



## Basic Setup & Configuration

### Initialization

Before using `analytics.js`, you need to include the Google Analytics tracking code on your website.

**Important:** The `analytics.js` library is deprecated. Consider migrating to `gtag.js` for the latest features and support.

Basic setup involves adding a script tag to your HTML:

```
<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-XXXX-Y', 'auto');
  ga('send', 'pageview');
</script>
```

Replace `'UA-XXXX-Y'` with your actual Google Analytics tracking ID.

## Page and Event Tracking

### Page Tracking

The most basic form of tracking is a page view. This is automatically triggered in the basic setup.

```
ga('send', 'pageview');
```

To track page views with a custom page path:

```
ga('send', 'pageview', '/my-new-page');
```

To track page views with a custom title:

```
ga('send', 'pageview', {'page': '/my-new-page', 'title': 'My New Page
Title'});
```

## Custom Dimensions & Metrics

### Setting Custom Dimensions

Custom dimensions allow you to track data specific to your site.

```
ga('set', 'dimension[n]', 'dimensionValue');
```

Replace `[n]` with the dimension index (1-200) and `'dimensionValue'` with the value.

Example:

```
ga('set', 'dimension1', 'premium');
ga('send', 'pageview');
```

Or send it directly with the pageview:

```
ga('send', 'pageview', { 'dimension1': 'My custom dimension' });
```

## User and Session Management

### Alternative Initialization

An alternative initialization method, potentially referencing a Measurement ID:

```
const measurementId = document
  .getElementById('ga-init')
  .getAttribute('data-ga-measurement-id');
window.dataLayer = window.dataLayer || [];
function gtag() {
  dataLayer.push(arguments); // eslint-disable-line
}
gtag('js', new Date());
gtag('config', measurementId);
```

**Note:** This seems to be a hybrid approach more aligned with `gtag.js`.

### Tracking Multiple Properties

To track multiple properties, create multiple trackers with different names.

```
ga('create', 'UA-XXXX-Y', 'auto', 'tracker1');
ga('create', 'UA-ZZZZ-Y', 'auto', 'tracker2');

ga('tracker1.send', 'pageview');
ga('tracker2.send', 'pageview');
```

### Event Tracking

Events are used to track user interactions with your content.

```
ga('send', 'event', 'category', 'action', 'label', value);
```

Example:

```
ga('send', 'event', 'Videos', 'Play', 'My Cool Video');
```

With a numeric value:

```
ga('send', 'event', 'Downloads', 'Click', 'PDF Guide', 10);
```

**Note:** `category` and `action` are required. `label` and `value` are optional.

### Setting Custom Metrics

Custom metrics are similar to dimensions, but they represent numerical data.

```
ga('set', 'metric[n]', metricValue);
```

Replace `[n]` with the metric index and `metricValue` with the numeric value.

Example:

```
ga('set', 'metric2', 150);
ga('send', 'pageview');
```

## Setting User ID

The User ID feature allows you to track users across devices and sessions.

```
ga('set', 'userId', 'USER_ID');
```

Replace `'USER_ID'` with the user's unique ID. Must be consistent across devices.

Or during tracker creation:

```
ga('create', 'UA-XXXX-Y', { userId: 'USER_ID' });
```

## Session Control

You can manually start a new session:

```
ga('send', 'pageview', {'sessionControl': 'start'});
```

## Advanced Features & Considerations

### Exception Tracking

Track exceptions (errors) that occur on your website.

```
ga('send', 'exception', {
  'exDescription': 'DatabaseError', //
  Description of the exception
  'exFatal': false, //
  Whether the exception was fatal
  'appName': 'myapp', //
  Application name (optional)
  'appVersion': '0.1.2' //
  Application version (optional)
});
```

### Timing

Track how long it takes for certain tasks to complete (e.g., loading resources).

```
ga('send', 'timing', 'category', 'variable',
  value, 'label');
```

`category`: The category of the timing event.

`variable`: The name of the variable being timed.

`value`: The number of milliseconds.

`label`: An optional label.

### Ecommerce Tracking

**Note:** Ecommerce tracking with `analytics.js` is complex and involves pushing data to the `dataLayer`. Due to the deprecation of `analytics.js`, it's highly recommended to use `gtag.js` and its dedicated ecommerce features instead. Refer to Google Analytics documentation for gtag.js ecommerce tracking.

### Important Considerations (Deprecation)

The `analytics.js` library is deprecated. While the code snippets here may still function, they will not receive updates or new features. Migrating to `gtag.js` is strongly advised for continued support and access to the latest Google Analytics capabilities.