

Learn about custom variables in Google Analytics

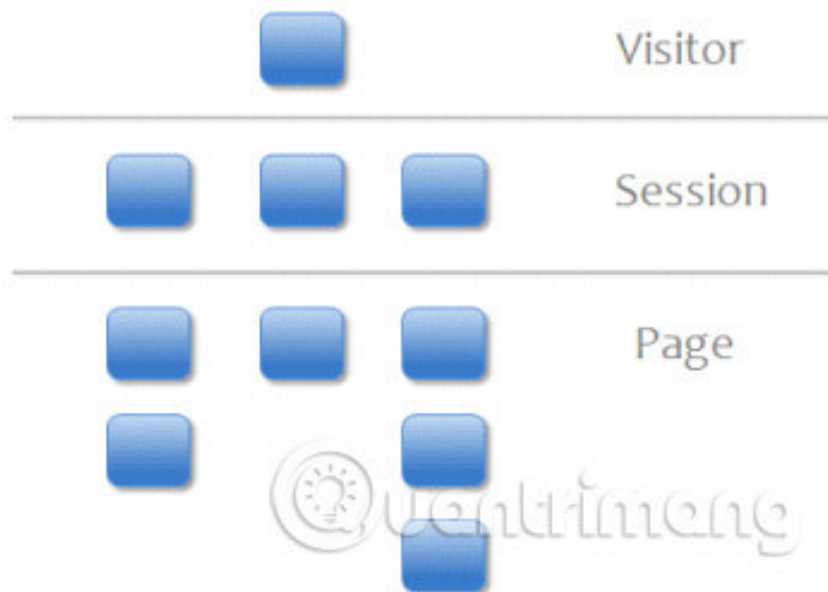
Custom variables are name-value pair tags (name-values) that you can insert into your code to refine Google Analytics tracking.

Custom variables are **name-value** pair tags (name-values) that you can insert into your code to refine Google Analytics tracking. With custom variables, you can define additional segments to apply to your visitors, in addition to the segments provided by Analytics. This article describes custom variables and how you can set them up.

overview

You will make the most of custom variables if you understand the basic visitor interaction model used in Google Analytics. In this model, visitors interact with your content over a period of time and interaction with your site is broken down into a hierarchy.

The chart illustrates this model for a visitor to your site, where each block represents the number of sessions and the interaction from that particular user.



Each level in this model is defined as follows:

1. **Visitor** - visitors visit the site, such as on the browser or the mobile phone of that person.

2. **Session** - the period of time that visitors are active on the site.
3. **Page** - works on behalf of the user to send GIF requests to the Analytics server.

Each of these three interaction levels defines a visitor's specific scope of interaction. This difference is important for custom variables because each custom variable is limited to a specific scope. For example, you might want to know the number of sessions that visitors deleted an item from their shopping cart. For that particular case, you will define the custom variable at the Session level, leaving the entire session for that visitor to be flagged as one of the items deleted from the online shopping cart.

Use Custom Variable

Because you can set up many custom variables to track user activity in your site, you'll often create your own JavaScript widget to manage them. Your script will use the basic method to create custom variables as follows:

```
_setCustomVar (index, name, value, opt_scope)
```

This method accepts four parameters:

1. **index** - Location for custom variables (very necessary). This is a number with a value between 1 and 5. Custom variables can only be placed in one location and cannot be reused on different locations.
2. **name** - The name for the custom variable (very necessary). This is a string that identifies the custom variable and appears in the top-level Custom Variables report in Analytics reports.
3. **value** - Value for custom variable (very necessary). This is a string that is paired with a name. You can pair some values with custom variable names. The value appears in the tabular list of the user interface for the selected variable name. Usually, you will have two or more values for a given name. For example, you can define the definition for a custom variable name and provide two male and nu values.
4. **opt_scope** - Scope for custom variables (optional). As described above, the scope determines the level of user interaction with your site. It is a number that has a value that can be 1 (Visitor level), 2 (Session level) or 3 (Page level). When not defined, custom variable ranges are defaulted to Page level interaction.

The following code illustrates how you can set custom variables to track visits to items that users have deleted from their shopping cart. Here, the **_setCustomVar () method** is called immediately before the **_trackEvent () method**, to send in the GIF request from the **_trackEvent () method**. It uses the **Item Removed** name with the **Yes** value to determine the activity from the website user. In addition, you should also set the default custom variable for the **Items Removed** and **No**. This way, you will have visits to items that have been removed from the shopping cart, and the number of visits that did not include the item has been removed.

The code is not synchronized (recommended):

```
_gaq.push (['_ setCustomVar',
1, // Custom var là ??t ?? slot # 1. Required parameter.
, ?ã g? b? Items, // Tên tên ng??i dùng là m?t ki?u ph?n này cho ng??
i dùng ng??i dùng. Required parameter.
"Yes", // This value of the custom variable. Required parameter.
2 // Sets the scope to session-level. Optional parameter.
]);
_gaq.push (['_ trackEvent',
'Shopping', // category of activity
'Item Removal', // Action
]);
```

Traditional (ga.js) Snippet (Traditional Code):

```
pageTracker . _setCustomVar (
  1 , // Custom var là ??t ?? slot # 1. Required parameter.
  "Removed Items" , // Tên tên ?ã v?i m?t ki?u ki?u n?i dung cho ng??
  i dùng ng??i dùng. Required parameter.
  "Yes" , // This value of the custom variable. Required parameter.
  2 // Sets the scope to session-level. Optional parameter.
);
pageTracker . _trackEvent (
  "Shopping" , // category of activity
  "Item Removal" // Action
);
```

Once you've set up a custom variable, you can use the **`_deleteCustomVar (index)`** method to delete your custom variable.

Examples of cases using custom variables

Custom variables can be implemented in a variety of ways, depending on your site model and business needs. The following are examples of different use cases, in which each case illustrates a different level of scope.

Page-level custom variable

Use custom page level variables to determine the set of page-level activities of users in the site.

For example, suppose you manage the website for an online newspaper site, where visitors view many different articles. Although it is easy to determine which specific article is the most popular, you can now use custom variables to determine which parts of the most visited newspaper. This is done by setting the page-level custom variable for each article, where the section for that article is set as a custom variable. For example, you might have sections like Life & Style, Opinion and Business. You can set custom variables to track all your posts by section.

The code is not synchronized (recommended):

```
_gaq.push (['_ setCustomVar',
  1, // Custom var là ??t ?? slot # 1. Required parameter.
  'Section', // The top-level name for your online content categories. Required
  'Life & Style', // Sets the value of "Section" to "Life & Style" for this part.
  3 // Sets the scope to page-level. Optional parameter.
]);
```

Traditional code:

```
pageTracker._setCustomVar (
  1, // Custom var là ??t ?? slot # 1. Required parameter.
  "Section", // The top-level name for your online content categories. Required
  "Life & Style", // Sets the value of "Section" to "Life & Style" for this part.
  3 // Sets the scope to page-level. Optional parameter.
)
pageTracker._trackPageview ();
```

Let's continue with this example and suppose that you not only want to tag the part of a specific article, but also the subsection. For example, the **Life & Style section** of your newspaper may also have some extras, such as **Food & Drink, Fashion and Sports** . So, for a specific article, you can track both the main and the subsections. You can set additional custom variables to track all your posts by subsection.

Asynchronous code (recommended)

```
_gaq.push (['_ setCustomVar',  
2, // Custom var này ???c ??t ?? slot # 2. Required parameter.  
'Sub-Section', // The 2nd-level name for your online content categories. Requ  
'Fashion', // Sets the value of "Sub-section" to "Fashion" for this particular  
3 // Sets the scope to page-level. Optional parameter.  
]);
```

Traditional code:

```
pageTracker._setCustomVar (  
2, // Custom var này ???c ??t ?? slot # 2. Required parameter.  
"Sub-Section", // The 2nd-level name for your online content categories. Requ  
"Fashion", // Sets the value of "Sub-section" to "Fashion" for this particular  
3 // Sets the scope to page-level. Optional parameter.  
);  
pageTracker._trackPageview ();
```

In this example, you place two custom page-level variables simultaneously for a page. For any page, you can track up to five custom variables, each with a separate location. This means you can specify 3 additional custom variables on this same page. For all the posts on your site, you can set up a series of page-level custom variables to track them in multiple sections and subsections. For more information on how to properly use page level custom variables, see the Usage guidelines section below.

Variable Session-level customization

Use session-level custom variables to distinguish different visitor experiences in sessions.

For example, if your site provides users with the ability to log in, you can use the custom variable adjusted to the session level for the user login status. That way, you can segment visits by visits from logged-in members versus anonymous visitors.

Asynchronous code (recommended)

```
_gaq.push (['_ setCustomVar',  
1, // Custom var là ??t ?? slot # 1. Required parameter.  
,, ,, 'User Type', // Tên c?a tùy ch?n. Required parameter.  
'Member', // Sets the value of "User Type" to "Member" or "Visitor" depending o  
2 // Sets the scope to session-level. Optional parameter.  
]);
```

Traditional code:

```
pageTracker._setCustomVar (  
1, // Custom var là ??t ?? slot # 1. Required parameter.  
"User Type", // The name of the custom variable. Required parameter.
```

```

"Member", // Sets the value of "User Type" to "Member" or "Visitor" depending on
2 // Sets the scope to session-level. Optional parameter.
);
pageTracker._trackPageview ();

```

Suppose you want to track both types of users and whether there is a purchase action in a particular session. If it is assumed that each page provides the user with the ability to log in, you will need to set the number 1 position for the **User Type** variable and use another location for **Shopping Attempts** :

Recommended code:

```

_gaq.push (['_ setCustomVar',
2, // Custom var này ???c ??t ?? slot # 2. Required parameter.
'Th??ng m?i này,' // // tên c?a các bi?n khác. Required parameter.
'Yes', // The value of the custom variable. Required parameter.
// (you may set giá tr? này b?ng m?c ??nh)
2 // Sets the scope to session-level. Optional parameter.
]);

```

Traditional code:

```

pageTracker._setCustomVar (
2, // Custom var này ???c ??t ?? slot # 2. Required parameter.
"Shopping Attempts", // The name of the custom variable. Required parameter.
"Yes", // The value of the custom variable. Required parameter.
// (you may set giá tr? này b?ng m?c ??nh)
2 // Sets the scope to session-level. Optional parameter.
);
pageTracker._trackPageview ();

```

Visitor level custom variable

Use the custom visitor level variable to distinguish visitor categories across multiple sessions.

For example, if your site offers premium content to paid subscribers, you can set access-level custom variables to analyze which users are paying, at what level, and users. Which are using free services in the website. You can set this custom variable as a function once because the value will continue to exist throughout the visitor's cookie.

Asynchronous code (recommended)

```

_gaq.push (['_ setCustomVar',
1, // Custom var là ??t ?? slot # 1. Required parameter.
'Member Type', // Tên c?a các bi?n khác. Required parameter.
'Premium', // The value of the custom variable. Required parameter.
// (possible values ??might be Free, Bronze, Gold, and Platinum)
1 // Sets the scope to visitor-level. Optional parameter.
]);

```

Traditional code:

```

pageTracker._setCustomVar (
1, // Custom var l  t ?? slot # 1. Required parameter.
"Member Type", // The name of the custom variable. Required parameter.
"Premium", // The value of the custom variable. Required parameter.
// (possible values ??might be Free, Bronze, Gold, and Platinum)
1 // Sets the scope to visitor-level. Optional parameter.
);
pageTracker._trackPageview ();

```

Principles used

This section describes the differences between different types of custom variables and how to use them correctly.

Types of custom variables

The following table identifies the main characteristics of different types of variables. Note that there are certain limitations when using different variables on the same location.

Duration When placing the same position with other variables **Quantity** Allowing Page Level One page visit, event or transaction call. The last page-level variable called on a page is the variable applied to that page.

For any web property (set of pages), many page-level variables can be placed in the same location and limited only by the number of visits in a given session.

For any page, you can set up to five custom variables simultaneously.

Level Session Current visitor.

The last session variable called during the session will be used for that session.

For example, if **login = false** for position 1 at the beginning of the session and **login = true** for position 1 later, the session is set to **true** for **login**.

Pass any previous level page variable and be called in the same session.

For example, if position 1 is used for the first time for **category = sports** and then for **login = true** for a session, **category = sports** will not be recorded in that session.

For any web property, you can create multiple variables that customize a separate session level, which can be specified with a 128-character key-value limit.

For any particular user session, you can set up to five session-level variables.

Grant Visitor Current and all future sessions within the visitor's cookie. The final value set for a visitor is the value applied to the current and future sessions. For any web property, you can create up to five separate visitor grants.

Use caution when combining different types of variables

In general, different custom variables should not be used in the same location as it may lead to strange metric calculations.

When you use multiple page level, session and visitor customization variables for your web properties, you need to carefully identify reuse of locations. If the situation arises on your site where a custom page or session-level variable is used at the same location and at the same time, only the session-level variable (or page level) is recorded.

The following scenarios illustrate the combination of page level, session and visitor variables set by a user on the same browser. In each example, the position is indicated by a number in parentheses and **S:** denotes the range of the variable.

Case 1 - The last session variable has priority

Here, the last page re-uses the custom session level variable in zone 1 so that it receives the priority.

Visit 1 Page 1

(1) S: page-level

section=opinion **Page 2**

(1) S: session-level

login=true **Page 3**

(1) S: session-level

converted=true

The visit report will be:

1. # visits for section = opinion: 0
2. # visits for login = true: 0
3. #visits for converted = true: 1

Case 2 - Initial Visitor number variable will take precedence

Here, position 1 which is first modified to customize the visitor's visit 1 followed by the page-level custom variable in the counter 3. In this order, the visitor level variable does not override the page-level variable.

Visit 1 Page 1

(1) S: visitor-level

gender=male Visit 2 **Page 1**

(2) S: session-level

converted=false Visit 3 **Page 1**

(1) S: page-level

section=opinion

The visit report will be:

1. # visits for gender = male: 2
2. # visits for converted = false: 1
3. # visits for section = opinion: 1

Important note

1. Do not use duplicate key names on locations.
2. Set the previous `_setCustomVar ()` function to require viewing a GIF page or event.

In certain cases, this may not be possible and you need to place another `_trackPageview ()` request after setting the custom variable. This is usually only needed in situations where the user activates a custom variable at the session or visitor level, which cannot be combined with a call to page views, events, or e-commerce tracking.

1. Use position matrix to track bulk custom variables.

If you have complex monitoring requirements, where the combination of page-level variables and sessions may conflict, you should create a placement matrix to ensure that session-level variables do not accidentally exceed the level variables. page.

1. Consider using the **Event Tracking** feature for certain applications, instead of using custom variables.

For example, suppose you have an online music site and you want to track both the login session, the purchase session and the session of the music samples played. You should use the Event Tracking feature to keep track of how many times your music plays instead of using session level variables to achieve this. Here, you can use the 4th value parameter of the Event Tracking command to transfer session data from your own cookie.

1. Do not use session level variables to monitor behavior you can follow with page level variables.

For example, suppose you track the login status and the status of searching for purchases by session and your site has a "**Members' special**" page that you also want to track. Because the page-level custom variable will display the number of visits to that particular variable, you will have access to the 'Members' special' number included in it.

See more:

1. 5 best real-time web analytics tools (2018)
2. Which websites monitor website traffic you already know?
3. 3 perfect tools for web analytics statistics

You finished reading the article "**Learn about custom variables in Google Analytics**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.