Analytics mania

8 Steps to Become An Experienced Google Tag Manager User

Created by

Julius Fedorovicius
Founder of Analytics Mania
julius@analyticsmania.com
@fedorovicius

About The Author

Hi, I'm Julius Fedorovicius, Founder of Analytics Mania.

In 2011, I started actively using Google Analytics, in 2014 – Google Tag Manager.

Since 2011, I've spent the first eight years of my professional career at Omnisend being surrounded by developers, thus learned various tricks from them. This experience helped me better understand how digital tools work and gave me more curiosity to dive into Google Analytics and Google Tag Manager. Thus, I became the company's "GA and GTM guy".

In 2016, I started running on-premise Google Tag Manager workshops in Vilnius, Lithuania. At the same, time I also decided share my knowledge by writing blog posts. That's when Analytics Mania was born. The blog constantly grew, I started to help other people in the community and also learn a lot from them/their cases.

Between 2017 and 2019, I also had various side-projects and side-gigs related to web analytics. At that time, I had a full-time job at Omnisend, and I was writing a blog, helping others in the community, running GTM workshops, and spending time with my wife and kids. To sum up, I was having two full-time jobs with little to no sleep.

In 2018, I launched my first Google Tag Manager online course for beginners, and that's where things started to kick-off.

Also, I founded a Google Tag Manager community on Facebook, which is currently the largest GTM forum in the world.

In 2019, I left the awesome team of Omnisend and am now working fully on Analytics Mania. I continue to write blog posts, publishing online courses, working with existing and new clients.

Everyone interested in learning ins and outs of Google Tag Manager are welcome at analyticsmania.com.

Disclaimer

The first thing that I want to emphasize here is that this e-book is designed for intermediate GTM users -- for those who have been already using GTM for at least six months and find no troubles in the form tracking, conversion tracking, are starting to get familiar with terms like cookies, auto-event variables, etc.

IMPORTANT

So, if you are starting with GTM (or have just recently jumped on this train), I suggest you postpone reading this e-book.

Here are several questions to test your knowledge:

- Do you know what auto-event tracking is?
- Do you know what data layer is?
- Do you know both ways how to enable auto-event listeners in GTM?

If you answered NO to **at least one** of these questions, it’s too soon for you to read this e-book.

I believe that there are still many things (belonging to the beginner zone) that you should learn first (like the relationships between tags, triggers, and variables, etc.). My other e-book GTM for Beginners might be a good resource for you.

Introduction

I started using GTM in 2014, and there was almost no information about it available at that time. Therefore, I had very long, frustrating sessions of trial and error.

Today, I wish I had taken a course on GTM in the past and got up-and-running much faster. First of all, back then, there was no GTM course available in the market.

But there is another (even worse) thing. Even if some GTM course existed back in the day, I wouldn't have taken it. Why? Because then I was unnecessarily stubborn and thought that I could learn things myself.

Now I realize that I was wrong.

I always used to take the long path of self-educating. This choice was mainly driven by the approach "it's cheaper that way". However, after 3-4 years, I realized that trying to figure out things on your own is a much more time-consuming and expensive approach.

Yes, I still had my money in the bank account and didn't purchase anything, however, the number of hours I've spent on self-learning was mindboggling. If I converted all those hours into monetary value (because I know my hourly rate), the final sum would be insane, probably 10x or 50x course's price.

Having someone to teach you a topic X via course can get you quite comfortable with that topic over the course of several weeks. In comparison, learning something on your own would probably take me months (and you would still make A LOT of mistakes along the way that could have been easily avoided if you had a mentor).

Why am I telling you this?

Because now you have options to take shortcuts in the learning process (unlike in 2014). Don't repeat my mistakes.

I hope that this e-book will work as an introduction to your next step of Google Tag Manager master.

But if in the future, you get an opportunity to take a real-shortcut (e.g., an online course, on-premise workshop or something similar (where the instructor/mentor is teaching you)), consider it more carefully. It may cost you considerably a lot of money in the beginning, but over the time that investment will pay off.

And now, let’s go to the *meaty* party of this e-book.

In this e-book, I will briefly explain several topics/areas that you should know/learn in order to successfully transition from the fresh intermediate user to a more experienced GTM professional.

#1 Learn Data Layer

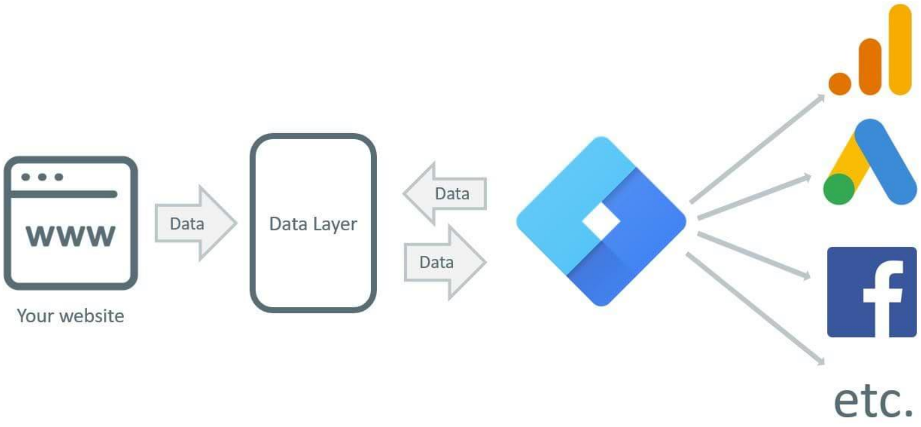
Data Layer is the essential component of Google Tag Manager. Without it, event tracking would not work, and the majority of variables within GTM would be useless.

Important!

I will not go into the introduction of the Data Layer because you should already know that. After all, this e-book is for intermediate users, and the topic itself is introduced in the beginner zone.

Here are the core things you need to know about the Data Layer (there are more, but just make sure that you first get the fundamentals right).

Data Layer is an intermediary layer between your website and Google Tag Manager. If you want to have some information sent to Google Tag Manager (e.g., purchase data), it first must be sent to the Data Layer. From there, GTM will take the info and process it based on your configuration.

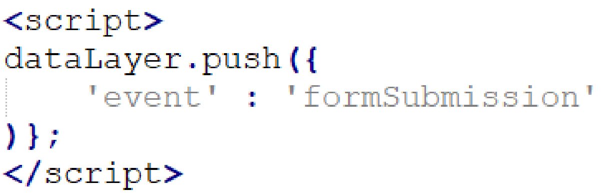


Data Layer is not a standard/default part of the website. It is usually bound to Google Tag Manager. Websites without GTM usually don't have the Data Layer. When GTM loads, it automatically initiates the Data Layer. Also, a developer/plugin can create/set one up.

The only working way to add data to the Data Layer is via *.push()* method. With help of *dataLayer.push()*, events and data points can be sent to the Data Layer. From there, GTM can take that information and use it in tags, triggers, and variables.

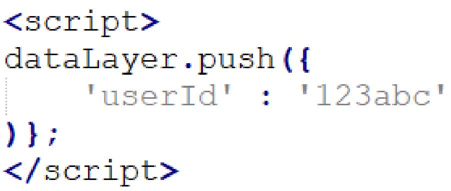
By pushing data to the Data Layer, you're feeding Google Tag Manager's internal data model, and with help of Data Layer variables, you can access that data and reuse in your tags/triggers/variables.

Code example of an event →



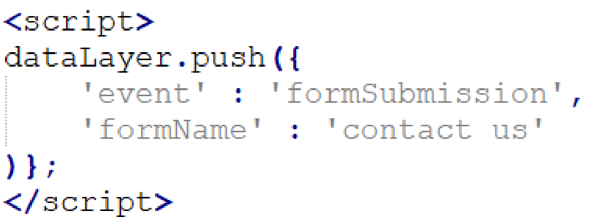
```
<script>
dataLayer.push({
    'event' : 'formSubmission'
});
</script>
```

Code example of a data point →



```
<script>
dataLayer.push({
    'userId' : '123abc'
});
</script>
```

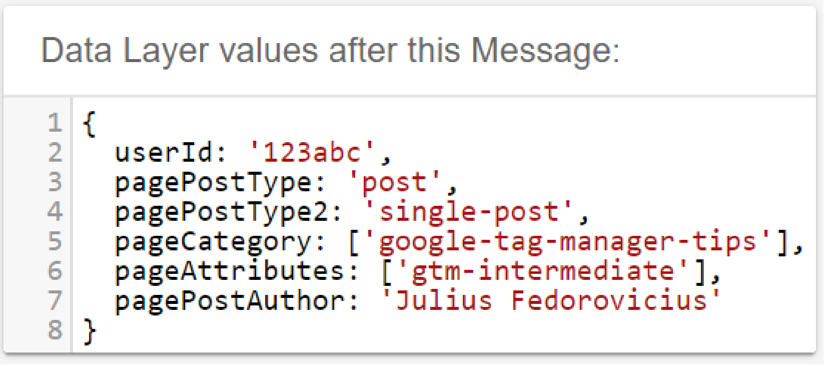
Code example of an event + data point →



```
<script>
dataLayer.push({
    'event' : 'formSubmission',
    'formName' : 'contact us'
});
</script>
```

If some custom data is pushed to the Data Layer, Google Tag Manager does not recognize it by default. I mean, you will not be able to use it as a variable out of the box.

Your Data Layer might contain various information about a user, page, or whatever…

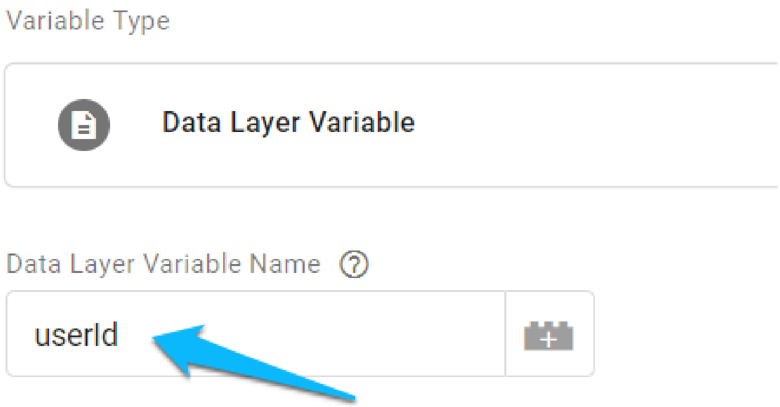


Data Layer values after this Message:

```
{
  userId: '123abc',
  pagePostType: 'post',
  pagePostType2: 'single-post',
  pageCategory: ['google-tag-manager-tips'],
  pageAttributes: ['gtm-intermediate'],
  pagePostAuthor: 'Julius Fedorovicius'
}
```

…but by default, it will not be visible in the Preview and Debug tab's Variables section, therefore, you cannot use them in your tag management.

That's why you need to create a **Data Layer Variable** for each data point you want to access from the Data Layer. Create a new **Data Layer Variable** and enter the name of the key that you want to access in the Data Layer



Variable Type

Data Layer Variable

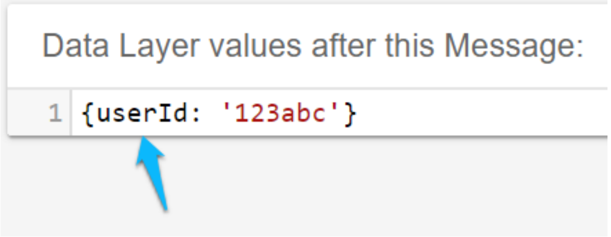
Data Layer Variable Name

userId

However, data can be stored in different structures. Therefore, there are some additional rules you need to keep in mind.

<table>
<tr><th>Data in the Data Layer</th><th>Data Layer Variable Configuration</th></tr>
<tr><td>

If key in the Data Layer is a primitive value (regular text (string) or number)

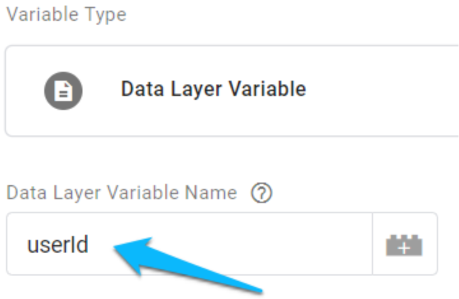


Data Layer values after this Message:

```
{userId: '123abc'}
```

</td><td>

Just enter its name in this field



Variable Type

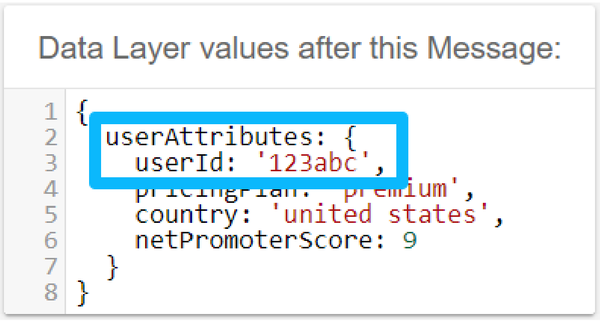
Data Layer Variable

Data Layer Variable Name

userId

</td></tr>
<tr><td>

If data is nested in the Data Layer

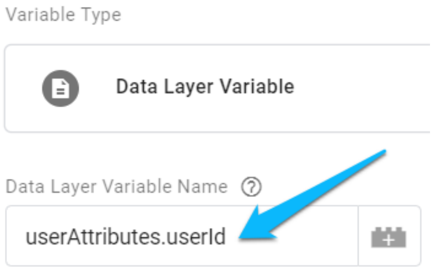


Data Layer values after this Message:

```
{
  userAttributes: {
    userId: '123abc',
    pricingPlan: 'premium',
    country: 'united states',
    netPromoterScore: 9
  }
}
```

</td><td>

You need to separate every level with a dot (also, this is case-sensitive)



Variable Type

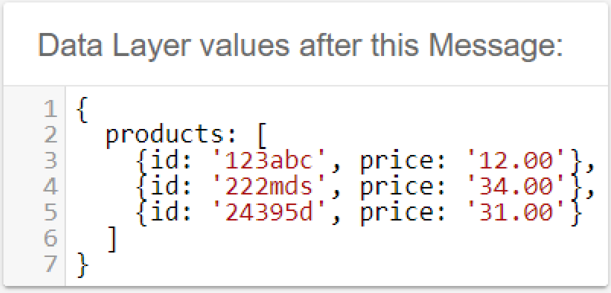
Data Layer Variable

Data Layer Variable Name

userAttributes.userId

</td></tr>
<tr><td>

If you’re working with an array (list of items)



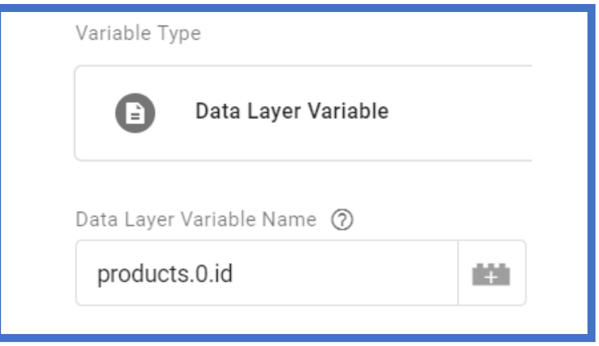
Data Layer values after this Message:

```
{
  products: [
    {id: '123abc', price: '12.00'},
    {id: '222mds', price: '34.00'},
    {id: '24395d', price: '31.00'}
  ]
}
```

</td><td>

You also need to define the index number of the item wish to access. P.S. in JS, the index starts not from 1, but 0. As a result, with the variable below, I’ll get the 1st product’s id.

</td></tr>
</table>



Variable Type

Data Layer Variable

Data Layer Variable Name

products.0.id

Remember, we’re just scratching the surface of the Data Layer here, and, unfortunately, we will not dive deeper (at least for now). The point of this e-book is to show you the direction and what areas you should dig into to up your GTM game.

If I covered all the possible things and edge cases, this would probably be more than 300 pages (which is way too long for a free e-book).

Related resources

- Official GTM’s documentation

#2 Getting Familiar with HTML

HTML is the language in which most websites are written. HTML is used to create pages and make them functional. If you see a website element (e.g., button, table, etc.), most likely, it was created with help of HTML.

When you get started with HTML, you need to learn what are *elements*, *tags*, and *attributes*.

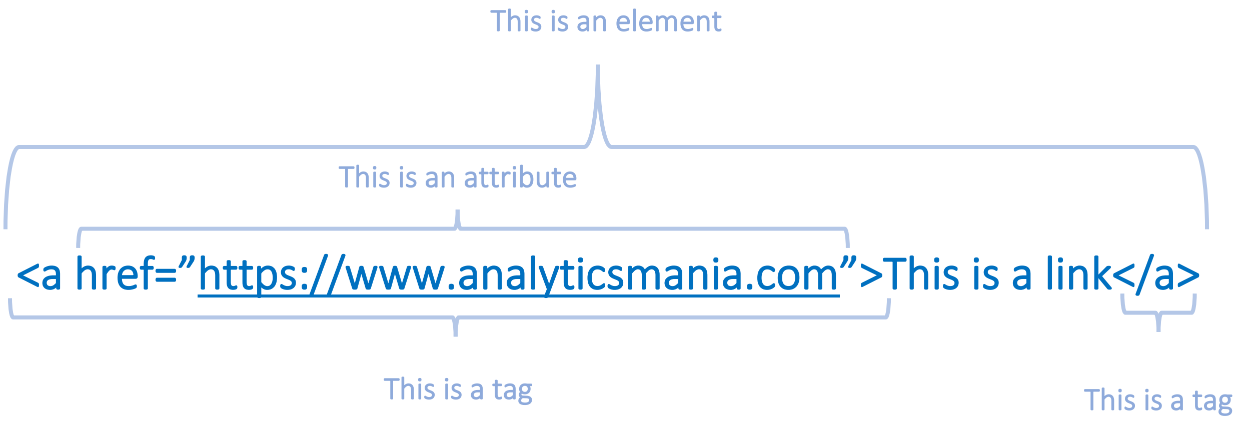
The example of an element is **paragraph, table, heading,** etc.

Tags are used to mark up the start of an HTML element, and they are usually enclosed in angle brackets. An example of a tag is: **<h1>**

Attributes contain additional pieces of information. Attributes take the form of an opening tag, and additional info is placed inside. An example of an attribute: **<a href="https://www.analyticsmania.com">This is a link</a>**

In this example, *href* is an attribute.

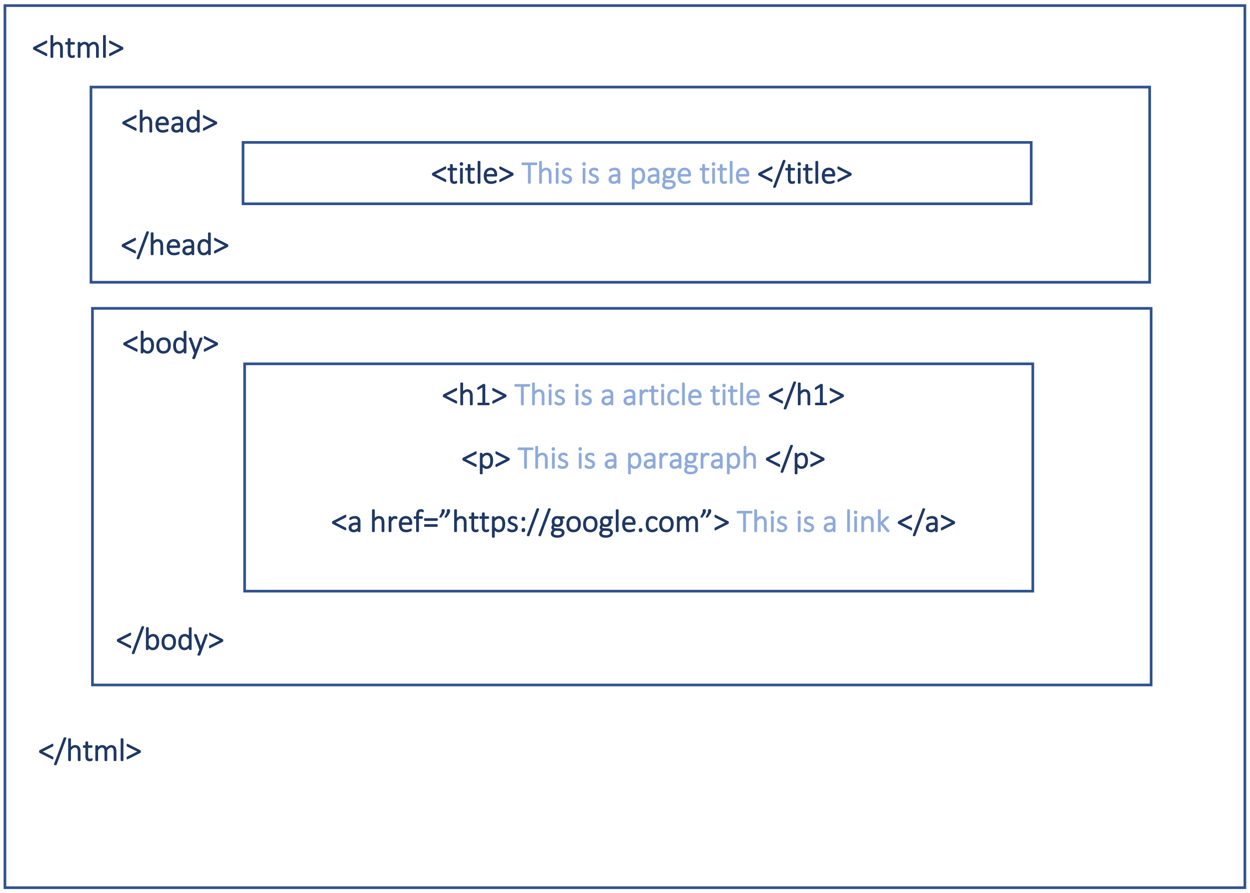
All in all, here's a picture that shows the relationship of an *element*, *tag*, and *attribute*.



There are various types of HTML tags. To name a few:

- **<html>** - this tag signals that from here on the code is written in HTML
- **<head>** - this is where the metadata of the page will go (like keywords, title, etc.)
- **<body>** - This is where the content of the page goes
- **<h1>** - heading 1
- **<table>** - that's the table
- **<p>** - paragraph
- **<title>** - page title
- **<a>** - link (*a.k.a.* anchor)
- and many more…

Every website's code consists of <head> and <body> (both at surrounded by the <html>



```
<html>
    <head>
        <title> This is a page title </title>
    </head>
    <body>
        <h1> This is a article title </h1>
        <p> This is a paragraph </p>
        <a href="https://google.com"> This is a link </a>
    </body>
</html>
```

As you can see, there are relationships between HTML elements. For example, in the previous scheme, the top-level element is <html>, it has two children, <head> and <body>. <body> has multiple child elements, a heading 1, a paragraph, and a link.

A link element (<a>) contains an attribute called *href*. This attribute instructs a browser where a visitor should be redirected after a link is clicked.

HTML elements can contain other attributes as well, e.g., ID, class, etc. I marked them in different colors.

```
<a class=”red” id=”main-cta” href=”https://google.com”> This is a link </a>
```

How can HTML knowledge be useful in Google Tag Manager? There are many use cases. Here’s one of them:

Imagine that a web page contains an image. That image can be clicked, and a visitor will be redirected to a pricing page (the image also contains a button imitation). Here’s an example:

#stockPhotos

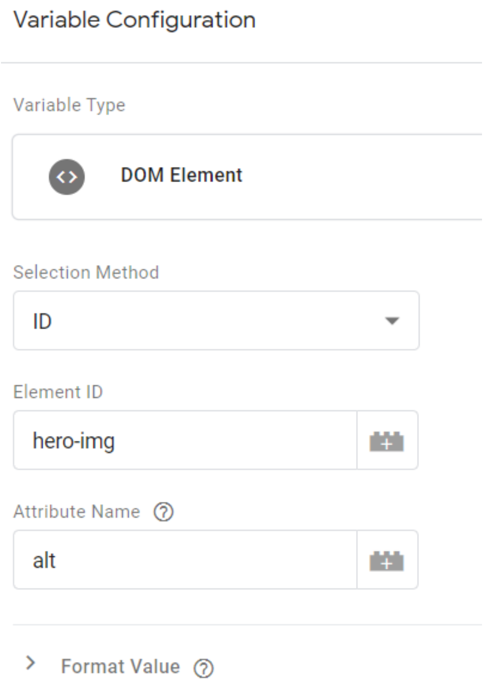
That image’s code is:

```
<img class=”full-width-main-img” id=”hero-img” src=”main-img.jpg” alt=”learn more about the pricing”>
```

Say that you want to access image’s *alt* text and use it as a variable (with every pageview you want to send which exact image was displayed to a visitor so that you could track which one gets more impressions).

By understanding what element attributes in HTML are, you would be able to access that *alt* attribute pretty easily.

In Google Tag Manager, go to Variables > DOM Element and enter the following settings:



Variable Configuration

Variable Type: DOM Element

Selection Method: ID

Element ID: hero-img

Attribute Name: alt

DOM element variable accesses elements that are available on a page. Since our image has an **id** attribute, we used the **ID** selection method. In the **Element ID** field, we enter the value of the attribute (*hero-img*), and eventually, we entered the value of the attribute we want to access, *alt*.

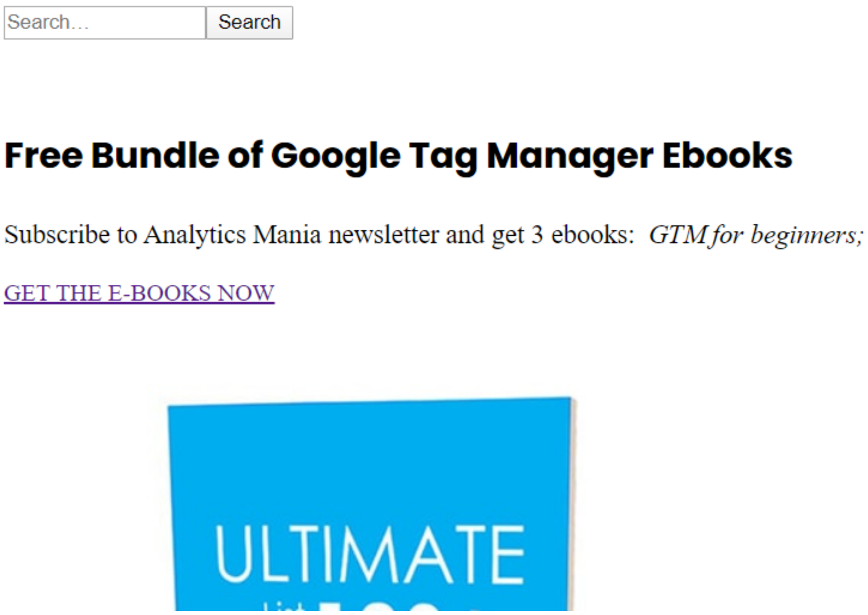
Once again, we are just scratching the surface of what is possible but I hope that this example gave you some ideas where to dig deeper.

Related resources

- HTML Tutorial (W3C)
- HTML Guides and Tutorials (Mozilla)

#3 Getting Familiar with CSS Selectors

If you code an element with HTML and don’t define style for every individual element, they might look like this. Here’s an example of how analyticsmania.com homepage would look without any styling.



Free Bundle of Google Tag Manager Ebooks

Subscribe to Analytics Mania newsletter and get 3 ebooks: *GTM for beginners;*

GET THE E-BOOKS NOW

Plain elements, blue/purple links, etc.

That’s where CSS becomes handy.

CSS is the language that defines the presentation of a web page. It is used to add color, background images, and textures, and to arrange elements on the page, etc.

How is this different from HTML?

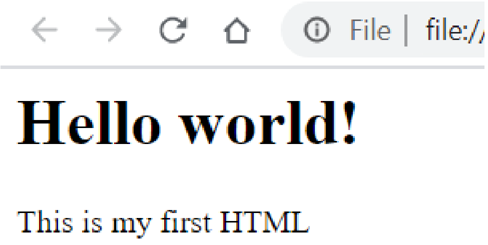
- All critical website content should be added to the website using a markup language such as HTML.
- Presentation of the website content (*a.k.a.* how that website content looks) should be defined by a styling language such as CSS.

To sum up. HTML – content. CSS – style.

Here's an example of a plain HTML code:

```
<html>
        <head>
                <title>This is text</title>
        </head>
        <body>
                <h1>Hello world!</h1>
                <p>This is my first HTML</p>
        </body>
</html>
```

Once the browser renders this code, the page will look like this:



Hello world!

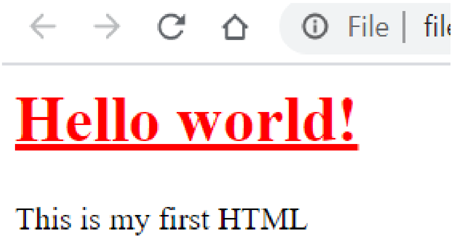
This is my first HTML

Pretty simple, right? Here we see two content-related elements, h1 ("Hello world!") and p ("This is my first HTML").

However, if we wanted to "spice things up" and change the style of these elements, one of the ways would be with the help of CSS. Here's an example of the code:

```
h1 {
    color: red;
    text-decoration: underline;
   }
```

Thanks to this code, the final result of my *Hello world* page would look like this:



Hello world!

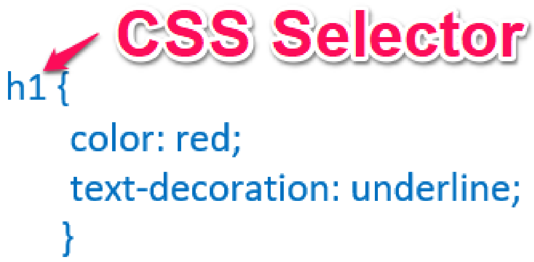
This is my first HTML

Even though this visually looks terrible, the goal was achieved. The style of an element was changed.

The key component (for GTM users) in this chapter is a term called *CSS Selector*.

CSS Selectors allow developers to select certain elements on a page by using a particular syntax.

In the code below, the CSS selector is ***h1*** (which means that "all h1 elements on a page are selected") and a custom style should be applied (*color: red* and *text-decoration: underline*)



```
h1 {
      color: red;
      text-decoration: underline;
     }
```

If you wanted to apply a style just to certain headings on a page (not ALL of them), you would need to write a more precise CSS Selector.

How is this useful in Google Tag Manager?

You can use CSS selectors to instruct GTM which elements you are interested in. Let's return to the previous chapter's example, the image that can be clicke (and then a visitor is redirected to the pricing page).

This was that image's HTML code:

```
<img class="full-width-main-img" id="hero-img" src="main-img.jpg" alt="learn more about the pricing">
```

Now, what if that image did not have an **id** attribute?

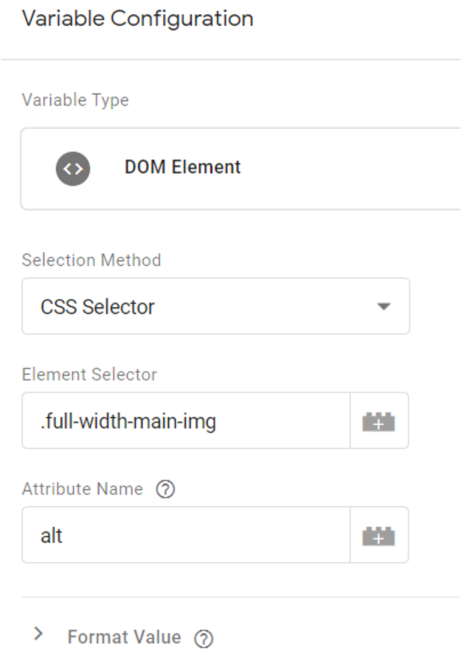
```
<img class="full-width-main-img" src="main-img.jpg" alt="learn more about the pricing">
```

Without it, we wouldn't be able to select which element's **alt** attribute do we want to access. Or would we?

Thanks to CSS Selectors, we would be able to pick the correct element on a page.

Let's go back to the DOM Element Variable we created in the previous chapter. First, you would need to change the Selection Method from *ID* to *CSS Selector* and then enter the actual selector (in the *Element Selector* field).

Since we are interested in an element of which **class** attribute *full-width-main-img*, the final CSS Selector would be **.full-width-main-img**



Variable Configuration

Variable Type: DOM Element

Selection Method: CSS Selector

Element Selector: .full-width-main-img

Attribute Name: alt

The dot in CSS Selectors means that we are targeting elements based on their **class** attribute.

Related resources

- CSS Tutorial (W3C)
- CSS Selectors examples (W3C)

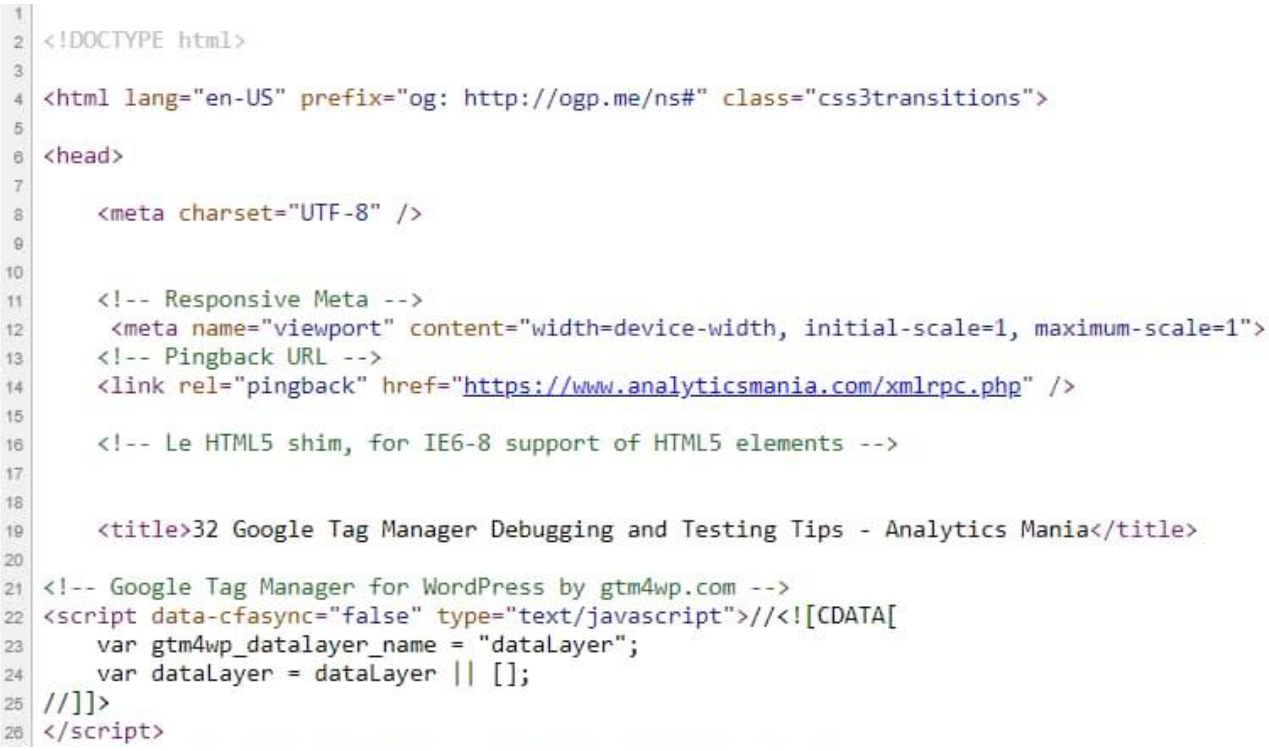
#4 Getting Familiar with DOM

First, you need to have at least a very basic understanding of what DOM is in general. **Document Object Model** (a.k.a. the **DOM**) is a dynamic tree-like hierarchical representation of the website's document.

Sounds complex, right? Let me explain.

It allows manipulating the website's document, get values of current website elements, add new, edit or remove existing ones.

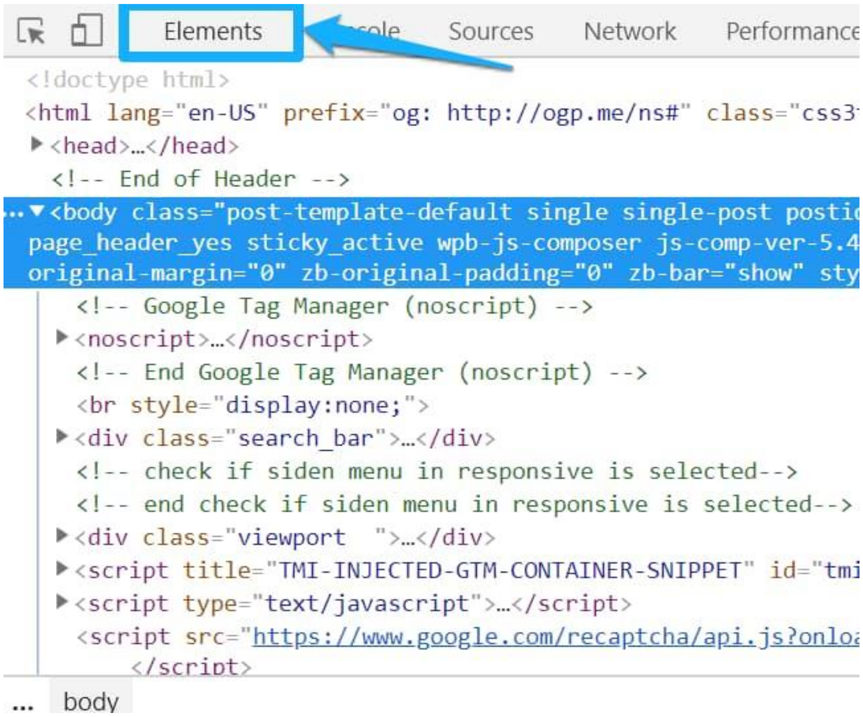
If you go to any website and open its source (e.g., CTRL+U on Chrome (*Windows*)), you'll see the HTML code of the page. This is the code that a developer wrote to make the website what it looks right now.



```

<!DOCTYPE html>

<html lang="en-US" prefix="og: http://ogp.me/ns#" class="css3transitions">

<head>

    <meta charset="UTF-8" />


    <!-- Responsive Meta -->
     <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
    <!-- Pingback URL -->
    <link rel="pingback" href="https://www.analyticsmania.com/xmlrpc.php" />

    <!-- Le HTML5 shim, for IE6-8 support of HTML5 elements -->


    <title>32 Google Tag Manager Debugging and Testing Tips - Analytics Mania</title>

<!-- Google Tag Manager for WordPress by gtm4wp.com -->
<script data-cfasync="false" type="text/javascript">//<![CDATA[
    var gtm4wp_datalayer_name = "dataLayer";
    var dataLayer = dataLayer || [];
//]]>
</script>
```

But it does not look very dynamic.

If you go back to that very same website and open developer tools (Chrome on Windows: press F12, Chrome on Mac: Command + Option + I), you'll see a more dynamic representation of the website's code. If not, switch to the Elements tab.



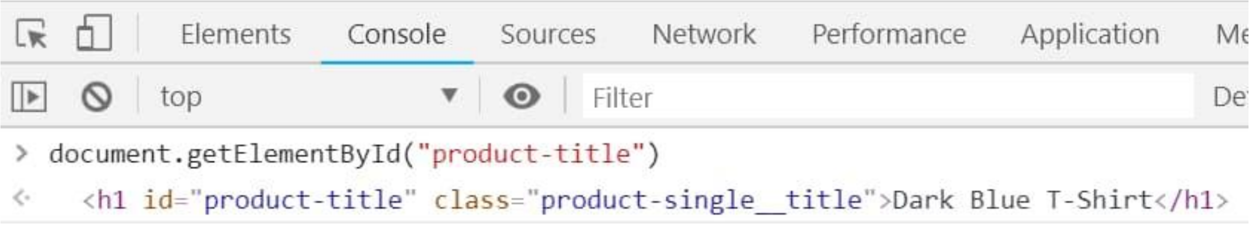
```
<!doctype html>
<html lang="en-US" prefix="og: http://ogp.me/ns#" class="css3
 ▶<head>…</head>
  <!-- End of Header -->
…▼<body class="post-template-default single single-post postic
 page_header_yes sticky_active wpb-js-composer js-comp-ver-5.4
 original-margin="0" zb-original-padding="0" zb-bar="show" sty
    <!-- Google Tag Manager (noscript) -->
   ▶<noscript>…</noscript>
    <!-- End Google Tag Manager (noscript) -->
    <br style="display:none;">
   ▶<div class="search_bar">…</div>
    <!-- check if siden menu in responsive is selected-->
    <!-- end check if siden menu in responsive is selected-->
   ▶<div class="viewport  ">…</div>
   ▶<script title="TMI-INJECTED-GTM-CONTAINER-SNIPPET" id="tmi
   ▶<script type="text/javascript">…</script>
    <script src="https://www.google.com/recaptcha/api.js?onloa
        </script>
```

Some nodes can be expanded or collapsed (by clicking dark triangles), many nodes have children that can have their children, nodes can have multiple children, etc. It's a dynamic tree-like representation of website's documents. You're inspecting the HTML DOM here.

As I have mentioned before, DOM allows not only to view/inspect website elements but to manipulate them as well. The easiest and most down-to-earth way to edit nodes in the DOM is to double-click a certain node and then change its value or delete/edit/add attributes to an element.

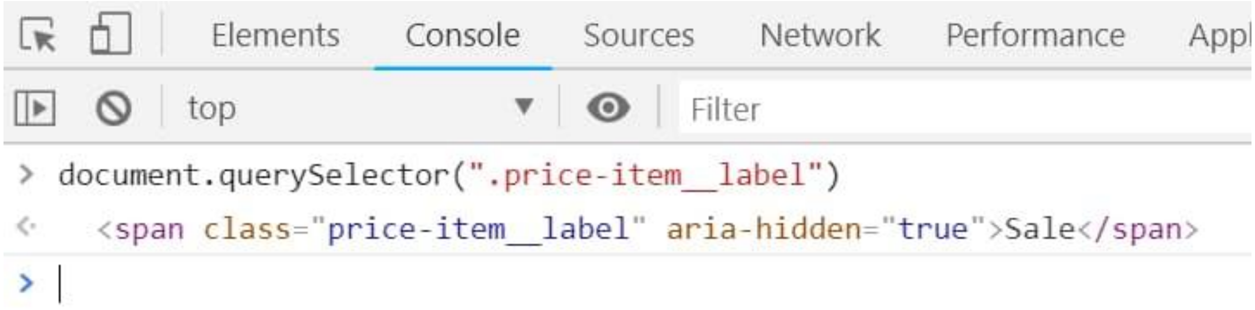
But this is not very scalable and useful. Instead, there should be some better way to do that, right? Yes. DOM supports a bunch of methods that enable developers (and you, if you are skilled enough) to add, edit, delete elements.

As for reading values, *getElementById("some_id")* method will return you the first website element that matches a ***some_id*** ID. In the screenshot below, you can see the command (1st line) and the returned element (2nd line).



```
> document.getElementById("product-title")
<·   <h1 id="product-title" class="product-single__title">Dark Blue T-Shirt</h1>
```

Another example could be ***querySelector()***, that returns the first website element that matches a certain CSS Selector defined by you.



```
> document.querySelector(".price-item__label")
<·   <span class="price-item__label" aria-hidden="true">Sale</span>
> |
```

And that's a good opportunity to introduce Google Tag Manager's DOM Element variable. This type of variable can access the value of a website element that matches a condition defined by you.

How is this related to Google Tag Manager?

DOM Element is the variable that uses the DOM to return values of particular elements you want to access.

Yes, that is the same variable we used in two previous chapters of this e-book.

Reading values of the DOM and using them in your setup is called *DOM Scraping*. Don't abuse it too much because your CSS Selectors in the DOM variable are fragile, and changes in the website's HTML code or DOM can break your setup.

The more advanced you become with GTM, the better you will notice the risks.

#5 Getting Familiar with Regular Expressions

Regex (also known as regular expressions) is a sequence of characters that is used to search for one or more matches in search pattern. It is a way to look for patterns in a sentence.

To use Regex in Google Tag Manager, I recommend learning at least the very basic syntax.

Regular Expressions use symbols that have "special powers". Here are the most common and basic ones:

Regex Symbol	Meaning	Example
| (a.k.a. a *pipe*)	OR	a|b (this will match letter "a" or "b")
. (dot)	Any single character	a.c (will match aac, abc, acc, etc.)
* (asterisk)	Zero or more of the previous character	a* (will match [nothing], a, aa, aaa, aaaa, etc.) .* (will match literally ANYTHING)
^	Start of the string (text)	^gtm (will match "gtm is cool", will not match "cool is gtm")
$	End of the string (text)	gtm$ (will match "cool is gtm", will not match "gtm is cool") ^gtm$ (must exactly match "gtm")
?	One previous symbol is optional	docx? (will match "doc" and "docx")
()	Group elements	Dec(ember)? (will match "Dec" and "December")
\ (backslash)	Gives a special meaning for the character that comes after it	But the most common use case among beginners is this: • If \ comes before a Regex symbol, that symbol will become a regular symbol, e.g., \? Will

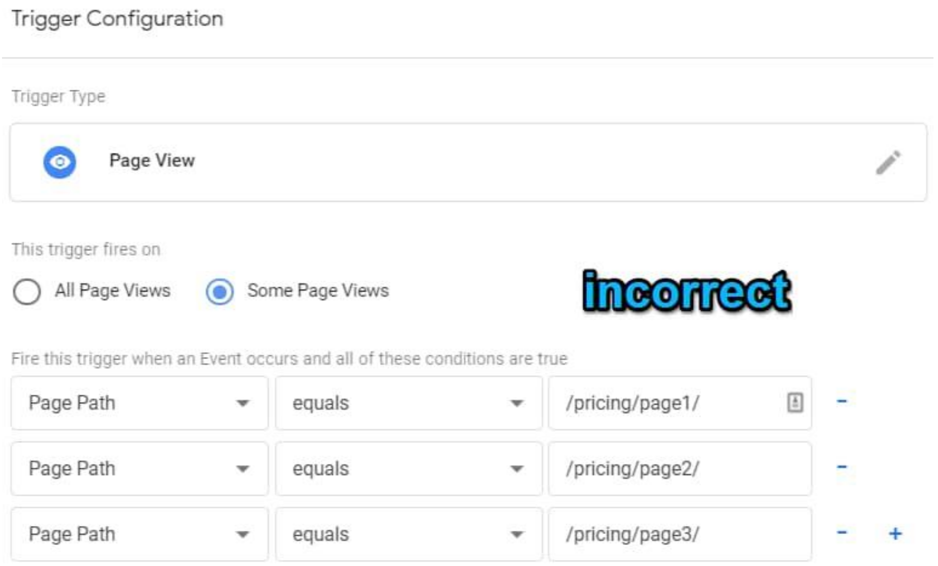
		actually match the question mark "?". In this case, \ is called "escape" character.

How are Regex useful in Google Tag Manager? Here is an example:

You have a tag that you want to fire on several occasions:

- When Page Path is/pricing/page1/
- OR when Page Path is /pricing/page2/
- OR when Page Path is /pricing/page3/

If you entered all conditions in a single trigger, that would not work.



Trigger Configuration

Trigger Type

Page View

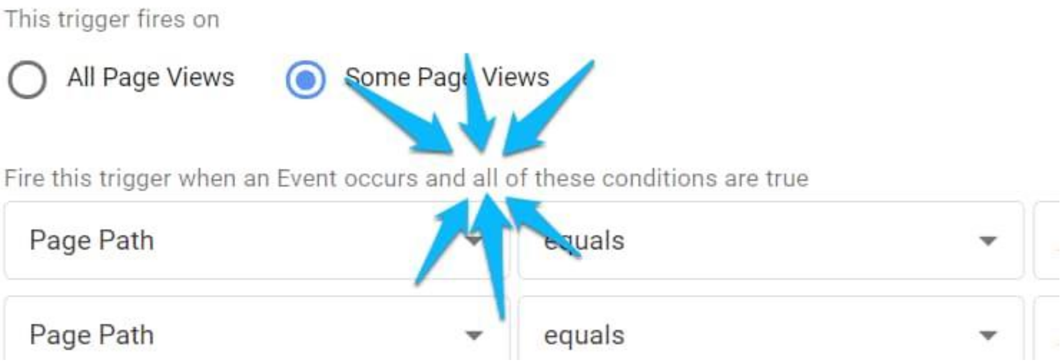
This trigger fires on

○ All Page Views ● Some Page Views

Fire this trigger when an Event occurs and all of these conditions are true

Page Path	equals	/pricing/page1/
Page Path	equals	/pricing/page2/
Page Path	equals	/pricing/page3/

Why? Because if a trigger contains multiple conditions, ALL of them must be met (and a Page Path cannot simply contain all those three conditions)



This trigger fires on

○ All Page Views ● Some Page Views

Fire this trigger when an Event occurs and all of these conditions are true

Page Path	equals
Page Path	equals

What alternatives do you have?

- Option A: Create three separate triggers to a single tag. If one of the triggers is activated, the tag will fire
- Option B: Use Regex in a single trigger (which is a much better option) where Page Path *matches RegEx* /pricing/page(1|2|3)/ (or some other pattern that meets your needs).

This will make your container cleaner and more optimal.

Another use case (but not the only one) where Regex is also useful is Regex Table Variable.

Related resource

- Verify your regular expressions with regex101
- JavaScript RegEx reference

#6 Getting Familiar with JavaScript

This topic is the most complex from this e-book. I will not dive deep into it because there are separate books and courses dedicated just to JavaScript. Therefore, you should not expect me to explain it in chapter of a small e-book.

Also, I am still in the progress of learning it.

My point in this chapter is to say that even knowing a little about JavaScript is already valuable in Google Tag Manager.

You can use it to get custom data from the page or do manipulation to the data you already have.

To illustrate my previous statement, let me show you a variable called **JavaScript Variable** (not Custom JavaScript).

I promise I'll try to explain this as simply as possible, so please, bear with me. Before we start digging deeper, you need to learn about the scope. In JavaScript, there are two types of scope:

- Local scope
- Global scope

Variables declared within a JavaScript function, become LOCAL to the function, meaning that they are not accessible (visible) from outside the function. Take a look at the example below:

```
// code here can not use authorName variable

function myFunction() {
 var authorName = "Julius";
 // code here can use authorName variable
}
```

There is a variable *authorName,* and it only can be accessed within the function called *myFunction* (because that variable is WITHIN the function)

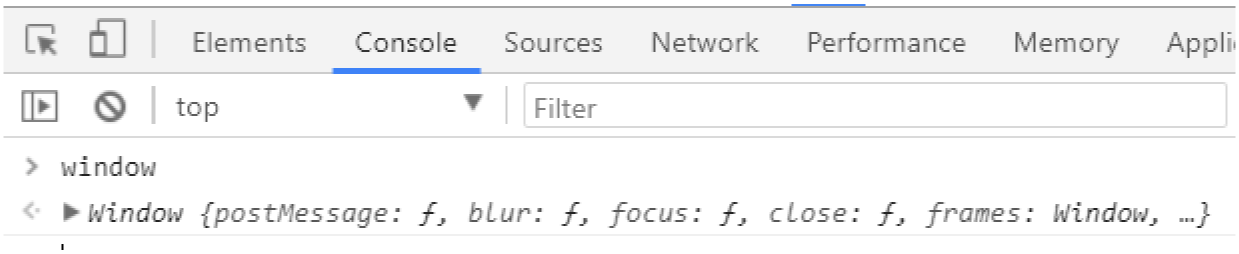
Contrary to that, a variable declared outside a function becomes GLOBAL. A global variable has global scope: all scripts and functions on a web page can access it. Let's use the same code example but place the variable outside the function.

```
// code here can use authorName variable
  var authorName = "Julius";
function myFunction() {

// code here can use authorName variable
}
```

What does it have to do with the JavaScript Variable in Google Tag Manager?

Well, it can fetch values of any global variable which is present at that moment on a page.

If you want to know what useful Global JavaScript variables are available on a page, go to the developer console of your browser (here's a tutorial how to do that on Chrome), type **window**, and hit enter.



```
> window
< ▶Window {postMessage: ƒ, blur: ƒ, focus: ƒ, close: ƒ, frames: Window, …}
```

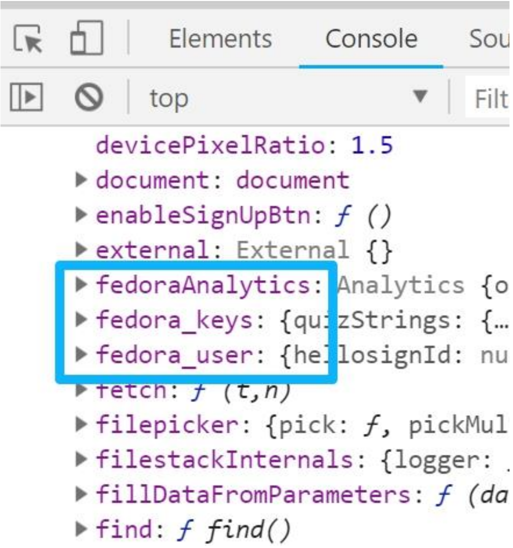
Click the black triangle near the Window, and you'll see a list of global JavaScript Variables.

Here's an example what useful information I could find.

I'm hosting my Google Tag Manager courses on a platform called *Teachable*.

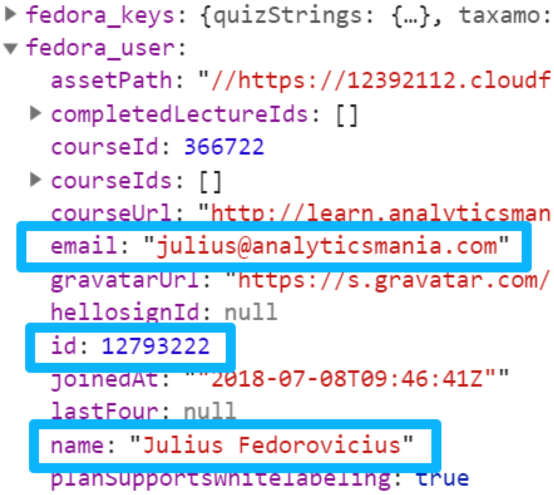
I know that previously Teachable was known as Fedora. So my guess would be to keep looking for variables somehow related to either *Teachable* or *Fedora*.

Let's start playing a detective. Bingo, I see three variables related to Fedora:



```
  devicePixelRatio: 1.5
▶ document: document
▶ enableSignUpBtn: ƒ ()
▶ external: External {}
▶ fedoraAnalytics: Analytics {o
▶ fedora_keys: {quizStrings: {…
▶ fedora_user: {hellosignId: nu
▶ fetch: ƒ (t,n)
▶ filepicker: {pick: ƒ, pickMul
▶ filestackInternals: {logger: .
▶ fillDataFromParameters: ƒ (da
▶ find: ƒ find()
```

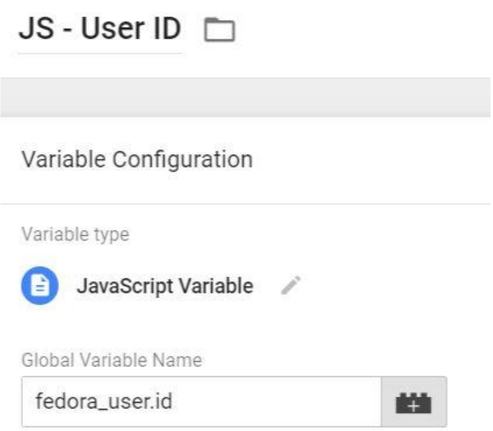
Since I'm currently interested in user data, I'll click the black triangle next to *fedora_user* to expand it. Bingo, there's a bunch of data that I could possibly use, for example, ID, name, etc.



```
▶ fedora_keys: {quizStrings: {…}, taxamo:
▼ fedora_user:
    assetPath: "//https://12392112.cloudf
  ▶ completedLectureIds: []
    courseId: 366722
  ▶ courseIds: []
    courseUrl: "http://learn.analyticsman
    email: "julius@analyticsmania.com"
    gravatarUrl: "https://s.gravatar.com/
    hellosignId: null
    id: 12793222
    joinedAt: ""2018-07-08T09:46:41Z""
    lastFour: null
    name: "Julius Fedorovicius"
    planSupportsWhitelabeling: true
```

Next, let's create a JavaScript variable for the **id.** In GTM, Go to Variables > New and choose JavaScript Variable.

Let's remember how we found that data point in the developer's console. First, we clicked *fedora_user* and then *id*. So the value that we need to enter in the JavaScript variable (in GTM) is *fedora_user.id* (every level needs to be separated by a dot if we are working with JavaScript objects).



JS - User ID

Variable Configuration

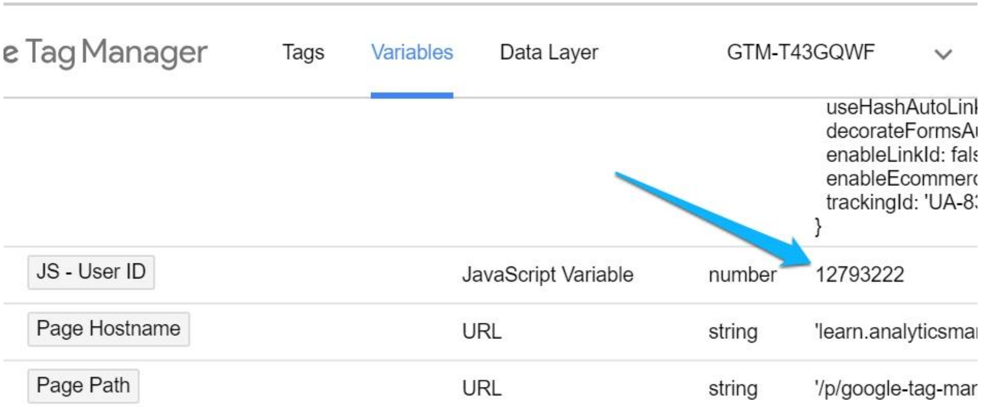
Variable type

JavaScript Variable

Global Variable Name

fedora_user.id

After you refresh the preview and debug mode, you should see that variable's value:



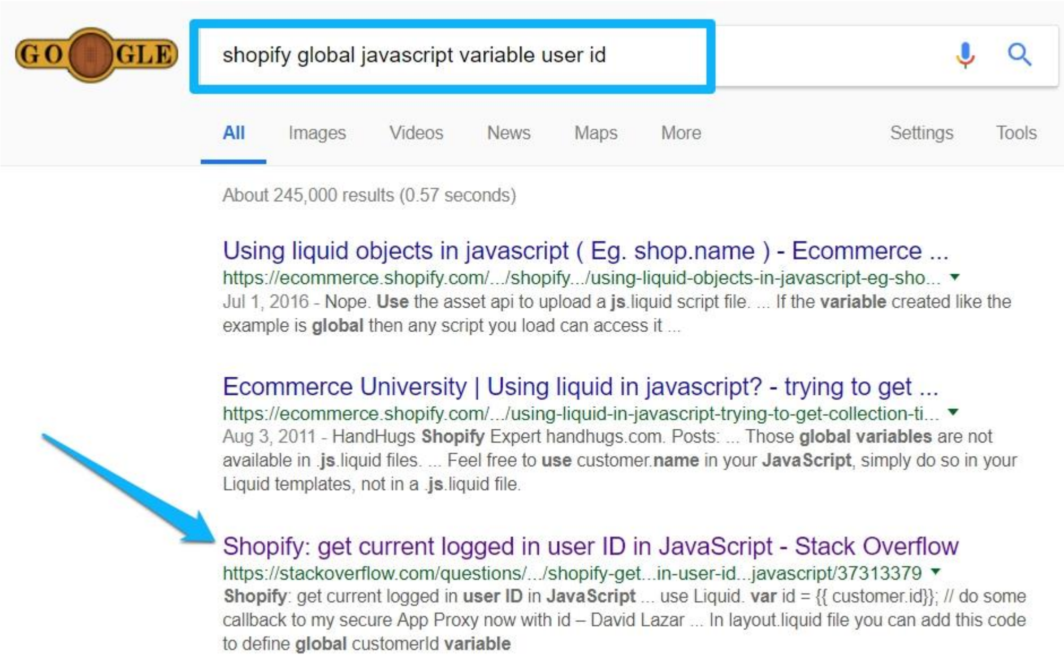
```
useHashAutoLink
decorateFormsA
enableLinkId: fals
enableEcommer
trackingId: 'UA-8
}
```

JS - User ID	JavaScript Variable	number	12793222
Page Hostname	URL	string	'learn.analyticsma
Page Path	URL	string	'/p/google-tag-mar

If you are working on a popular website/platform, chances are that someone has already described how to access that certain data.

For example, in Shopify, a logged-in user's ID can be found with the JavaScript Variable **ShopifyAnalytics.meta.page.customerId**.

How can you possibly know where to find such ID? Try googling it. For example, if you entered the following search query, you would find your answer regarding Shopify User ID:



shopify global javascript variable user id

About 245,000 results (0.57 seconds)

Using liquid objects in javascript (Eg. shop.name) - Ecommerce ...
https://ecommerce.shopify.com/.../shopify.../using-liquid-objects-in-javascript-eg-sho...
Jul 1, 2016 - Nope. **Use** the asset api to upload a **js**.liquid script file. ... If the **variable** created like the example is **global** then any script you load can access it ...

Ecommerce University | Using liquid in javascript? - trying to get ...
https://ecommerce.shopify.com/.../using-liquid-in-javascript-trying-to-get-collection-ti...
Aug 3, 2011 - HandHugs **Shopify** Expert handhugs.com. Posts: ... Those **global variables** are not available in .**js**.liquid files. ... Feel free to **use** customer.**name** in your **JavaScript**, simply do so in your Liquid templates, not in a .**js**.liquid file.

Shopify: get current logged in user ID in JavaScript - Stack Overflow
https://stackoverflow.com/questions/.../shopify-get...in-user-id...javascript/37313379
Shopify: get current logged in **user ID** in **JavaScript** ... use Liquid. **var** id = {{ customer.id}}; // do some callback to my secure App Proxy now with id – David Lazar ... In layout.liquid file you can add this code to define **global** customerId **variable**

If you’re working on a custom-built project, try asking developers. Maybe they will tell you what is the global JavaScript variable for the User ID.

Other areas where JavaScript knowledge can be valuable is Custom JavaScript Variables, Custom HTML tags or Custom Templates.

Related resources

- Basic JavaScript (Free Code Camp)

#7 Learning how to Properly Test Your Implementation

You can know all the possible tricks in GTM, but if you don't properly test your setup, everything is useless.

I cannot stress enough how testing is important in Google Tag Manager.

And this goes WAY beyond the Preview and Debug mode.

There are three rules that I follow when it comes to debugging GTM implementation:

1. Check whether a tag fired in the Preview and Debug mode
2. Check whether the network request was sent to your analytics/marketing tool
3. Check whether the data reached its final destination

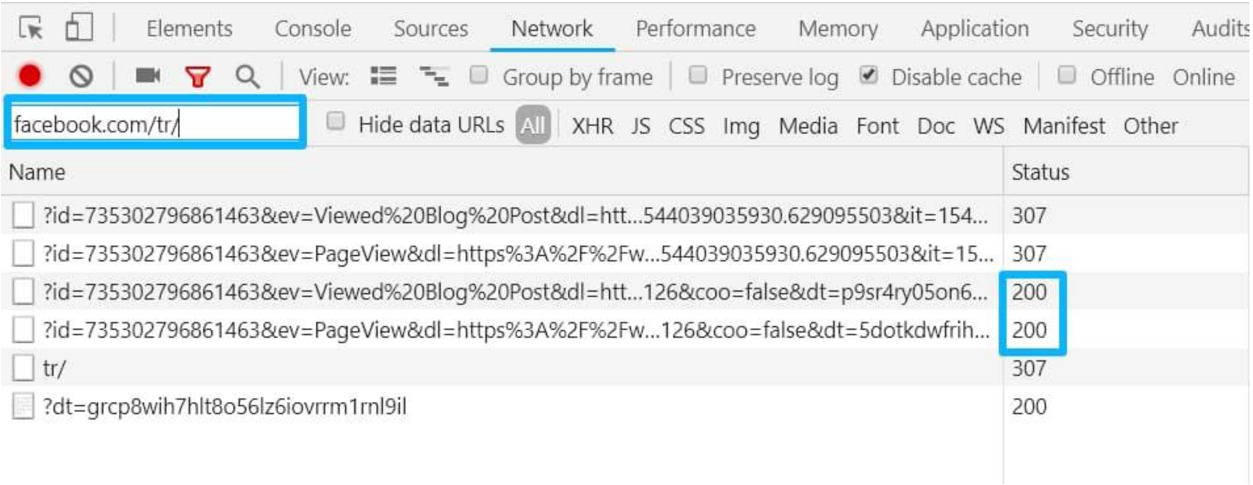
You should follow ALL these three rules. I've seen too many people relying on just GTM preview and debug mode, and that, unfortunately, is not a good thing.

#1. By this time, you should already be familiar with the debug mode. Therefore, I will not dive into the first rule. Let's move onto the built-in browser's tools.

#2. Check the Network tab in your browser's Developer Tools. In developer tools (on windows, you can press F12 while using Chrome), go to the **Network** Tab to see the list of requests that your browser has made on that page.

You can locate a specific type of request by using the filter feature. For example, if you're debugging Facebook Pixel requests, enter *facebook.com/tr/* to see the requests sent by Facebook Pixel.

If status codes are **200**, this means that everything is great. Sometimes internal redirects are possible, so **307** status code is also fine, as long as you see other requests getting the **200** OK.



facebook.com/tr/

Name	Status
?id=735302796861463&ev=Viewed%20Blog%20Post&dl=htt...544039035930.629095503&it=154...	307
?id=735302796861463&ev=PageView&dl=https%3A%2F%2Fw...544039035930.629095503&it=15...	307
?id=735302796861463&ev=Viewed%20Blog%20Post&dl=htt...126&coo=false&dt=p9sr4ry05on6...	200
?id=735302796861463&ev=PageView&dl=https%3A%2F%2Fw...126&coo=false&dt=5dotkdwfrih...	200
tr/	307
?dt=grcp8wih7hlt8o56lz6iovrrm1rnl9il	200

Another way how to check the requests is to use *Helper* browser extensions. For example, Facebook Pixel Helper is built to help you debug your Facebook Pixel tracking codes and whether they were successfully sent.

#3. Check the reports of your marketing/analytics platform ASAP. This sounds like an obvious thing; however, sometimes, it is still overlooked. If they exist, always utilize real-time reports:

- For example, in GA real-time reports, you can debug page views, events, conversions. When your tag is fired in the Preview and Debug mode, check the real-time reports, and see if the data is displayed there. Keep in mind that custom dimensions, custom metrics, e-commerce data are not available in RT reports.

If the platform you're using does not offer real-time debugging, check their regular reports regularly until your data comes in. If you notice mistakes, fix them as soon as possible.

Although there are many more tips related to GTM debugging, these three rules are the fundamentals/essentials that should always come first.

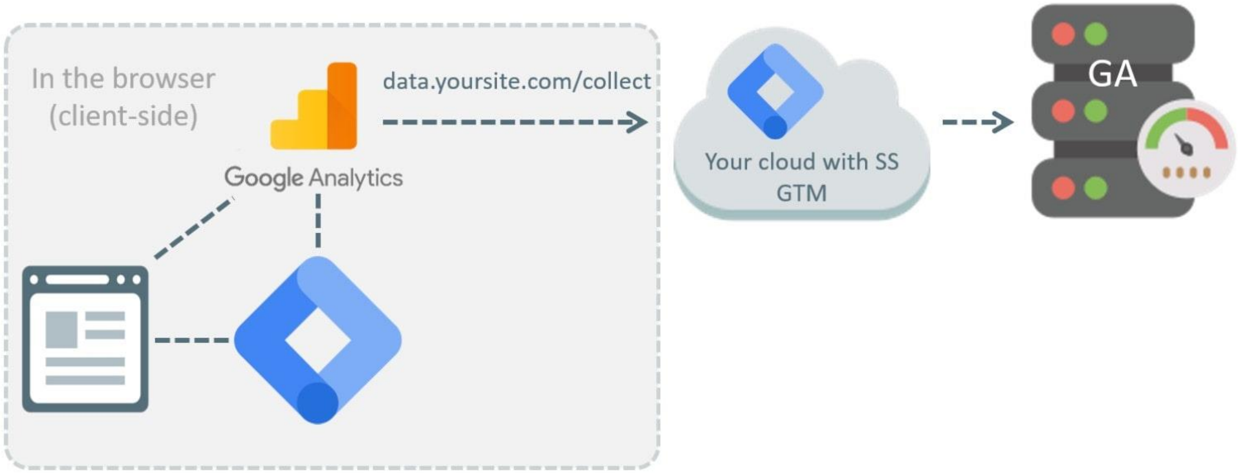
#8 Server-side Tagging, HTTP requests

Over the last several years, a major shift has been happening in the digital analytics space:

- Various privacy-related regulations are launched in different parts of the world (GDPR, CCPA, etc.)
- Browser creators constantly implement new privacy-enhancing features
- It's more important than ever for businesses to control what kind of data do they collect and for what purpose do they do that

As a result, a rise in popularity of server-side tagging has been noticed and it continues to grow. Especially, when Google Tag Manager introduced this feature in its own stack in early 2020 (although, this concept is not new).

What is server-side tagging? It is a technique that allows business to send analytics/marketing data to their own server from their website and then that server sends the data further to tools like Google Analytics, Facebook Pixel, etc.



There are many benefits of this approach. To name a few:

- Businesses have more control over what data is collected by third parties (like Facebook)
- You need to load less of tracking codes on your site (thus your page speed performance increases)
- Various privacy-enhancements (like adblockers or Apple's ITP) have lesser impact on your data quality

However, server-side tagging introduces additional technical topics that marketers/analysts should learn in order to understand how to work with it. And let me tell you – it's not simple. You will need to have some understanding of how the web works in general, get familiar with the concept of HTTP requests, their main components (like headers, body).

You will need to get familiar with what "serverless" means, what is Google Cloud Platform and App Engine.

Also, server-side tagging in Google Tag Manager introduces a new concept called *Clients*. They are the most important part of SGTM setups that is responsible for accepting the incoming requests to your tagging server.

There's a lot to cover and you could start with this tutorial (that also includes a video walkthrough).

Bonus: Learn More Technical & Challenging GA features

This is a little bonus chapter that serves as a reminder that multiple GA features are very challenging to implement.

If you master them, your value (as a digital marketer/analyst) will massively increase. You will be able to charge your clients more for them, or you will be able to complete those tasks faster.

To prove my point, take a look at online communities and forums. You will notice that those challenging topics appear more often when someone is asking for help.

So, what are those topics?

- GA4 Ecommerce implementation
- iFrame tracking

These two are the biggest headaches for intermediate Google Analytics and Google Tag Manager users.

I will not explain how to implement them (in this e-book) because every topic would require a separate e-book. This little chapter serves the purpose of a reminder.

If you are an aspiring digital marketer or analyst, and you often work with the implementation of tracking codes, these topics should eventually become one of your top items in the "To learn" list.

Personally, after learning them, I noticed an improvement in my business bottom line. I started to charge more for the setup and was able to implement it faster (hence accept even more projects to implement/fix such issues).

Final Words: 8 Steps to Become an Experienced GTM User

The goal of this e-book was to show you further direction, what topics to learn next if you want to become a Google Tag Manager professional.

If you're coming from a non-technical background, remember: it is not necessary to be an expert at each of those topics (e.g. Regular Expressions, HTML, etc.).

It's perfectly fine to know the basics to feel the boost in your capabilities.

Of course, the more you know, the better you perform. But even taking a small dip in every of these topics will definitely help you.

If you feel overwhelmed by this list and how much you still need to learn, do not fear. You are not alone. We have all been there. Just try to dedicate at least several hours a week, and you'll definitely see some progress.

Impossible to find spare time? Here are my tips on how I find time to keep learning. I'm not an expert in optimizing my time and getting the most out of it. On the other hand, my small efforts are still much better than doing nothing at all.

Also, if you're looking for a shortcut and a step-by-step system to get better at GTM, consider enrolling in my Intermediate/Advanced GTM course.

This e-book was delivered to you by

Julius Fedorovicius
Founder of Analytics Mania
julius@analyticsmania.com
@fedorovicius

Analytics mania

www.analyticsmania.com