets you access all Google products: Gmail, Drive, Adwords, GTM, etc. By concept, Google Tag Manager account is similar to Google Analytics account. Google Analytics accounts contain *Properties* (which contain *Views*), while Google Tag Manager accounts contain *Containers*. It's recommended to use one account per client/company and have separate containers for company's/client's Mobile Apps, Websites, etc. 
Activity History	Account activity displays information about the actions that have been done in the account. It is useful to determine which user has performed a certain action and when it was done.

Term	Definition
	You can find account activity by opening any **container** of the account you're interested in > **Admin** > **Account Activity.** **It does not contain activity of containers which belong to the account.** If you're looking to container history, see **Container Activity**.
Advanced Tag Settings	Additional tag settings, which let you control: ▪ Tag Firing Priority ▪ Tag Firing Schedule ▪ Tag Firing Options ▪ Tag Sequencing ▪ Ability to fire a tag only in published containers
Advertiser Tracking Enabled	A tag type which lets you track Adwords conversions. Usually, this tag fires when a visitor/user completes registration, purchase or any other important action. To start a new AdWords tag, from the home screen select **New Tag → Tag Configuration → AdWords Conversion Tracking**.
Adwords Remarketing Tag	A tag type which lets you install Adwords remarketing code. This enables you to show ads to people who have visited your desktop or mobile website, or (did not) complete a particular action. To create a new AdWords tag, from the home screen select **New Tag → Tag Configuration → AdWords Remarketing**.
AJAX	It stands for Asynchronous JavaScript and XML. AJAX is a set of techniques for creating highly interactive websites and web applications. It enables your browser to send and receive data without refreshing the window. A lot of web forms are created using AJAX and standard GTM Form submission trigger cannot track it. That's where AJAX listener (created by Lunametrics) comes in handy. Learn more how to track AJAX forms with Google Tag Manager.
AMP	It's an abbreviation of Accelerated Mobile Page. Simply put, Accelerated Mobile Pages is a stripped-down version of the mobile web which runs on a reinvented version of the language used to create web pages: HTML. This reimagined version of HTML, known as AMP HTML, removes most of the elements which cause web pages to load slower on mobile, like JavaScript and third-party scripts. Google Tag Manager and Google Analytics are also heavily reliant on JavaScript, thus it's not possible to implement them in a standard way.

Term	Definition
	When it comes to AMP and Google Tag Manager, you need to choose a different container type - AMP. 
Analytics Academy	API stands for **Application Programming Interface**, but basically, describes one way to plug your website/web application into another. Google Tag Manager has an API as well. This means that other developers can integrate their websites/systems with GTM. GTM API enables developers to control accounts, containers, folders, tags, triggers, etc. via other interfaces.
App ID	The value is set to the package name (Android) or bundle ID (iOS). A built-in variable of this type is provided in mobile app containers. It's not available in containers for websites.
App Name	A variable which is available only in mobile app containers. The value is set to the name of the currently running application. A built-in variable of this type is provided in mobile app containers; you don't need to define a new variable of this type.
App Version Code	A variable which is available only in mobile app containers. The value is set to the version of the currently running application.
Approval Queue	Due to its title, this term is surrounded by the confusion. Unfortunately, GTM currently does not have a proper approval workflow, and approval queue is not for accepting/declining changes in Google Tag Manager container. Approval queue is a feature **reserved for the DoubleClick Floodlight setup only** and is used exclusively as part of the DCF implementation process. You can learn more about approval queues here.

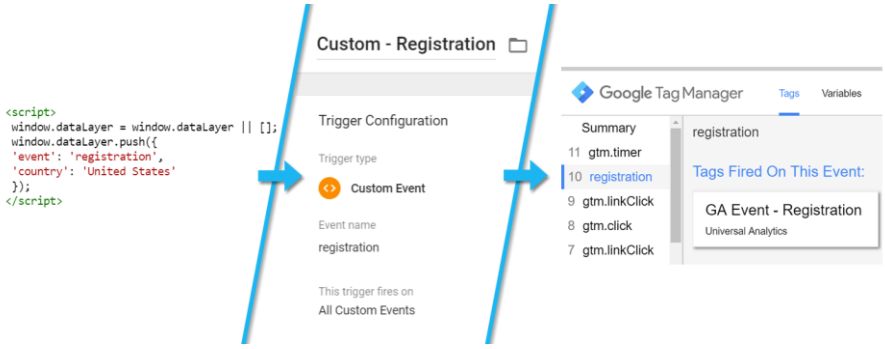
Term	Definition
Asynchronous loading	Unlike synchronous, asynchronous scripts can load simultaneously. They do not block page render and allows the browser to continue load elements while the JavaScript is being downloaded in the background. This significantly speeds things up, especially when the JS file is loaded from a remote server. Google Tag Manager, Google Analytics and the majority of other tracking scripts are loaded asynchronously.
Auto-Event Listener	A listener is a function which operates in the background. When creating the listener, you tell what operations it should wait for, and once these operations take place, the listener activates and fires any code within. In Google Tag Manager Web containers, default auto-event listeners *listen* to the following interactions: ▪ Clicks. ▪ Link clicks. ▪ Form submissions. ▪ JavaScript errors. ▪ Timer. ▪ History changes. In order to enable these auto-event listeners in Google Tag Manager, you need to toggle the corresponding triggers (e.g. *All Clicks* trigger, etc.).
Auto-Event Variable	Auto-Event Variables are used to access the target element of an auto-event action (e.g. Click, Error, Form Submission). For example: When you submit a form, you can access form's ID. When you click an image, you can access image's **alt** attribute (like in the image below).  Learn more about the Auto-event variable in GTM and how/where to use it.

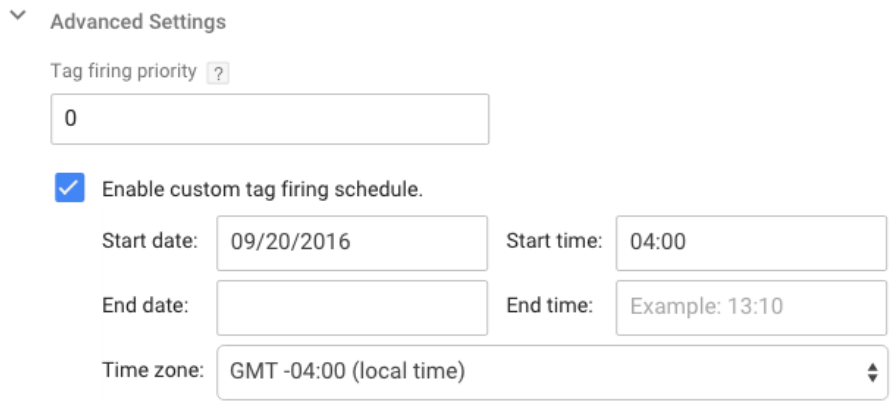
Term	Definition
Browser Limits	Although there are no limits of how many tags can be fired in Google Tag Manager, you should keep in mind browser limits. For web containers, Tag Manager can only fire tags within the capabilities of the browser. Most browsers handle a maximum of 6 to 8 simultaneous HTTP requests for a single domain. If you have a high number of tags on the same domain firing under the same conditions, tags will only fire within this browser limitation.
Classic Analytics	It's a legacy and much more limited version of Google Analytics. Compared to the current version, Universal Analytics, the classic one uses a different library (ga.js), does not allow to track the same user across multiple devices, etc. For more differences, check out this blog post. If you're working with the older website/web app, here's a guide how to migrate from *ga.js* to *analytics.js*.
Cleanup Tag	In a tag, under *Advanced Settings*, you'll find a tag sequencing section which contains two options: ▪ ***Fire a tag before XXXX fires*** option, which is called a **setup tag**. ▪ ***Fire a tag after XXXX fires*** option, which is called a **cleanup tag**. A cleanup tag fires immediately after the primary tag has fired. To specify a cleanup tag, check the "Fire a cleanup tag when tag XXXXX is finished" checkbox and select the tag you wish to use from the menu. Check the "Don't fire Tag YYYYYY if Tag XXXXXX fails" checkbox if you wish to have the cleanup tag **only fire when the main payload tag fires successfully**.
Click Classes	A built-in variable which fetches element's CSS classes upon click. The same result can be achieved by using a user-defined variable with the following settings: ▪ Type – Auto-Event Variable. ▪ Variable type - Element classes.
Click Element	A built-in variable which returns an HTML element that was the target of a click. You can achieve the same result by using a user-defined variable with the following settings: ▪ Type – Auto-Event Variable. ▪ Variable type - Element.
Click ID	A built-in variable which returns **id** value of the clicked element. You can achieve the same result by using a user-defined variable with the following settings:

Term	Definition
	▪ Type – Auto-Event Variable. ▪ Variable type - Element ID.
Click Target	A built-in variable which returns a string contained in the **target** attribute value of the clicked element. The same result can be achieved by using a user-defined variable with the following settings: ▪ Type – Auto-Event Variable. ▪ Variable type - Element Target.
Click Text	A built-in variable which returns the visible text inside the clicked element. You can achieve the same result by using a user-defined variable with the following settings: ▪ Type – Auto-Event Variable. ▪ Variable type - Element Text.
Click – All Clicks	A trigger which fires when a visitor clicks **anything** on the page (image, link, etc.). In Preview and Debug console, this event is displayed as *gtm.click*.
Click – Just Clicks	A trigger which fires when a visitor clicks **any link** on the page. It listens only for clicks which propagate up to a link (<a/>) node. This means that you can click on a SPAN in a BUTTON in a DIV, but as long as there's a link somewhere up there, GTM will register the event as a link click. In Preview and Debug console, this event is displayed as *gtm.linkClick*.
Click URL	A built-in variable which returns the value of **href** (or **action**) attribute of the clicked element. You can achieve the same result by using a user-defined variable with the following settings: ▪ Type – Auto-Event Variable. ▪ Variable type - Element URL
Constant	A user-defined variable which returns the string you choose to type in the **Value** field. Its value never changes. Learn more about its use cases.
Container	Every Google Tag Manager account contains one or more containers. Currently, there are 4 types available: Web, iOS app, Android app, or AMP. Each container has one container snippet.
Container Activity	Container activity displays information about the actions that have been done in the container. It is useful to determine which user has performed a certain action and when it was done. You can find container activity by opening any **container** of the account you're interested in > **Admin** > **Container Activity.**

Term	Definition
Container ID	Container ID is a part of Google Tag Manager Container snippet. Pretty much like Google Analytics ID (UA-XXXXXX-XX), container ID allows GTM to distinguish one website from another. Its structure is *GTM, dash,* and a string of 7 random letters/numbers, e.g. *GTM-11122AA*
Container Snippet	The Google Tag Manager container snippet is a small piece of JavaScript and non-JavaScript code that you need to paste into your pages. It enables Tag Manager to fire tags by inserting **gtm.js** into the page (or through the use of an iframe when JavaScript isn't available). Without this code, Google Tag Manager would not work on your (client's) website.
Container Snippet Placement	Container snippet consists of two parts: <script> and <noscript>. It's highly recommended to copy and paste the <script> as close to the opening <head> part as possible on every page of your website. <noscript> should be placed immediately after the opening <body> tag of the page. While you can place both these codes in <body> only, you must not add them both in <head>. The <noscript> part renders an iframe, which is not allowed in <head>, otherwise you're website's HTML will be invalid.
Container Type - AMP	Since AMP (Accelerated Mobile Pages) remove most of the elements which cause web pages to load slower on mobile (like JavaScript and third-party scripts), regular Website containers will not work. Consequently, Google Tag Manager offers AMP containers which have a different set of tags, triggers, and variables.
Container Type - Android	A type of Google Tag Manager container dedicated to Android apps. It has a different set of tags, triggers, and variables. There are two SDK versions available: ▪ Google Tag Manager + Firebase - V5. New users should use this version. ▪ Legacy Android - V4.
Container Type - iOS	A type of Google Tag Manager container dedicated to iOS apps. It has a different set of tags, triggers, and variables. There are two SDK versions available: ▪ Google Tag Manager + Firebase - V5. New users should use this version. ▪ Legacy iOS - V3.

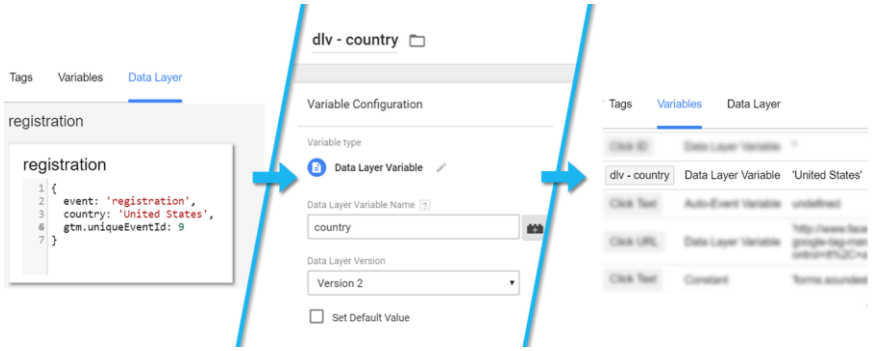
Term	Definition
Container Type - Web	Most common type of Google Tag Manager containers, designed for websites or web applications.
Container Version	A built-in variable which returns the version number of the container that is implemented on the site, or QUICK_PREVIEW if the version is in preview mode. You can achieve the same result by utilizing a user-defined variable - *Container Version Number* (no configuration is required).
Cookie	#1. A delicious dessert baked by your grandma. #2. A cookie is a tiny little file that's stored on your computer. It designed to hold a modest amount of data about a visitor. Marketing heavily relies on cookies, when it comes to remarketing, tracking behavior on a page (Google Analytics), etc. With help of Google Tag Manager, you can easily set and read cookies.
Cross-Domain Tracking	Cross-domain tracking is a problem any online marketer will face while trying to track goals with Google Analytics. It happens frequently and reduces our ability to really understand our best source of traffic. A typical case is an online store, hosted in one domain (www.mystore.com), and a sales page hosted under a different domain (www.mysalespage.com). Google Analytics can use only the cookie created by the domain itself, and by default, each domain is separate from the other domains. Each domain exists on their own, and therefore, cannot share information (by default). So when the visitor navigates from store to the sales page, a new session starts which leads to inaccurate data. Cross-domain tracking makes it possible for Analytics to see sessions on those two related sites as a single session.
CSS	CSS (Cascading Style Sheets) allows you to create great looking web pages. It's like a central place where all styles are defined. For example, *all buttons must be red, font size 30px, font Arial*. When there are multiple buttons (which look identical), you'll need to edit their style only once, because they all have the same CSS Class ***.red-button***. Otherwise, you'd have to edit buttons' style individually. This is what makes CSS a very scalable solution. When it comes to Google Tag Manager, you can distinguish interactions of elements with particular CSS classes/ID or other CSS selectors.

Term	Definition
CSS Class	CSS Classes are reusable styles that can be added to HTML elements. Two buttons may have the same class **.red-button** and look the same. Say that sometimes you want the font of the page element to be large and white, while other times you would prefer the font to be small and black. CSS would not be very useful if it did not allow you to have many different types of formats for a single element. For this purpose, one HTML element can have multiple CSS classes.
Custom Auto-Event Listener	A custom auto-event listener is a function which operates in the background which is waiting for the particular actions occur on the website. When the listener activates it pushes certain data to the Data Layer. In addition to the default auto-event listeners, you can easily add custom listeners to track interactions you need, for example, Facebook LIKE button clicks, Vimeo player, AJAX form submissions, and much more.
Custom Event	Custom events are used for tracking interactions that might take place on your website or mobile app that aren't handled by standard auto-event listeners. When a certain event is pushed to the Data Layer (by using *dataLayer.push*), Google Tag Manager does not catch it by default; you need to create a Custom Event trigger in GTM first. After the **Custom Event** trigger is created, you can use it as a condition to fire a tag. 
Custom Firing Schedule	Custom firing schedule allows you to indicate that the tag should fire only during your specified time window. To enable this, go to Tag's *Advanced Settings*, select **Enable custom tag firing schedule**.

Term	Definition
	
Custom HTML Tag	The Custom HTML Tag type allows you to manage a tag that isn't explicitly supported via Google Tag Manager, such as 3rd party tags, and have not yet been integrated into Tag Manager as a tag template. When creating a Custom HTML tag, you will be presented with a text box. Copy and paste your custom HTML and/or Javascript code where indicated.
Custom Image Tag	The Custom Image Tag type allows you to manage your own image tag with its own custom triggers and parameters. For example, if you use a pixel image for tracking, enter the image URL in the Image URL field. Select the Cache Buster checkbox and provide a cache-busting parameter to ensure that hits are registered from browsers that have cached versions of the image. Julian from Measureschool has created a video where he uses Custom Image tag to send data to his email marketing tool.
Custom JavaScript Variable	Custom JavaScript variable can be used to run arbitrary JavaScript on the page. Most popular use cases are: ▪ Name of the downloaded file. ▪ Is a visitor using a mobile device? ▪ Screen width, etc. All three aforementioned examples are explained in this blog post.
Data Layer	Technically speaking, a data layer is a JavaScript object or variable that stores and sends information from your site to Google Tag Manager (later that data is transferred to other tools, like Google Analytics). In other words – it's like a virtual layer of your website which contains various data points (Get it? That's why it's call Data Layer.).

Term	Definition
	Data Layer can contain various information, such as the page category or transaction value. Then, with help of Google Tag Manager, this data can be transferred to other tools, like Google Analytics or Mixpanel.

dataLayer.push	*dataLayer.push* lets you send additional data to the Data Layer when certain events occur on the website. For example, you have a newsletter signup form (which cannot be easily tracked with automatic GTM form listeners).
	
	In that case, you should ask your website developer to push the dataLayer event once a new subscriber has submitted his/her email address. The code of the event should look like this:
	
	`dataLayer.push({'event': 'new_subscriber'});`
	
	If you wish you can ask the developer for additional information (for example, **form position**).
	
	`dataLayer.push({`
	`'formPosition': 'footer',`
	`'event': new_subscriber`
	`});`

Data Layer Variable	When you create a Data Layer Variable, you specify the Data Layer key of which value you want to retrieve. By default, Google Tag Manager does not fetch all keys and values from the Data Layer, so you need to "teach it" by creating Data Layer variables.  After you've done it, the variable will become accessible in the **Variables** tab of the GTM Preview and Debug mode, and you'll be able to reuse it in any GTM tag. Learn more about how to access data in the Data Layer.
Data Layer Version 1	While creating a Data Layer variable you need to choose between version 1 and version 2 (default) of the Data Layer.

Term	Definition
	Version 1 of the Data Layer Variable has a very limited reach. Basically, it doesn't let you access nested keys, like here: `attributes: { pagePostAuthor: 'Julius Fedorovicius': { someOtherKey: 'example' } }` That's all the Version 1 does. There's no merging, no special data model functions, or anything else that you can use to further manipulate the retrieved data. If the variable returns a plain object, a new data push **would completely overwrite it**. Simo Ahava has explained it in greater detail.

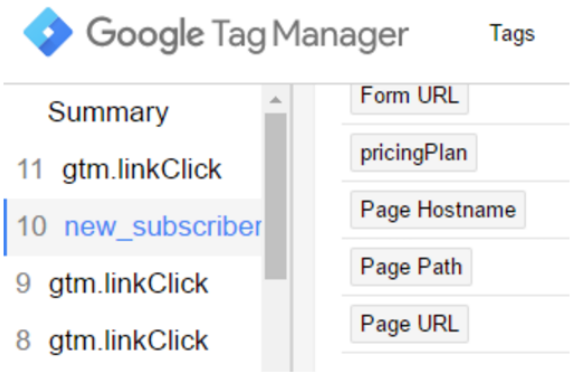
Data Layer Version 2	While creating a Data Layer variable you need to choose between version 1 and version 2 (default) of the Data Layer.
	
	Instead of **completely overwriting** existing keys (like in Version 1), Version 2 first checks whether a key with that name already exists in the data model. If it does, it recursively merges information in the new push with information in the existing object. It's easier to explain with the example.
	
	Here's the first Data Layer push:
	
	`window.dataLayer.push({ 'product' : { 'price' : '89.99', 'category' : 'Shoes' } });`
	
	And here's the second (there's a new key **name**):
	
	`window.dataLayer.push({ 'product' : { 'price' : '89.99', 'name' : 'Nike' } });`
	
	If it was Version 1, the final result would be, because the second data push completely removed previous data and wrote a new one.
	
	`Window.dataLayer.push({ 'product' : { 'price' : '89.99', 'name' : 'Nike' } });`

Term	Definition

	But with Version 2, the final result is like this (all three keys (**category, price, and name**) are intact):
	
	`window.dataLayer.push({`
	`'product' : {`
	`'category' : 'Shoes'`
	`'price' : '89.99',`
	`'name' : 'Nike`
	`}`
	`});`

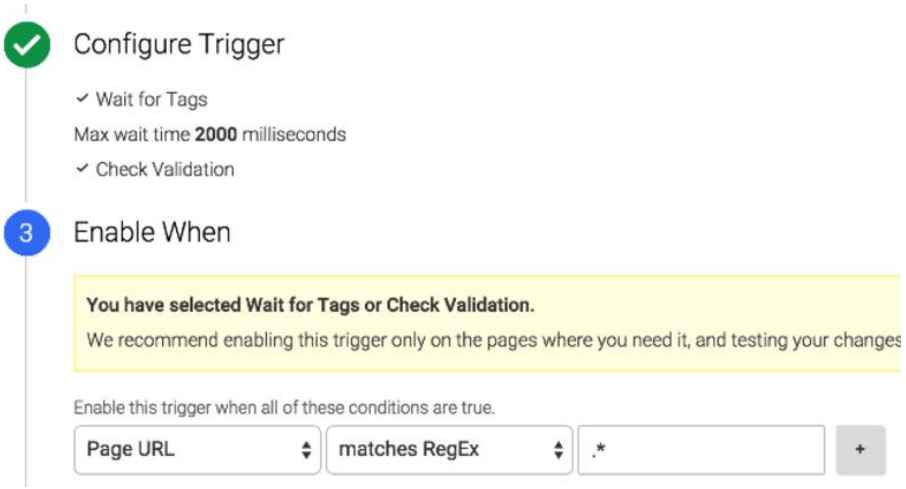
Debug Mode	A built-in variable. Its value is set to true if the container is being viewed in debug mode. There's also a user-defined variable under the same name.
Device ID	A built-in variable in mobile apps containers. For Android, the value is set to the device ID. For legacy iOS containers, the value is set to '' (the empty string). This variable is not available in Firebase (iOS) containers.
Device Name	A built-in variable in mobile apps containers. The value is set to the device name of the currently running application (e.g., "Samsung Android", "Android SDK built for x86"). A built-in variable of this type is provided in mobile app containers; you don't need to define a new variable of this type.
DOM (Document Object Model)	The Document Object Model (DOM) is a programming interface for HTML and XML documents. It represents the web page so that programs can change the document structure, style, and content. The DOM represents the document as nodes and objects. That way, programming languages can connect to the page. The Document object model provides a way for JavaScript to interact with all every single node (in other words, website element). In Google Tag Manager, you can access any element on the page and read its value. All thanks for the DOM Element variable. If you still feel like *WTF is DOM?*, then I highly recommend watching this video. It's targeted towards designers (who are also non-developers, like me).
DOM Element	You can use the DOM Element Variable to retrieve the text content of any given DOM Element. You can also use it to retrieve the value of any attribute of the DOM Element. If the value you're looking for was not set up in the data layer (see Data Layer Variable type, above), it's possible that the value can be retrieved from the DOM. Use this variable type if you can find the value you're looking for in the DOM by entering the value of the element's ID attribute.

Term	Definition
	A pretty common use case is to scrape DOM in order to track form submissions or to get product price.
DOM Ready	DOM ready is a Pageview-type trigger which occurs when the browser DOM is fully constructed (and *gtm.dom* Data Layer event is fired). You can see DOM ready event in the GTM Preview and Debug console (event list).
DOM Scraping	DOM scraping is the process when someone is fetching values by crawling the DOM. In the context of Google Tag Manager, with help of DOM variable, you can transfer any text from your website to other marketing tools, as long as it is selectable in the DOM. But keep in mind that this technique is risky because the developer can accidentally edit some nodes in the DOM which will break your "scraping implementation". If possible, ask developers to push the needed data to the data layer, instead.
DoubleClick Floodlight Counter	A 3rd party tag template. The DoubleClick Floodlight Counter tag allows you to count the number of times that users have visited a particular page after seeing or clicking one of your ads.
DoubleClick Floodlight Sales	A 3rd party tag template. The DoubleClick Floodlight Sales tag allows you to keep track of how many items users have purchased, as well as the total value of those purchases. You can learn more about this tag here.
Environment	Environments are browser cookies, which you use to link a Google Tag Manager container state with the browser of the user who needs to or wants to view that particular state. In other words, if you have a QA (quality assurance) process, or if you do most of your testing on a staging server you can create an Environment in GTM, after which you can publish container versions (even the draft) into that particular Environment alone. You can manage environments by going to **Admin > Container > Environments**.
Environment Name	A built-in variable which returns the name of the current environment (Live, Latest, Staging, etc.). There's also a user-defined variable under the same name.
Error Line	A built-in variable which returns a string containing the line number where the error was thrown. It is related to the JavaScript Error trigger.
Error Message	A built-in variable which returns a string containing the error message dispatched by a JavaScript Error Trigger.

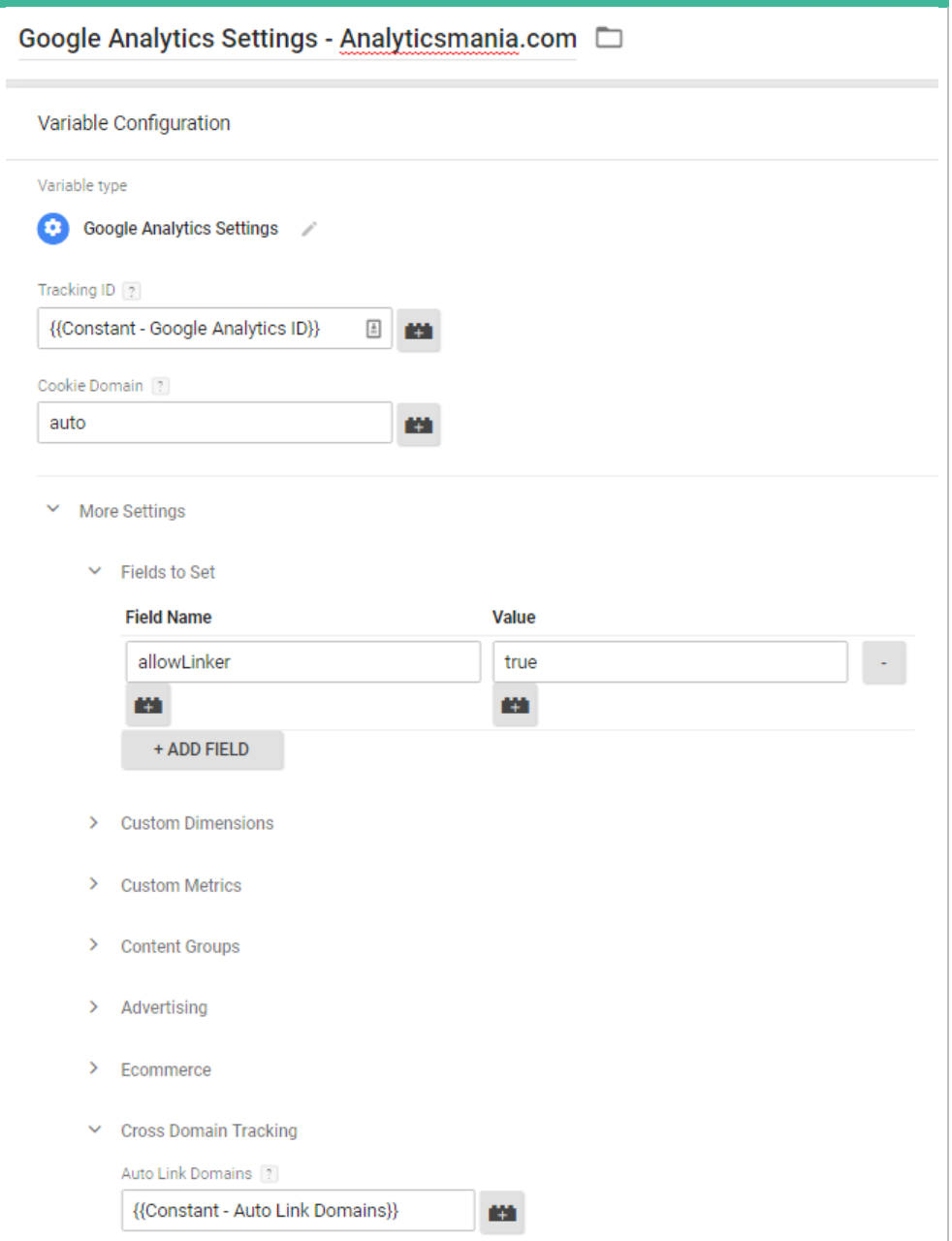
Term	Definition
Error URL	A built-in variable which returns a string containing the URL of the script where the error was thrown. This variable is related to the JavaScript Error trigger.
Event	Google Tag Manager event is an interaction which occurred on the web page and then was pushed to the Data Layer. If you enable GTM Preview and Debug mode, a special console will appear at the bottom of the screen. On the left side of that console is the list of Google Tag Manager **events**.  They actually play a very important role within the GTM – you can fire your Tags (scripts from various marketing tools) based on these Google Tag Manager events. A few examples to mention: A “Pageview” Google Tag Manager Event can be used as a trigger to fire Google Analytics Pageview Tag. A “gtm.linkClick” GTM event (which happens when a visitor clicks any link on your website) can trigger a Google Analytics Event Tag (which sends a particular event to GA).
Event (2)	A built-in variable which returns a string containing the value stored in the ‘event’ dataLayer key. There also a user-defined variable under the same name.

Event Name	A built-in variable in GTM containers for Mobile Apps. The value is set to **eventNameXYZ** when the following code in your app is executed:
	Android:
	`FirebaseAnalytics.getInstance(mContext).logEvent("eventNameXYZ", null);`
	iOS:
	`[FIRAnalytics logEventWithName:@"eventNameXYZ" parameters:parameters];`

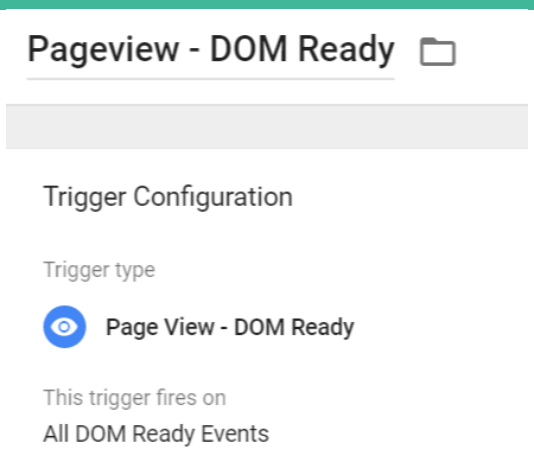
Term	Definition
Event Parameter	A built-in variable in GTM containers for Mobile Apps. The value is set to the value of a logged Firebase Analytics event parameter for the given key.
Export Container	Exported containers can be compared, modified, shared, and imported back into Google Tag Manager. Containers are exported as JSON text files. You can replicate a container setup by exporting the container and then importing the file into a different container. Or, you can modify the exported file to include only a subset of the triggers, tags, and variables before importing it. By exporting and importing containers, you can create and re-use container templates, and share them with others. You can export the container by navigating to **Admin > Container > Export Container**. A whole bunch of exported containers is available for free at our Google Tag Manager Recipes section.
External Account Links	**External account links** is a feature **reserved for the DoubleClick Floodlight setup only** and is used exclusively as part of the DCF implementation process. This is where linked DoublClick Floodlight accounts are displayed. To see existing account links, navigate to **Admin > External Account Links > Overview**. To unlink a DoubleClick Campaign Manager from the container, click the external account name, then click **Delete Link**.
Facebook Pixel Helper	The Facebook Pixel Helper is a troubleshooting tool that helps you validate your Facebook Conversion Pixel and Custom Audience Pixels. Using the tool you can verify whether the pixel is working properly, what events were fired and spot errors in no time. A small number will appear on the Facebook Pixel Helper icon to indicate the number of pixel events. When clicked, a panel will expand to show a detailed overview of the page's pixels, including warnings, errors, and successes. For more information, check these two blog posts: installing Facebook pixel and tracking FB conversions with Google Tag Manager.
Firebase User Property	A built-in variable in GTM containers for Mobile Apps. The value is set to the Firebase Analytics user property value for the given key.
Folder	As your containers grow over time, it can become difficult to keep track of all of your tags, triggers, and variables. With folders, you can organize these items into logical groupings, making them easier to work with for yourself and your team members. To see all folders within a container, click **Folders** in the left menu. Click a folder name to expand or collapse it in the list.

Term	Definition
	To file one or more items into a folder, select their checkboxes in the list and select a folder from the **Move** menu to place the items in an existing folder, or select **Folder from Selected** to add the items to a new folder.
Form Classes	A built-in variable which fetches element's CSS classes upon form submission.
Form Element	A built-in variable which returns an HTML element that was the target of a form submission.
Form ID	A built-in variable which returns **id** value of the submitted form.
Form Target	A built-in variable which returns a string contained in the **target** attribute value of the submitted form.
Form Text	A built-in variable which returns the visible text inside the submitted form.
Form Trigger	The Form Submit trigger waits for a **submit()** event to be dispatched. This means that if some script on your site hijacks the form event and proceeds with some proprietary AJAX function, for example, GTM's listener will not be able to pick it up. The trigger has two additional options: **Check Validation** and **Wait For Tags**. Once you check either, you'll see an extra step, **Enable When,** in the Trigger settings:  The point with **Enable When** is that you'll be able to specify a condition for **when the trigger is actively listening for the specified event**. The most common condition types you'll use here are page conditions since you might want to specify that the Form Submit trigger only listens for **submit** events on pages with forms.

Term	Definition
	Check Validation, when checked, will require that a valid action is propagated to GTM's listeners. With the Form Submit, it also checks for a prevented default action. For example, if validation fails, it's normal to prevent the default action of the submit event. **Wait For Tags** ensures that all tags that fire on the trigger execute first before proceeding with the action of the event. So if it's a Form trigger, the action is halted long enough for all dependent tags to complete execution, after which the action is resumed. Due to the diversity of forms and how they work, in many cases, GTM form trigger will not work properly. But that shouldn't be the problem because there are other 4 ways to track form submissions with Google Tag Manager.
Form URL	A built-in variable which returns the value of the **href** attribute of the submitted form, if any.
Google Analytics Settings Variable	This variable acts as a central location to configure sets of Google Analytics settings for use across multiple tags. Instead of having to enter your Google Analytics settings over and over again in each new Universal Analytics tag, you'll simply be able to select (or create) a Google Analytics Settings Variable to apply to multiple tags. Learn more about Google Analytics Settings Variable in GTM.

Term	Definition
	
Google Optimize	Google Optimize is Google's A/B testing and personalization platform. Like most A/B testing platforms, it allows marketers to test variations of a site in order to improve conversions. Unlike most A/B testing platforms, it natively integrates with Google Analytics. The Optimize tag may be deployed from any existing Tag Manager web container. However, for the best results, it is recommended to implement the JavaScript portion of your Tag Manager container snippet as high in the <head> of the page as possible, following the latest installation guide.

Term	Definition
	Google Tag Manager offers a ready-made Google Optimize tag template.
Google Surveys Website Satisfaction	Google Surveys (formerly Google Consumer Surveys) is a business product by Google that facilitates customized market research. The product was designed by Google as an alternative to internet pay walls for websites that publish content. Google Tag Manager offers a ready-made Google Surveys tag template, which requires you to enter Google Surveys Site ID.
Google Tag Manager	Google Tag Manager is a free tool that makes it easy for marketers to add and update website tags/scripts (including conversion tracking, site analytics, remarketing, and more) with just a few clicks, and without needing to edit your website code every time.
Google Tag Manager 360	Google Tag Manager 360 is a premium solution for Enterprise-level businesses. It is fully integrated with the Google Analytics 360 Suite and will be a platform to provide enterprise-focused features. Main differences between free and the 360 version are: ▪ 360 offers an unlimited number of workspaces ▪ An enterprise-level support (with dedicated specialists) for 360 users. ▪ 360 offers a service-level agreement
Google Tag Manager Extension	There are useful Google Tag Manager Extensions for Google Chrome will make your life/work much easier. Their main purpose is to help test GTM implementation and spot issues much faster. To name a few: Tag Assistant, DataLayer Inspector, Tag Manager Injector, etc. The full list can be found in my other blog post 12 Google Tag Manager Extensions for Chrome.
gtm.dom	A Data Layer event which is fired when the browser DOM is fully constructed. In Google Tag Manager Preview and Debug console, this event is displayed under the **DOM Ready** event name. It can be used as a trigger (**Page view > DOM Ready**).

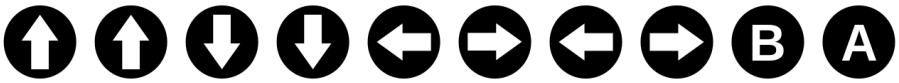
Term	Definition
	
gtm.js	A Data Layer event which is fired when the Google Tag Manager Container snippet is loaded. In Google Tag Manager Preview and Debug console, this event is displayed under the **Page View** event name. That event can be used as a trigger and it's the earliest moment (after the page starts loading) when you can fire tags via Google Tag Manager.
gtm.load	A Data Layer event which is fired when the Google Tag Manager Container snippet is loaded. In Google Tag Manager Preview and Debug console, this event is displayed as **Window Loaded**. It fires when everything else (including JavaScript) finishes loading on the page.
Hard-coded Scripts	Hard-coded scripts are little code snippets which developer added directly to the source code of the website/web application. Every time a change is required, a marketer needs to ask the developer to edit it. This process is really frustrating and time-consuming, that's where Google Tag Manager comes in handy. Instead of having all tracking codes/script hard-coded, you can control them from one place, Google Tag Manager interface.
Hard Refresh	Browser cache is useful for web browsing, but a real pain point for developers or sometimes those, who are working with web tracking. A **hard refresh** is a way of clearing the browser's cache for a specific page, to force it to load the most recent version of a page. Why is this important for Google Tag Manager users? Because sometimes Preview and Debug mode does not appear when it should. For Chrome users, click CTRL + F5 (for Windows users) or CMD + Shift + R (for Mac users). Here are the instructions for other browsers.

Term	Definition
	If your Preview and Debug mode still does not appear, check out this blog post how to fix it.
History Change	Triggers based on the History Change event will fire a tag when the URL fragment (hash) changes or when a site is using the HTML5 *pushstate* APIs. URL Fragment is a part of URL which comes after the hash (#) symbol, for example, https://www.analyticsmania.com/**#example_fragment.** So when the fragment changes, a History Change event can be fired to the Data Layer. This trigger is useful to fire tags tracking virtual Page view in an AJAX application, for instance.
History Source	A built-in variable which returns a string denoting the event that triggered the history change event (*popstate*, *pushState*, *replaceState*, or *polling*). Stored in Data Layer under the key **gtm.historyChangeSource**. The same variable can be created by the user with the following settings: ▪ Auto-Event Variable. ▪ Type: History Change Source.
HTTP Referrer	A built-in variable which provides the full referrer URL for the current page (the previously visited page which later redirected the user to the current page). With it, you can create triggers based on where the visitor is coming from. GTM users can also create the same variable by themselves by choosing **HTTP Referrer** variable type.
ID For Advertising	A built-in variable in containers for mobile apps. For Android, the value is set to the Advertising ID. More information is available at the Android Developer Center. SDK versions prior to v4 will always return '' (the empty string). For iOS versions 6 and greater, the value is set to the identifier for advertising (IDFA). Otherwise, the value is set to '' (an empty string).
Import Container	This is a feature in Google Tag Manager that lets you import a JSON file which contains tags, triggers, and variables. This may be an entire container or just items. Reasons to start using Import feature: ▪ You have a lot of sites and want to set up a default container for each of them. I have prepared a vast (and free) library of ready-made Google Tag Manager Container templates that you can easily import. ▪ Importing new tags, variables, and triggers that have been shared with you.

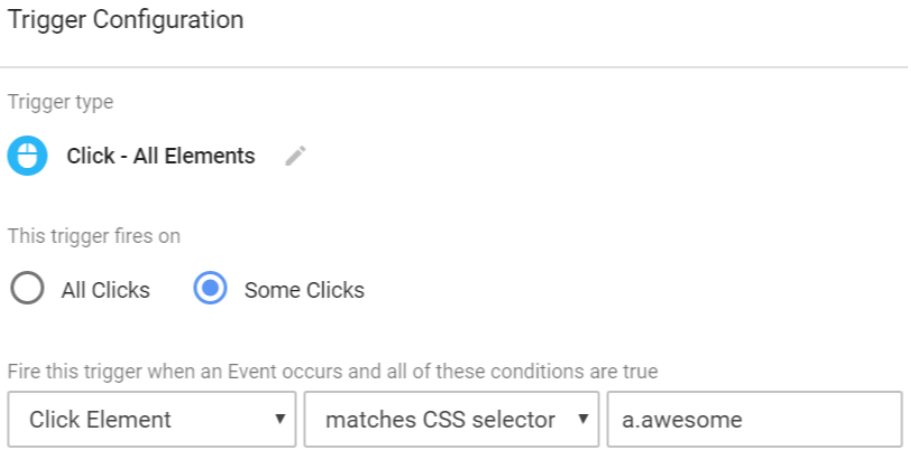
Term	Definition
	▪ Making bulk changes to your container. ▪ You're maintaining separate GTM containers in different environments, like a dev or staging site, and you want to export out of one environment and into another.
Import Container - Merge	When you start importing a GTM container, two options will be displayed: Overwrite or **Merge**. ▪ **Merging** containers will let you keep your existing tags, triggers, and variables, and just add in the new ones. If you choose to Merge the new container with your existing container, you'll have to then decide whether you want to overwrite conflicting tags or rename conflicting tags. ○ **Overwrite** – If a variable, tag, or trigger in the new container has the same name, overwrite the old one with the new one. ○ **Rename** – If a variable, tag, or trigger in the new container has the same name, keep the old one and rename the new one.
Import Container	When you start importing a GTM container, two options will be displayed: **Overwrite** or Merge. **Overwriting** the existing container will remove all your existing tags, triggers, and variables, and will replace them with those in the imported container. A new container version will be created before the import. Use this option if you want to start fresh with the imported container.
JavaScript Error	A default Google Tag Manager trigger which fires when the JavaScript Error occurs on the page. There are 3 built-in variables that are useful while working with JavaScript errors: ▪ Error message ▪ Error URL ▪ Error line In Google Tag Manager Preview and Debug mode, these events are displayed as **gtm.pageError**.
JavaScript Variable	The JavaScript Variable returns the value stored in the global JavaScript variable you specify. Note, this is NOT the same as the Custom JavaScript Variable, which is a function declaration. Most common use case: Page Title variable. By default, Google Tag Manager does not offer Page Title as a built-in variable. If this

Term	Definition
	information is important to you, create a Javascript variable with **document.title** as its value.

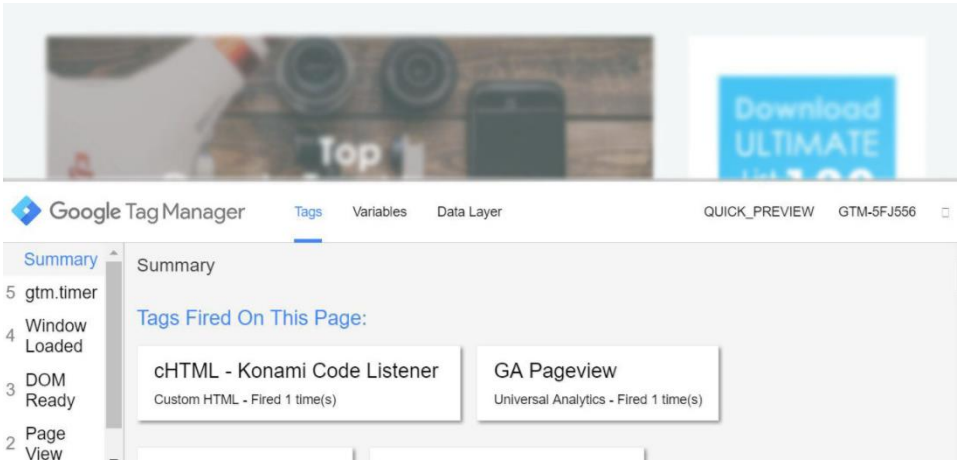
JSON	JSON is short for **JavaScript Object Notation** and is a way to store information in an organized, easy-to-access manner. In a nutshell, it gives us a human-readable collection of data that we can access in a really logical manner.
	
	As a simple example, information about a person might be written in JSON as follows:
	
	`var person = {`
	`"age" : "24"`
	`"gender" : "male"`
	`};`
	
	Google Tag Manager exports and imports containers in JSON.

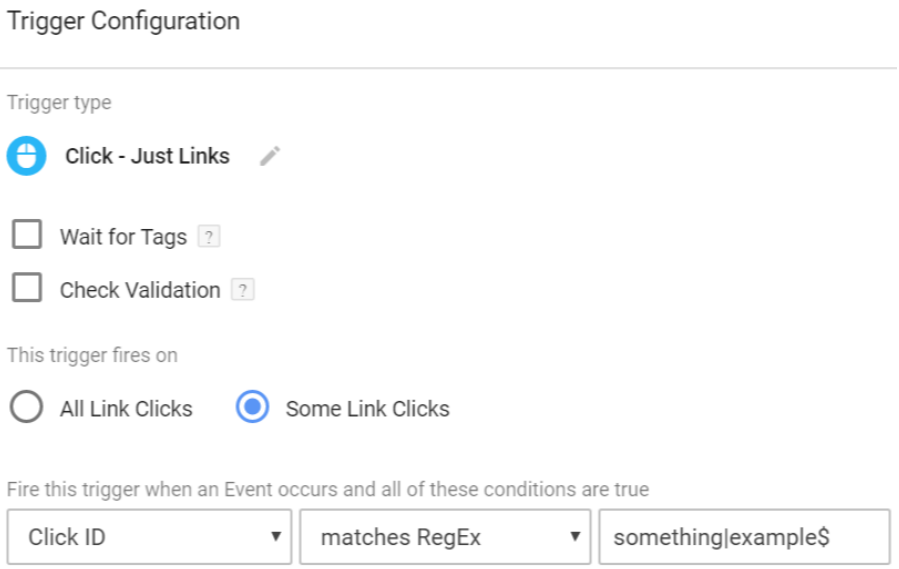
Konami Code	The Konami Code is a cheat code that appears in many Konami video games, although the code also appears in some non-Konami games as well. Years later, it became an internet meme and was adopted by multiple website creators. Various web developers have hidden an Easter egg in their projects that can be accessed by entering the following sequence of key presses on the keyboard:  In this blog post, I have demonstrated 4 examples how to implement Konami code on your website with the help of Google Tag Manager.
Language	A built-in variable in containers for mobile apps. The value is set to the two letter language code representing the user-set device language.
Listener	In JavaScript, a listener is a function which listens to specific interactions on the web page, and, when one occurs, runs a particular code (e.g. sends data to the Data Layer). See also **Auto-Event Listener**.
Lookup Table	Lookup table is a variable that has the value of another variable as the input. It works like this: **When [input variable] equals to _____, set [this output variable] to_____.**

Term	Definition
	 It can drastically reduce the number of tags required and turn your messy GTM into a neat environment. The most popular use case from my experience: **Different Google Analytics Tracking IDs** – one for production (live) website, the other one for the development version. With Lookup Table all page views from "development" website will be sent to my GA test account and will not mess my Live tracking data: ▪ If **Page Hostname** equals to *https://www.analyticsmania.com,* set **Lookup Table's** value to UA-11111111. ▪ If **Page Hostname** equals to *https://dev.analyticsmania.com*, set **Lookup Table's** value to UA-22222222.
Malware Detection	The term malware, short for "malicious software," refers to any software specifically designed to harm a computer or the software it's running. Malware can steal sensitive information (like credit card numbers or passwords) or even send fake emails from a user's email account, often without the user's knowledge. To protect the safety and security of our users, Google Tag Manager will stop firing tags pointing to sites where we find malware. You can learn more about Malware detection here.
Matches CSS Selector	CSS Selectors are patterns used to select the element(s) you want to style. In Google Tag Manager, **Matches CSS Selector** option empowers you to target website elements by their features.

Term	Definition
	 For example, you can track clicks of only those buttons which have a CSS class **.button** and are descendants of a block with class **.main-div**. This is especially useful when a single page has multiple buttons with the same style, links, and text, and you need to track them separately.
Naming Conventions	In computer programming, a naming convention is a set of rules for choosing the character sequence to be used for identifiers which denote variables, types, functions, and other entities in source code and documentation. In other words, it's a logical way of naming items (e.g. files and folders) to keep everything in order. In Google Tag Manager, naming conventions are important when naming tags, triggers, and variables. Instead of naming a tag as "Button click", try giving it a more descriptive name **GA Event - Click - Contact Support Button**. Lunametrics have posted a neat blog post about Naming Conventions.
New History Fragment	A built-in variable which returns the new URL fragment set with a browser history event. Stored in Data Layer under the key *gtm.newUrlFragment*. URL fragment is the part of web address after the # symbol (e.g. www.example.com/#contact-us). This is especially popular among one-page sites where URL fragment changes when you scroll.
Old History Fragment	A built-in variable which returns the old URL fragment replaced in the browser history event. Stored in Data Layer under the key *gtm.oldUrlFragment*.
Old History State	A built-in variable which returns the old state object replaced in the browser history event. Stored in Data Layer under the key *gtm.oldHistoryState*.

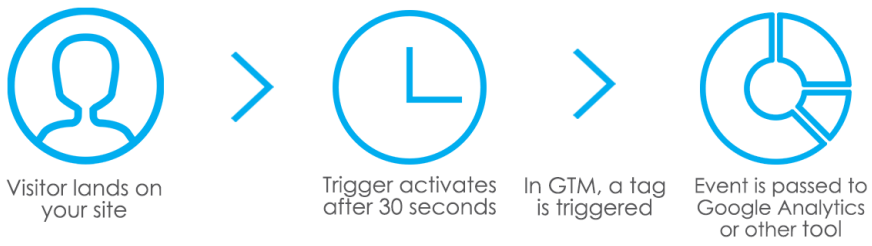
Term	Definition
Only Fire This Tag In Published Containers	It's an option located in Advanced settings of the tag. If firing certain tag results in a fee being charged, you can avoid firing it in testing environments (e.g., during previewing and debugging). After you enable this option, Google Tag Manager will not fire the tag in Preview and Debug mode. For the most accurate testing, leave this option unchecked.
Operating System Version	A built-in variable in containers for mobile apps. The value is set to the version of the operating system in which the application is installed.
Overview	A section of the Google Tag Manager User Interface which displays the container version number, a list of workspace changes, version description, and activity history.
Page Hostname	A built-in variable which returns a string containing the hostname of the current page (e.g. www.analyticsmania.com). The same result can be achieved by creating a user-defined variable with the following settings: ▪ Variable type – URL. ▪ Component type - Host name.
Page Path	A built-in variable which returns a string containing the relative path of the current page. If the page's address is https://www.analyticsmania.com/post/example?gclid=4482923423492342324, Page Path's value will be equal to /post/example. The same result can be achieved by creating a user-defined variable with the following settings: ▪ Variable type - URL. ▪ Component type - Page Path.
Page URL	A built-in variable which returns a string containing the full URL of the current page, e.g. **https://www.analyticsmania.com/post/example?gclid=4482923423492342324.** The same result can be achieved by creating a user-defined variable with the following settings: ▪ Variable type - URL. ▪ Component type - Full URL.
Page View	This event corresponds to the rendering of a page in the user's web browser. It is the earliest moment when your tags can be fired via Google Tag Manager. **Page view** is also known as *gtm.js* Data Layer event.

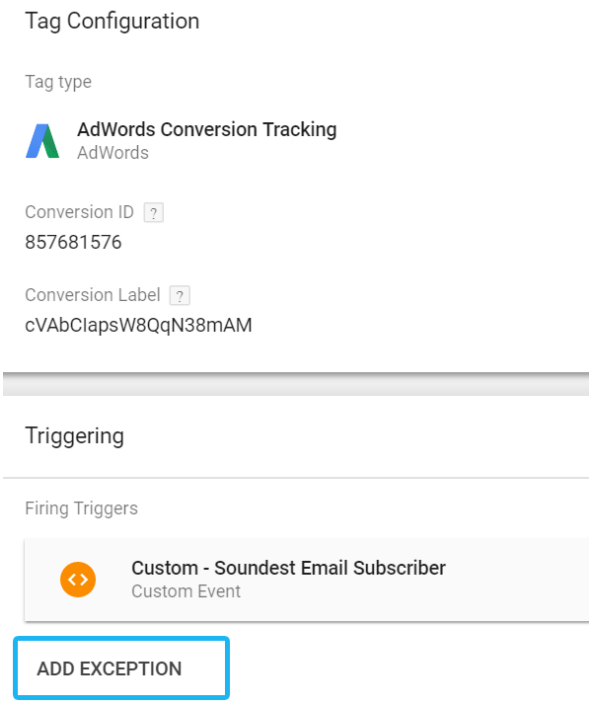
Term	Definition
Permissions	Google Tag Manager allows you to delegate access to other users at the Account and Container level. Users can be granted the ability to view or administer other users at the Account As for permissions, users can be granted the ability to view or administer other users at the Account level and can be granted read, edit, approve, or publish rights at the Container level.
Platform	A built-in variable available in containers for mobile apps. The value is set to the platform of the currently running application (e.g., "Android").
Preview and Debug Mode	Google Tag Manager Preview and Debug (P&D) mode enables you to browse a site on which your GTM container code is implemented. Sites with preview mode enabled will display a debugger pane (a.k.a. console) at the bottom of your browser screen so that you can inspect which tags fired and in what order. And the best part: Preview and Debug console is visible only to you, not your website visitors. 
Random Number	A built-in variable which returns a number, randomized between 0 and 2147483647. Use case: sample your visitors. If you want to fire a tag for every 10th visitor, you can set the rule **{{Random Number}}** *ends with 1*.
Recipe	A ready-made Google Tag Manager container template that can be imported to GTM container. Go to the library of Google Tag Manager recipes, download any recipe you need, and follow the instructions.
Regular Expression	Regular expressions (also known as regex) are used to find specific patterns in a list/text. Instead of having multiple “contains” conditions, you can match different data with just one Regular Expression. Here are few examples: ▪ A|B means ***A or B.*** ▪ a$|^b means ***ends with A or starts with B***

Term	Definition
	In Google Tag Manager, Regular Expressions are useful when it comes to triggers. 
Release Notes	Release notes is a page, where Google Tag Manager team announces new features. Although these notes have some delay (compared to Google Analytics blog), this is still a pretty reliable source of news. Another way to stay up to date is to join GTM Google+ community.
Revoke Authorization Code	This term is related to the **Share Preview** feature of GTM. If you have preview links sent out and you'd like to invalidate that preview, you can do so from the Actions menu: ▪ In the **Environments** listing (*Admin* > *Container* > *Environments*), find the entry for the environment you would like to install. ▪ Click the **Actions** menu. ▪ Select **Reset Link.** A confirmation box will appear. Click **Reset link** to cancel the old code and create a new one.
Screen Resolution	A built-in variable available in containers for mobile apps. The value is set to the screen resolution of the device of the currently running application. The format is '<width>x<height>', e.g. '1024x768'.
SDK Version	A built-in variable available in containers for mobile apps. The value is set to the SDK (Software development kit) version of the operating system in which the application is installed.

Term	Definition
Setup Tag	Opposite to the **Cleanup Tag**. In a tag, under *Advanced Settings*, you'll find a **Tag Sequencing** section which contains two options: ▪ ***Fire a tag before XXXX fires*** option is called a **setup tag**. ▪ ***Fire a tag after XXXX fires*** option is called a **cleanup tag**. The Setup Tag which is a tag which must fire before the main Tag fires.
Share Preview	When you enable preview mode, the previewed container configuration and debugger pane is only visible from the same browser from where you activated preview mode from. However, you may share your preview mode with others using the share preview feature. To share your preview mode, click Share Preview from the preview notification banner. This brings up a dialog box where you can copy a URL to send to someone else. Enter the website domain (such as https://example.com/) and copy the resulting preview URL in the box below. You can then send the preview URL to another person.
Supported Tags	Google Tag Manager features a powerful tag template system to help simplify publishing of tracking codes and eliminate errors. In addition to the templates for Google tags such as Analytics, AdWords, and DoubleClick, templates for a growing list of certified vendors are also supported. You can find the full list here. You can also submit your own Tag Template to become an officially supported Vendor.
Synchronous Loading	Unlike asynchronous, synchronous scripts must load one after another. They might block page to render and do not allow the browser to continue loading elements until X or Y script is loaded. Google Tag Manager, Google Analytics and the majority of other tracking scripts are loaded asynchronously. So if you have a synchronous script to put, add it directly to the website's source code. Currently, GTM does not support synchronous scripts.
Tag	A tag is a piece of code that sends information to a third party, such as Google Analytics. Tags are often provided by analytics, marketing, and support providers to help you integrate their products into your website or mobile app. With Google Tag Manager, you no longer need to maintain each of these JavaScript snippets in your source code. Instead, you specify the tags that you want to fire, and when you want them to fire, from within the Google Tag Manager user interface.
Tag Assistant	The Tag Assistant Chrome extension is a very useful tool in testing and debugging your implementation. Not only does it help with the

Term	Definition
	troubleshooting of Google Tag Manager, but also Google Adwords, Analytics, and DoubleClick implementation can be easily verified. One of the most powerful features within the tool is recordings. This allows you to “record” a browsing session and produce a report on all of the hits being sent through and even how they will potentially show up within Google Analytics.
Tag Firing Options	You can control how a tag fires using Tag Firing Options, which is found in the Advanced Settings of any tag definition in a web container. You may select from the following settings: ▪ **Unlimited**: Tags will fire whenever triggers tell it to. (Used only with Tag Sequencing) ▪ **Once per event**: The tag will fire only once when the specified event occurs. This might be used when adding information to the data layer. There is no need to add multiple pieces of the same data to the data layer, so the tag is only fired once. ▪ **Once per page**: The tag will fire only once per page load. This might be used if you were loading an external JavaScript library, which would only need to happen once when the page loads (e.g. Facebook Pixel).
Tag Firing Priority	Tag Firing Priority determines the order in which tags will be fired. Tag Firing Priority can be a positive or negative integer, and tags with higher priorities will be fired first. (For example, a priority 3 tag will fire before priority 1 or 2 tags). Priority defaults to 0 if none is specified. Tags with higher numbers for priority will be fired first. Priority defaults to 0 if none is specified. Tags will still be fired asynchronously (tags will fire whether or not the previous tag has finished).
Tag Manager Injector	Tag Manager Injector is a Google Chrome Extension which simplifies the process of inserting GTM code on any site on the fly where no Tag Manager container code currently exists. Injector enables you to imitate as if your GTM container code was added to any website. But everything happens only within the boundaries of your browser window. So you can safely play with tag manager without any consequences. Don’t worry! You’re not hacking anyone’s website.
Tag Sequencing	A cleanup tag is a part of the which comprises three phases: ▪ **The setup**, it is a Tag which must successfully complete before the main Tag fires. You can set it under *Advanced Settings* of the tag.

Term	Definition
	▪ **The main tag**, which is the Tag with the dependencies. It's either dependent on the setup, or it establishes a dependency with the cleanup. ▪ **The cleanup**, which is a Tag which fires after the main tag has completed successfully. Simo Ahava has a detailed explanation of how the tag sequencing works.
Tag Template	Google Tag Manager features a powerful tag template system to help simplify publishing of tracking codes and eliminate errors. Instead of writing your own code, you can use tag templates in Google Tag Manager which offer a bunch of predefined fields with settings. This approach is much more user-friendly. Several examples: Universal Analytics Tag Template, Google Optimize, CrazyEgg, etc.
Timer	Timer Trigger is a type of trigger which fires **after a certain duration of time has passed on the web page**.  The most common uses for the timer trigger seem to be either to send an event to Google Analytics after X seconds (to lower the Bounce rate) or to defer a tag from firing until some asynchronous request has completed with certainty. But don't limit yourself!
Trash Can	The Trash Can is a temporary holding area for accounts and containers before they are permanently deleted. Accounts and Containers are moved to the Trash Can and held for 30 days before they are permanently deleted.
Trigger	A trigger is a condition that evaluates to either true or false at runtime. Triggers attached to a tag govern when the tag should fire or not. All tag firing in Tag Manager is event-driven. Anytime an event is registered by Tag Manager, triggers from the container are evaluated and tags are fired accordingly. No tag can be fired unless an event occurs.

Term	Definition
	An event can be a page view, a click on a button, a form submission, or any custom event that you define. Tag Manager has 6 built-in event types plus a custom event option: ▪ Page view ▪ Click ▪ Form submission ▪ Javascript Error ▪ History Change ▪ Timer ▪ Custom event (which is created with dataLayer.push)
Trigger Exception	Trigger Exceptions are special Triggers that you can use to block Tags from firing in certain instances. If you want some Tags to only fire on a Live site, then you can create a Trigger Exception, or a blocking Trigger, that says – "Do NOT fire if you are NOT on the Live site." You can set the exception in Tag's Triggering Section (see the image below). 
Universal Analytics	Universal Analytics is the new version of Google Analytics. All newly created Web Properties in Google Analytics are Universal Analytics properties, so you don't need to do any additional configuration.

Term	Definition
	Here are key differences between Universal Analytics and Classic Analytics (the old GA version): ▪ **User ID.** This feature shows the engagement activity of a user across different devices and visits to your site to provide a more user-centric view of your traffic and help you build a more tailored experience for your customers as well. ▪ **The tracking code** is different. ▪ **Custom Data.** Custom variables have become custom dimensions, and the limit was increased to 20 (for the free plan) and 200 (for Premium users). ▪ **The Measurement Protocol** allows you to send data to Google Analytics from anything that has access to the internet – think point-of-sale (POS) systems, kiosks, call centers, etc.
URL Variable	The URL Variable lets you access components of the current page URL. This is a very versatile Variable type and is especially useful for traversing query parameters and hash fragments in your URLs. Learn more about it. Let's say the address of web page that I am currently on is *https://www.example.com/welcome:8080?gclid=aabbcc123#home*. With help of URL Variable, you can turn a particular fraction of the URL into a variable. Here's a list of available component types which can be extracted: ▪ **Full URL**, e.g. **https://www.example.com/welcome?gclid=aabbcc123**. ▪ **Protocol**, e.g. **https**. ▪ **Host Name**, e.g. **www.example.com**. ▪ **Port**, e.g. **8080**. ▪ **Path,** e.g. **/welcome/**. ▪ **Query.** ▪ **Fragment.**
URL Variable Component – Fragment	URL Variable's Component type which returns the value of the URL's fragment without the leading '#'. Let's say the address of web page that I am currently on is *https://www.example.com/welcome:8080?gclid=aabbcc123#home*. In this case, variable's value would be **home**.

Term	Definition
URL Variable Component – Full URL	URL Variable's Component type which returns the full URL without the hash fragment. Let's say the address of web page that I am currently on is *https://www.example.com/welcome:8080?gclid=aabbcc123#home*. In this case, variable's value would be **https://www.example.com/welcome:8080?gclid=aabbcc123**.
URL Variable Component – Hostname	URL Variable's Component type which returns hostname of the URL without the port number. Let's say the address of web page that I am currently on is *https://www.example.com/welcome:8080?gclid=aabbcc123#home*. In this case, variable's value would be **www.example.com.** You can choose to strip 'www.'
URL Variable Component – Path	URL Variable's Component type which returns only the path name in the URL (in other words – everything after host name, except port, fragment, and query parameters). Let's say the address of web page that I am currently on is *https://www.example.com/welcome:8080?gclid=aabbcc123#home*. In this case, variable's value would be **/welcome/.**
URL Variable Component – Port	URL Variable's Component type which returns the port number used in the URL. Let's say the address of web page that I am currently on is *https://www.example.com/welcome:8080?gclid=aabbcc123#home*. In this case, variable's value would be **8080.** Or 80 for HTTP / 443 for HTTPS, if the URL had no port number.
URL Variable Component – Query	URL Variable's Component type which returns the entire query parameter string (without '?'), if you don't specify a query key. Let's say the address of the web page that I am currently on is *https://www.example.com/welcome:8080?gclid=aabbcc123#home*. In this case, I could specify query key to **gclid,** thus URL variable's value would be **aabbcc123**. If you do specify a query key, but that key is absent, variable's value would be *undefined.*
Variable	Variables are name-value pairs for which the value is populated during runtime. For example, the predefined variable named *url* has been defined such that its value is the current page URL. Variables are used in triggers and in tags. In triggers, they are used to define filters that specify when a particular should be executed (e.g. to execute a pageview trigger when the url variable is *example.com/index.html*).

Term	Definition
	In tags, variables are used to capture dynamic values (e.g. passing the transaction value and products purchased to a conversion tracking tag).
Vendor Tag Template Program	The Google Tag Manager Tag Vendor Program allows tag providers to natively integrate with Google Tag Manager's tag templating system. Benefits: ▪ Google Tag Manager users will be able to add a tag easily. ▪ Google Tag Manager users will be exposed to a brand.
Version	A container version can be thought of as an immutable copy (snapshot) of a container's configuration at a particular point in time. You can save the current state of a workspace as a version at any time. This allows you to save and preserve your work and go back to a previous version if you need to.Every time a container is published, a version of that container is recorded. Versioning can make it easier to recover from mistakes. For example, if someone accidentally publishes a container version before it's ready for production, you can simply publish another version that has been debugged and is known to work.
Window Loaded	Window Loaded is a Pageview-type trigger which occurs when everything else (including Javascript) finishes loading on the page (and *gtm.load* Data Layer event is fired). You can see Window Loaded event in the GTM Preview and Debug console (event list).
Workspace	In Google Tag Manager, workspaces enable you to create multiple and differing sets of changes to your container. Different users and teams can work on these sets of changes in separate workspaces to independently develop and test tag configurations.
Workspace Conflicts	If there any conflicts between the changes synced to your workspace while updating and any of the changes already in your workspace, you will see a “Conflict found” indication on the Workspace Overview page. Selecting to resolve the conflicts will bring you into a conflict resolution tool. For each entity in your workspace where a conflict has been found (i.e. each tag, trigger, variable, etc.), you will be able to resolve conflicts one field at a time. The configuration of the entity in the latest synced version is shown on the left, and the configuration of the entity in the current workspace is shown on the right. Each conflict is highlighted, and the color of the highlighting indicates the nature of the conflict:

Term	Definition
	▪ **Blue**: Something has been modified between the latest synced version and the current workspace. ▪ **Red**: Something from the latest synced version is not present in the current workspace. ▪ **Green**: Something from the current workspace is not present in the latest synced version.

This guide was compiled by Julius Fedorovicius, analyticsmania.com
Contact the author via Twitter or Linked In