# [Google Analytics – GA](https://developers.google.com/analytics/)

<https://moz.com/blog/absolute-beginners-guide-to-google-analytics>

## Setup GA

<https://developers.google.com/analytics/devguides/collection/analyticsjs/>

### Step 1: Find your tracking ID and tracking-code snippet [link](https://support.google.com/analytics/answer/1008080#trackingID)

When you [create an Analytics property](https://support.google.com/analytics/answer/1042508), Analytics **generates** a tracking ID and a JavaScript tracking-code snippet specific to that property.

### Step 2: Adding analytics.js to Your Site

### Async tracking snippet - support for preloading

Degrade to synchronous loading and execution on IE 9 and older mobile browsers that do not recognize the async script attribute. Only use this tracking snippet if your visitors primarily use modern browsers to access your site.

<!-- Google Analytics -->  
<script>  
window.ga=window.ga||function(){(ga.q=ga.q||[]).push(arguments)};ga.l=+new Date;  
ga('create', 'UA-XXXXX-Y', 'auto');  
ga('send', 'pageview');  
</script>  
<script async src='https://www.google-analytics.com/analytics.js'></script>  
<!-- End Google Analytics -->

### Not allowing modern browsers to preload the script

Added near the top of the <head> tag and **before** any other script or CSS tags, and the string 'UA-XXXXX-Y' should be replaced with the [property ID](https://support.google.com/analytics/answer/1032385) (tracking ID) of the Google Analytics property you wish to track.

<!-- Google Analytics -->  
<script>  
(function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){  
(i[r].q=i[r].q||[]).push(arguments)},i[r].l=1\*new Date();a=s.createElement(o),  
m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)  
})(window,document,'script','https://www.google-analytics.com/analytics.js','ga');  
  
ga('create', 'UA-XXXXX-Y', 'auto');  
ga('send', 'pageview');  
</script>  
<!-- End Google Analytics -->

The above code does four main things:

1. Creates a <script> element that starts asynchronously downloading the analytics.js JavaScript library from https://www.google-analytics.com/analytics.js
2. Initializes a global ga function (called [the ga() command queue](https://developers.google.com/analytics/devguides/collection/analyticsjs/how-analyticsjs-works)) that allows you to schedule commands to be run once the analytics.js library is loaded and ready to go.
3. Adds a command to the ga() command queue to [create a new tracker object](https://developers.google.com/analytics/devguides/collection/analyticsjs/creating-trackers) for the property specified via the 'UA-XXXXX-Y' parameter.
4. Adds another command to the ga() command queue to [send a pageview to Google Analytics](https://developers.google.com/analytics/devguides/collection/analyticsjs/sending-hits) for the current page.

#### Check your web-tracking-code setup - Real-Time:

<https://support.google.com/analytics/answer/1008083>

## Track type

1. Pageview for each page your users visit.
   1. The total time a user spends on your site.
   2. The time a user spends on each page and in what order those pages were visited.
   3. What internal links were clicked (based on the URL of the next pageview).
2. The IP address, user agent string, and initial page inspection analytics.js does when creating a new tracker is used to determine things like the following:
   1. The geographic location of the user.
   2. What browser and operating system are being used.
   3. Screen size and whether Flash or Java is installed.
   4. The referring site.
3. Events are user interactions with content that can be tracked independently from a web page or a screen load. Downloads, mobile ad clicks, gadgets, Flash elements, AJAX embedded elements, and video plays are all examples of actions you might want to track as Events.

## Custom dimensions & metrics

<https://support.google.com/analytics/answer/2709828?hl=en>

<https://support.google.com/analytics/answer/6164990?hl=en>

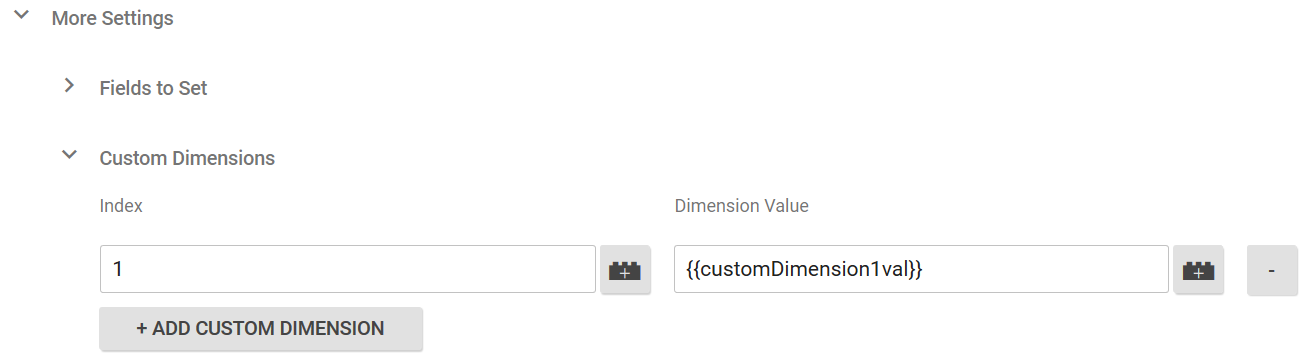
<https://yoast.com/google-analytics-custom-dimensions/>

Custom dimensions are a way for you to track extra things. And as the name implies; they’re custom. This means that you can actually specify yourself what you want Google Analytics to track.

Custom variables had a maximum of 5.

Custom dimensions, variables you’re allowed to have is 20. Custom dimensions and custom variables are essentially the same.

Once you’ve created a new dimension, you should be given an example code. This is where you should check the dimension number - in the below example you can see that it is ‘dimension1.’ Remember this, as it is the same dimension number you will want to use in your Custom Dimension settings in Google Tag Manager for the ‘Index’ input later.



# Google Tag Manager – GTM

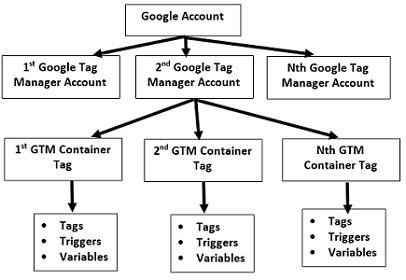
<http://www.lunametrics.com/blog/2016/02/15/what-is-google-tag-manager/>

Tag Manager is **NOT** the same thing as (or the latest version of) Google Analytics. But a completely separate tool – tag management system.

With Google Tag Manager, you NO 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 GTM user interface.

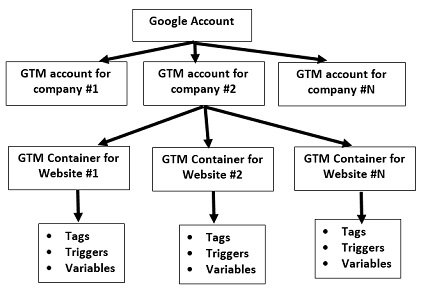
## Google Tag Manager Account Structure

<https://www.optimizesmart.com/google-tag-manager-implementation-guide/>



The rule of thumb is to create one container tag for each website. An account contains one or more **containers**, one for each type of property you own: web, iOS app, Android app, or AMP.

The tags, triggers and variables are collectively called **GTM configurations**. You can’t share GTM configurations across containers.



## Setup GTM

<https://developers.google.com/tag-manager/quickstart>

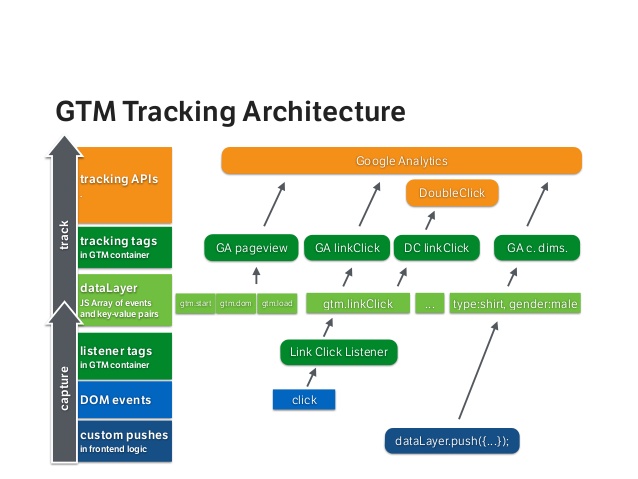
1. Create GTM account
2. Create container
3. Copy & paste container code
4. Add, update and publish tags.

### What Does This Code Do?

<http://www.lunametrics.com/blog/2016/11/22/google-tag-manager-snippet-placement/#what-does-this-code-do>

## How it works

<https://www.slideshare.net/liip/google-tag-manager-an-introduction?from_action=save>

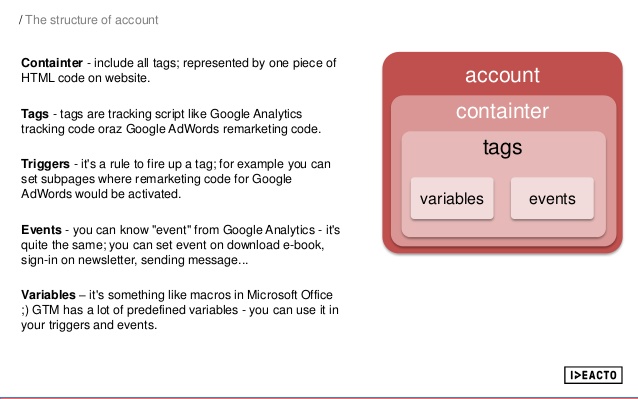


## GTM Tracking Process

1. Browser loads page with GTM <script> tag
2. GTM tag interpreted
   1. creates hidden iframe for browsers that doesn’t allow scripts OR
   2. loads gtm.js:
      1. your speciﬁc container (tags, triggers, variables)
      2. generic GTM logic
3. GTM ﬁres up:
   1. if dataLayer does not exist, it creates it
   2. all tags/triggers/variables are instantiated and now listening
   3. gtm events thrown
4. When a DOM event happens …
   1. Triggers deﬁne which tracker tags gets ﬁred and throws events on dataLayer
5. When any event or data gets pushed on the dataLayer …
   1. Triggers deﬁne which tracker performs

## Tags, Triggers, Variables and Data layer

<https://www.wholewhale.com/tips/google-tag-manager-v2-3-new-updates/>

[](https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=0ahUKEwin6sK97_jTAhVJ4mMKHdv3DCAQjRwIBw&url=https://www.slideshare.net/ideacto/slide-share-google-tag-manager&psig=AFQjCNEbfCeDEeiNx8zGXUaN8ZfkdUp6ig&ust=1495019275582253&cad=rja)

### Tags

<https://support.google.com/tagmanager/answer/2574372?hl=en&topic=2574304&ctx=topic>

A tag is snippet of JavaScript that sends information to a third party, such as Google. Tags are used to collect and send your data to any service you use.

* [templates for supported Google and DoubleClick tags](https://support.google.com/tagmanager/answer/2574372?hl=en&topic=2574304&ctx=topic#GoogleTags)
* [templates for certified vendor tags](https://support.google.com/tagmanager/answer/2574372?hl=en&topic=2574304&ctx=topic#nonGoogleTags)
* [custom tag types](https://support.google.com/tagmanager/answer/2574372?hl=en&topic=2574304&ctx=topic#CustomTags) for all other tags

When we speak of analytics/tracking/marketing tags, we are referring to the **block of code** that the **provider has given** us to install their tracking code or functionality onto our website or mobile app.

<https://support.google.com/tagmanager/answer/3281060>

### Triggers

Triggers are used to make sure your tags fire when you want them to. Every trigger now has to be associated with a particular event. These events are:

* Page View – Fires upon pageview
* Click – Fires when the user clicks a particular event (This trigger type has replaced Auto-Event Listener Tags)
* Form – Fires when a user submits a form
* History Change – Fires when the URL changes, useful for tracking virtual pageviews
* Javascript Error –  Fires when the browser script finds an uncaught error
* Timer – Fires on a set interval of time

Once you select an event you have to add filters; these filters essentially give additional instructions to your triggers such as how often to fire, what clicks to fire on, or what forms to fire on.

### Variables

Most used: Data Layer Variable, Constant.

Built-in Variables: <https://www.simoahava.com/analytics/variable-guide-google-tag-manager/>

**Auto-Event Variables** are used to access the target element of an [auto-event action](http://www.simoahava.com/analytics/auto-event-tracking-gtm-2-0/) (e.g. Click, Error, Form Submit). When you create a new Auto-Event Variable, you need to specify just which component of the target element you want to access.

<https://www.simoahava.com/analytics/custom-event-listeners-gtm/>

Trigger/Custom Event/Event Name:

* *beforeunload* – Fire a listener when the window, the document, and all resources are about to be unloaded (e.g. when someone is closing the browser window).
* *blur* – An element has lost focus (e.g. the user has left a form field). Note, this doesn’t bubble by default, meaning a listener on the document node won’t be able to catch it. To activate event delegation, you’ll need to set the last parameter in the document.addEventListener() call to **true** instead of **false**.
* *change* – The value of an element changes between receiving and losing focus (e.g. the user enters a form field, types something in, and leaves the field).
* *click* – A click is registered on an element (use GTM’s Click Listener instead).
* *contextmenu* – The right mouse button is clicked.
* *copy* – Text is copied to the clipboard.
* *cut* – Text is cut to the clipboard.
* *dblclick* – A double-click is registered on an element.
* *focus* – An element has received focus (e.g. the user has left a form field). Note, this doesn’t bubble by default, meaning a listener on the document node won’t be able to catch it. To activate event delegation, you’ll need to set the last parameter in the document.addEventListener() call to **true** instead of **false**.
* *keydown* – A key is pressed down.
* *keyup* – A pressed down key is released.
* *mousedown* – The mouse button is pressed down.
* *mouseenter* – The mouse pointer is moved over the element where the listener is attached. Won’t really work if the listener is on the document node.
* *mouseleave* – The mouse pointer is moved off the element where the listener is attached. Won’t really work if the listener is on the document node.
* *mouseout* – The mouse pointer is moved off the element where the listener is attached or one of its children.
* *mouseover* – The mouse pointer is moved over the element where the listener is attached or one of its children.
* *mouseup* – The pressed down mouse button is released.
* *orientationchange* – The orientation (portrait / landscape) of the screen changes.
* *reset* – A form is reset.
* *scroll* – A document view or element is scrolled.
* *submit* – A form submit is registered (use GTM’s Form Submit Listener instead).

Existing code: Custom Event Dimensisions VPV: (all event && event.contain(vpv)), vpv event Pushed to the data layer when click links DDS 6 (structure Emberjs) ui\sites\cc-classic\client\app\router.js

### Custom Variable of GA

Used in custom reports and advanced segments.

<http://cutroni.com/blog/2011/05/18/mastering-google-analytics-custom-variables/>

There are 4 parts to a custom variable:

1. The name of the variable  
2. The values for each variable  
3. The index or slot of the variable  
4. The scope of the variable

**Index or Slot**

The index is a way to organize your custom variables. Index is also referred to as “slot”. At a basic level, you get 5 custom variables. But this can actually be misleading. Think of the index as a parking lot with 5 parking space. You can put or “park” a custom variable in each space.

You can technically have more than 5 custom variables.

**Scope:** three different scopes

Determine **how long the custom variable will persist**. How long a car will stay parked in the custom variable parking lot.

### Data layer

<https://developers.google.com/tag-manager/devguide#datalayer>

An object that contains all of the information that you want to pass to Google Tag Manager.

Setup data layer:

<script>  
  dataLayer = [];  
</script>

<!-- Google Tag Manager -->  
...  
<!-- End Google Tag Manager -->

#### Default data layer events for web

Google Tag Manager pushes a certain set of values to the data layer of web applications by default. These values are:

* gtm.js – Pushed to the data layer as soon as Google Tag Manager is ready to run.
* gtm.dom – Pushed to the data layer when the DOM is ready.
* gtm.load – Pushed to the data layer when the window is fully loaded.

[

{

"gtm.start": 1495597614202,

"event": "gtm.js",

"gtm.uniqueEventId": 0

},

{

"event": "gtm.dom",

"gtm.uniqueEventId": 3

},

{

"event": "gtm.load",

"gtm.uniqueEventId": 4

}

]

## GTM with cookie

<https://stackoverflow.com/questions/37052310/disable-google-tag-manager-according-to-the-decision-of-single-users-opt-out>

Just to point out, GTM is neither a tracking tool nor does it by itself set any cookies.

More programming related, the GTM code cannot disable itself based on a cookie because GTM needs to be loaded to check if a cookie exists.

Cookies are sent as part of the http request only to the domain that has set the server; GTM resides on a Google server that will not have access to the cookies set on your domain. So if an opt-out cookie is set on your domain the GTM server will not know about it.

Cookies are mostly a client-side technology; GTM interacts with cookies by injecting JavaScript into your page so that it runs in the context of your domain, and then have the script evaluate the contents of your cookie (if you set up a cookie variable or a custom script). At that point the GTM code is already loaded.

That is why you cannot use a cookie to prevent GTM from loading; however you can use a cookie to disable all tags within GTM. If that's not good enough you have to write your own logic to disable GTM conditionally (you could even write a routine in your CMS that doesn't render GTM code based on a cookie - after all that's your own domain, so cookie data is sent along the request; you just cannot expect Google to do this for you).

GTM cannot set the cookie by itself (unless you write a custom HTML tag with a script that sets cookies, which is not different from doing it via inline code), so I will assume for an example that you already a cookie called "opt-out". It does not matter what value is stored in that cookie, we will just check if it is there.

Go to GTM, to the "variables" section, click "new" and select "First Party Cookie". Name it e.g. "Opt Out Cookie" and set the name field to "opt-out". Save. Now you have a variable that checks for the opt-out cookie, returns a value if it is set and returns "undefined" if the cookie is not there.

Now go to the triggers section and create a new trigger of the type page view. Call it e.g. "Opt Out Trigger". In the "fire on" section you select the "Opt Out Cookie" variable in the first field, set "does not equal" as condition and "undefined" as value (so the trigger evaluates true when the cookie is set).

Now go through your tags and add the "Opt Out Trigger" to the tags you want to disable when the cookie is set. Save, publish.

The only caveat is that a pageview trigger might either fire on pageview, DOM ready or pageload. An exception trigger that fires on pageview will not prevent tags from firing that are set to DOM ready or pageload, so you might need multiple exceptions, one for each stage of the loading process.

# Google Analytics via Google Tag Manager

## How it works





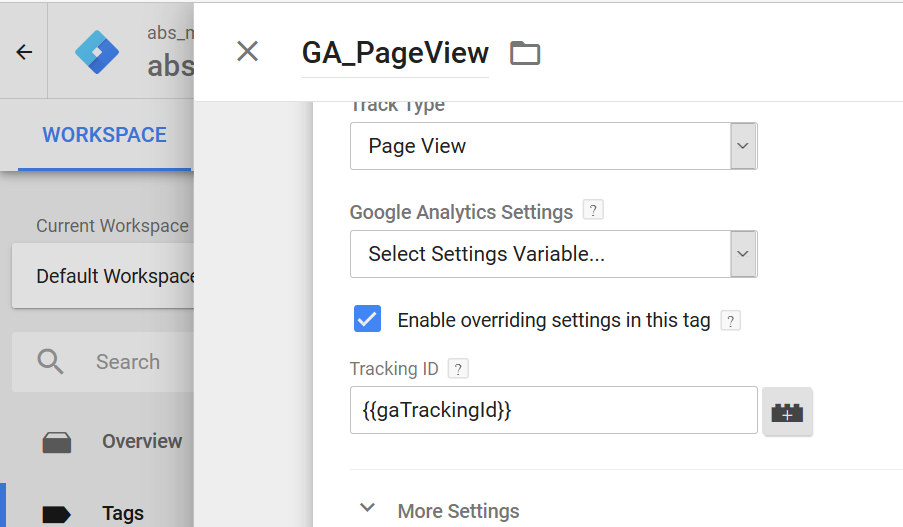
## Setup

### Step 0: Define a measurement plan [Analytics Academy](https://analyticsacademy.withgoogle.com/course01/unit?unit=2&lesson=4)

### Step 1: Setup GTM

### Step 2: Link to GA

When create Tag (in GTM), select “**Enable overriding settings in this tag**”, add GA tracking ID (eg. UA-99257465-1). Usually via a Variable constant.



# Event

## Unique Events and Unique Dimension Combinations

<https://support.google.com/analytics/answer/7084499?hl=en>

### Unique Events metric

Unique Events are interactions with content by a single user within a single session that can be tracked separately from pageviews or screenviews.

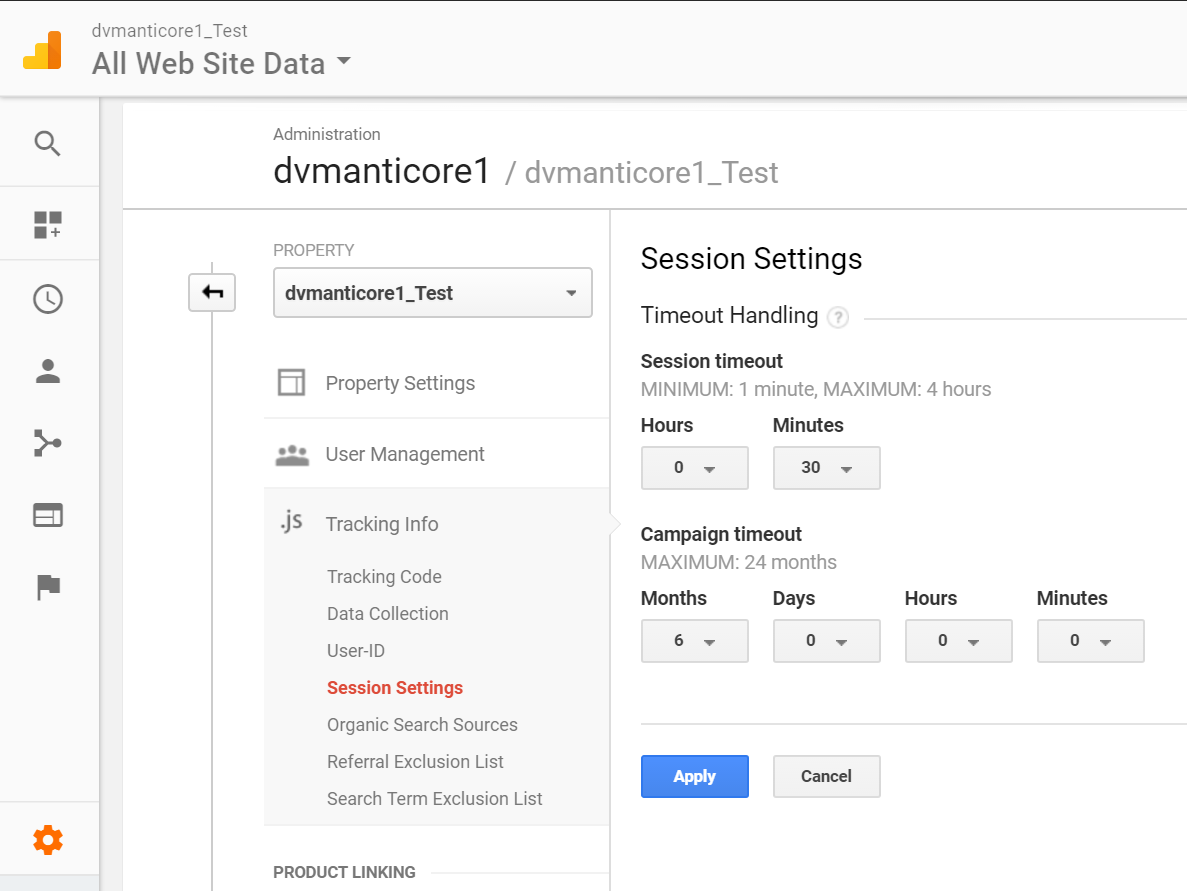
For example: During the same session, a user clicks the *Play* button for the same video 5 times:

* Unique Events = 1
* Total Events = 5

#### How a web session is defined in Analytics

<https://support.google.com/analytics/answer/2731565?hl=en>

By default, a session lasts until there's 30 minutes of inactivity, but you can adjust this limit so a session lasts from a few seconds to several hours. [Learn more about adjusting session settings](https://support.google.com/analytics/answer/2795871).



# History

## GTM

Version 1 announced in October 2012.

Version 2 launched in October 2014. Google also announced the Tag Manager [API](https://en.wikipedia.org/wiki/API) with the release of version 2.

## GA

UPDATE January 20, 2015:  All new Google Analytics accounts can now only use Universal Analytics.

ga.js is a legacy library. GA

[analytics.js](https://developers.google.com/analytics/devguides/collection/analyticsjs/) latest version. UA

New UA provide custom dimensions, but still have custom variables. All new custom variable type work using custom dimensions and metrics.

## Google Analytics 360 Suite

Regular Google Analytics users get 20 of them, while Google Analytics Premium users get a whopping 200 of them.

Data management platform announced on 15 March 2015. There are six products in the suite: Analytics 360 (Google Analytics), Tag Manager 360(Google Tag Manager), Optimize 360, 360 attribution, Audience Center 360, and Data Studio 360.

Compare Chart with standard version <http://www.blastam.com/google-analytics-360-standard-comparison>

Pricing: $150k/yr - $180k/yr (from [Quora](https://www.quora.com/What-is-the-cost-of-Google-Analytics-360-Suite)).

Biggest competition is the **Adobe Marketing Cloud**.

# Reference

<https://www.optimizesmart.com/beginners-guide-google-tag-manager-v2/>

<https://support.google.com/tagmanager/answer/6102821?hl=en>

<https://support.google.com/tagmanager/answer/6163791>

<https://support.google.com/tagmanager/answer/6103696>

<https://www.slideshare.net/liip/google-tag-manager-an-introduction?from_action=save>

<https://support.google.com/tagmanager/answer/6163827>