

The Pinterest tag implementation guide

1. The Pinterest tag events
 - 1.1. Events
 - 1.2. Event data
2. The Pinterest tag implementation
 - 2.1. Base code
 - 2.2. Event code
 - 2.3. Event data in Javascript
 - 2.4. Event data in the tag
 - 2.5. Callback
3. Conversion reporting
 - 3.1. Value and quantity

1. The Pinterest tag events

The Pinterest tag allows you to track specific actions, or events, on any pages on your website. Also, with the Pinterest tag, you can attach event data on every event to gather information on visitors' behaviors on your website. You can then use this information to create and build well-defined audiences for your ad campaigns on Pinterest.

1.1. Events

The table below shows the events that you can track with the Pinterest tag. Besides what's shown in the table, you can create and name your own events (see *[Partner-defined event]*). All events in the Pinterest tag—including your own—can be used to create audiences. However, if you want to track events for conversion reporting, you need to use our pre-defined events shown in the table below .

Event	Purpose	Audience creation	Conversion reporting
PageVisit <i>current tag type</i>	Track people who view primary pages, such as product pages and article pages	✓	✓
ViewCategory	Track people who view category pages	✓	✓
Search	Track people who perform searches on your website to look for specific products or store locations	✓	✓
AddToCart	Track people who add items to shopping carts	✓	✓

Checkout <i>current tag type</i>	Track people who complete transactions	✓	✓
WatchVideo	Track people who watch videos	✓	✓
Signup <i>current tag type</i>	Track people who sign up for your product or service	✓	✓
Lead	Track people who show interest in your product or service	✓	✓
Custom <i>current tag type</i>	Track a custom event. Use this event name to track a special event that you want to include in your conversion reporting	✓	✓
[Partner-defined event]	Track a special event that you've defined for the purpose of audience targeting. If you create your own event name, you won't be able to track this event in your conversion reporting.	✓	

The examples below show how you can specify events on your website using the Pinterest tag. More details can be found later in the section 2.3. [Event data in Javascript](#).

PageVisit

```
<script>
  pintrk('track', 'pagevisit', {
    promo_code: 'WINTER10'
  });
</script>
```

Signup

```
<script>
  pintrk('track', 'signup', {
    lead_type: 'New release promotion'
  });
</script>
```

1.2. Event data

For every event in the Pinterest tag, you can attach event data to provide more details on the event. We recommend you include `value` and `currency` for each event at a minimum. We use the values from these fields to calculate your *Return on Ad Spend* (ROAS) on Pinterest. In the future, we'll be able to optimize ad serving to maximize your ROAS across campaigns.

In addition to what's listed in the table below, you can create your own set of event data fields and define their usages or purposes according to your needs. However, keep in mind that if you define your own event

data, you can only use them for audience targeting; your custom event data will not be tracked for conversion reporting.

Event data	Sample value	Audience creation	Conversion reporting
property	Athleta (For a brand with mult-property)	✓	
search_query	boots	✓	
product_name*	Parker Boots	✓	
product_id*	1414	✓	
product_category*	Shoes	✓	
product_variant_id*	1414-Red	✓	
product_variant*	Red	✓	
product_price*	99.99	✓	
product_quantity*	1	✓	
product_brand*	Parker	✓	
order_quantity	1	✓	✓
order_id	X-151481	✓	
promo_code	WINTER10	✓	
value	100.00	✓	✓
currency	USD	✓	
video_title	<i>How to style your Parker Boots</i>	✓	
lead_type	Newsletter	✓	

* means the event data is in a `line_items` array

The example below shows a `Checkout` event that includes event data containing product information. More details can be found later in the section 2.3. [Event data in Javascript](#).

```
<script>
  pintrk('track', 'checkout', {
    value: 48,
    order_quantity: 1,
```

```

currency: 'USD',
line_items: [
  {
    product_name: 'Pillows (Set of 2)',
    product_id: '11',
    product_price: 48.00,
    product_quantity: 1
  }
]
});
</script>

```

2. The Pinterest tag implementation

The Pinterest tag has two components:

- The base code that you place on every page where you want to track visitors or conversion events
- The event code that you place on selected pages in addition to the base code*

You can use the event code on various pages of your website to:

- Track specific events
- Set up reporting on those specific events
- Collect detailed event data, which allow you to refine audiences for ad campaigns

* The base code **must** appear only once on a page and **must** run before running the event code.

2.1. Base code

You need to place the base code on a page where you want to track conversion events. Later when you create an audience for ad campaigns on Pinterest, you can include those who have visited these page into the audience.

Be sure to place the base code between the <head> and </head> tags in HTML.

Base code template

```

<script type="text/javascript">
!function(e){if(!window.pintrk){window.pintrk=function(){window.pintrk.queue.push(Array.prototype.slice.call(arguments))};var
n=window.pintrk;n.queue=[],n.version="3.0";var
t=document.createElement("script");t.async=!0,t.src=e;var
r=document.getElementsByTagName("script")[0];r.parentNode.insertBefore(t,r)}}("https://s.pini
ng.com/ct/core.js");
pintrk('load', 'YourTagID');

```

```
pintrk('page', {
  page_name: 'My Page',
  page_category: 'My Page Category'
});
</script>
<noscript>
  
</noscript>
```

In practice, you should retrieve your personalized code from our UI, or API. When you obtain your base code from our UI, or API, it will have your *Pinterest Tag ID** inserted in the proper locations of the code. To complete the customization of the base code, you will need to replace `'My Page'` and `'My Page Category'` with the descriptions of your page.

* Your *Pinterest Tag ID* will be different from any existing conversion tag IDs.

2.2. Event code

For every specific event you want to track on your website, you need to use the event code to attach detailed information about the event, such as the event type, order value, order quantity, currency, etc.

Event code placement

The event code **must** run after the base code has run on the page. Typically, this means that the event code should be placed somewhere after the base code in HTML. However, the exact placement of the event code will depend on how the event is triggered on the page.

If the loading of a webpage itself indicates that an event is triggered, then you can simply place the event code between the `<head>` and `</head>` tags after the base code in HTML. For example, when your website loads a checkout confirmation page, we're certain that a checkout has occurred. In this case, you should place the event code between the `<head>` and `</head>` tags, after the base code, on your checkout confirmation page. This placement ensures that the event code runs as soon as the page is loaded.

On the other hand, if an event is triggered only after a user takes a specific action on the page, such as clicking a button or submitting a form, then you need to make sure that the event code runs only as a result of the action. The latter case requires understanding of the existing event handlers in your page's Javascript code.

Event code template

See below for our event code template. At a minimum, you need to specify an event type in the event code; notice the event type parameter `'checkout'` in the function `pintrk()`.

```
<script>
  pintrk('track', 'checkout');
```

```
</script>
<noscript>
  
</noscript>
```

If you copy-and-paste this code template onto your page, you need to adjust the event type in two places: in the Javascript code (between `<script>` and `</script>` tags) and in the `` tag*.

- In Javascript, change the parameter `'checkout'` in `pintrk()` to your own event type
- In the `` tag, edit the parameter `event=checkout` and replace `YourTagID` with your *Pinterest Tag ID*

* The `` tag runs only when visitors to your page have disabled Javascript on their browsers, which, however, is not common.

2.3. Event data in Javascript

You can provide additional information about an event by attaching an object that contains event data. The event code below tracks a `checkout` event with event data describing a purchase of two products:

```
<script>
pintrk('track', 'checkout', {
  value: 10.00,
  order_quantity: 2,
  currency: 'USD',
  line_items: [
    {
      product_name: 'Parker Boots',
      product_id: '1414',
      product_price: 5.00,
      product_quantity: 1
    },
    {
      product_name: 'Parker Sandals'
      product_id: 'ABC',
      product_price: 5.00,
      product_quantity: 1
    }
  ]
});
</script>
```

Possible fields

The object below displays the names of all the available event data fields. Although every field is optional, the more information you provide, the more details you can work with later when you create audiences for ad campaigns.

```
{
  value: string,
  order_quantity: number,
  currency: string,
  property: string,
  search_query: string,
  order_id: string,
  promo_code: string,
  video_title: string,
  lead_type: string,
  line_items: [
    {
      product_name: string,
      product_id: string,
      product_category: string,
      product_variant_id: string,
      product_variant: string,
      product_price: string,
      product_quantity: number,
      product_brand: string
    }
  ]
}
```

Note: `line_items` is an array of objects where each object in the array contains details about a product.

Event data examples

The examples below showcase the event code with various events and their event data. Keep in mind that you can include a different combination of event data per event besides what's shown in the examples below.

PageVisit

```
<script>
  pintrk('track', 'pagevisit', {
    promo_code: 'WINTER10'
  });
</script>
```

Signup

```
<script>
  pintrk('track', 'signup', {
    lead_type: 'New release promotion'
  });
</script>
```

Checkout

```
<script>
  pintrk('track', 'checkout', {
    value: 116,
    order_quantity: 2,
    currency: 'USD',
    line_items: [
      {
        product_name: 'Pillows (Set of 2)',
        product_id: '11',
        product_price: 48.00,
        product_quantity: 1
      },
      {
        product_name: 'Pillows, Large (Set of 2)',
        product_id: '15',
        product_price: 68.00,
        product_quantity: 1
      }
    ]
  });
</script>
```

AddToCart

```
<script>
  pintrk('track', 'addtocart', {
    value: 499,
    order_quantity: 1,
    currency: 'USD',
    line_items: [
      {
        product_name: 'Red leather boots',
        product_id: '3486',
        product_category: 'shoe',
        product_variant_id: 'JB11103000',
        product_price: 499.00,
        product_quantity: 1,
      }
    ]
  });
</script>
```

```
        product_brand: 'My brand'
      }
    ]
  });
</script>
```

WatchVideo

```
<script>
  pintrk('track', 'watchvideo', {
    video_title: 'My Product Video 01'
  });
</script>
```

Lead

```
<script>
  pintrk('track', 'lead', {
    lead_type: 'Newsletter'
  });
</script>
```

2.4. Event data in the tag

When you copy-and-paste the event code template, in the section [2.2. Event code](#), onto your page, given that the majority of users will have Javascript enabled on their browsers, you may decide to leave the `<noscript>` tag as is—that is, without adding event data details. However, if you choose to, you can also include event data in the `` tag as shown in the example below:

```
<noscript>

</noscript>
```

The code below shows what the `` tag above looks like when we separate the data parameters for better readability. However, be aware that adding whitespace in the `src` attribute **will break** the code.

```
<noscript>

</noscript>
```

2.5. Callback

In addition to the event data, you can pass a Javascript callback function as an optional argument to the function `pintrk()`. This callback function runs every time when an event tracking call, the function `pintrk()`, is triggered.

The callback function takes the following two arguments:

```
function (didInit, error) {}
```

- `didInit`: This first argument receives a boolean value, which is set to be true if the event tracking call is successfully constructed, and false if an error is detected while making the event tracking call.
- `error`: This second argument receives a string, which describes the error when `didInit` returns false; if `didInit` is true, this string is undefined

Example

```
<script>
pintrk('track', 'checkout', {
  value: 10.00,
  order_quantity: 2,
  currency: 'USD',
  line_items: [
    {
      product_name: 'Parker Boots',
      product_id: '1414',
      product_price: 5.00,
      product_quantity: 1
    },
    {
      product_name: 'Parker Sandals'
      product_id: 'ABC',
      product_price: 5.00,
```

```

        product_quantity: 1
      }
    ]
  }, function(didInit, error) { if (!didInit) { console.log(error); }
});
</script>

```

3. Conversion reporting

3.1. Value and quantity

Here are the rules for calculating the value and quantity in conversion reporting from the Pinterest tag:

- If `order_quantity` is specified at the top level of event data, we read the field directly to obtain the total quantity
 - If not, we calculate the total quantity by summing up `product_quantity` from all items in `line_items`
 - If `product_quantity` is not specified for an item in `line_items`, we assume that the item's quantity is 1
- If `value` is specified at the top level of event data, we read the field directly to obtain the total value
 - If not, we calculate the total value by summing up the values from all items in `line_items` where a value of each item is calculated by multiplying an item's `product_price` by its `product_quantity`
 - If `product_price` is not specified for an item, we do **not** include the item in the calculation

Examples

```

{
  value: 10.00,           // Read this value
  order_quantity: 5,     // Read this quantity
  line_items: [ // No need for calculation ]
}

```

Reported value: 10.00

Reported quantity: 5

```

{
  // Notice neither 'value' nor 'order_quantity' is specified here
  line_items: [
    {
      product_name: "My Product Name A",
      product_price: 0.50
      // No product_quantity is specified. Assume 'product_quantity' is 1.
    }
  ]
}

```

```
    },  
    {  
      product_name: "My Product Name B",  
      product_quantity: 2,  
      product_price: 10.00  
    },  
    {  
      product_name: "My Product Name C",  
      product_quantity: 3  
      // No product_price is specified. Ignore this item.  
    }  
  ]  
}
```

Reported value: $(0.50 \times 1) + (10.00 \times 2) = 20.50$

Reported quantity: $1 + 2 + 3 = 6$