

错过了今年**Chrome Dev**峰会的举措？在**YouTube**上赶上我们的播放列表。[现在观看。](https://www.youtube.com/playlist?list=PLNYkxOF6rcICUD5nBfRdAR6Fveosnqa5m)  
(<https://www.youtube.com/playlist?list=PLNYkxOF6rcICUD5nBfRdAR6Fveosnqa5m>)

# 用断点暂停你的代码



**通过 凯斯·巴斯克斯**

(<https://developers.google.com/web/resources/contributors#kaycebasques>)

(<https://developers.google.com/web/resources/contributors#kaycebasques>)

Google的技术撰稿人

使用断点来暂停JavaScript代码。本指南解释了DevTools中可用的每种类型的断点，以及何时使用以及如何设置每种类型。有关调试过程的实践教程，请参阅在[Chrome DevTools](https://developers.google.com/web/tools/chrome-devtools/javascript/)中调试JavaScript的入门 (<https://developers.google.com/web/tools/chrome-devtools/javascript/>)。

## 概述何时使用每个断点类型

最著名的断点类型是代码行。但是，如果您不知道在哪里查找，或者您正在处理大型代码库，则可能无法设置代码行断点。通过知道如何以及何时使用其他类型的断点，您可以节省调试时间。

| 断点类型                            | 当您想要暂停时使用此...                          |
|---------------------------------|--|
| <u>行的代码</u> (#loc)              | 在一个确切的代码区域。                            |
| <u>条件行代码</u> (#conditional-loc) | 在一个确切的代码区域，但只有在其他条件成立的情况下。             |
| <u>DOM</u> (#dom)               | 在更改或删除特定DOM节点或其子项的代码上。                 |
| <u>XHR</u> (#xhr)               | 当一个XHR URL包含一个字符串模式。                   |
| <u>事件监听器</u> (#event-listeners) | 在事件之后运行的代码，例如 <code>click</code> ，被触发。 |
| <u>例外</u> (#exceptions)         | 在引发捕获或未捕获异常的代码行上。                      |
| <u>功能</u> (#function)           | 每当一个特定的函数被调用。                          |

## 一行代码断点

当您知道需要调查的确切代码区域时，请使用代码行断点。在执行这行代码之前，DevTools 总是暂停。

在DevTools中设置一行代码断点：

1. 点击**Sources**选项卡。
2. 打开包含要打破的代码行的文件。
3. 去代码的行。
4. 行号的左边是行号列。点击它。一个蓝色的图标显示在行号列的顶部。

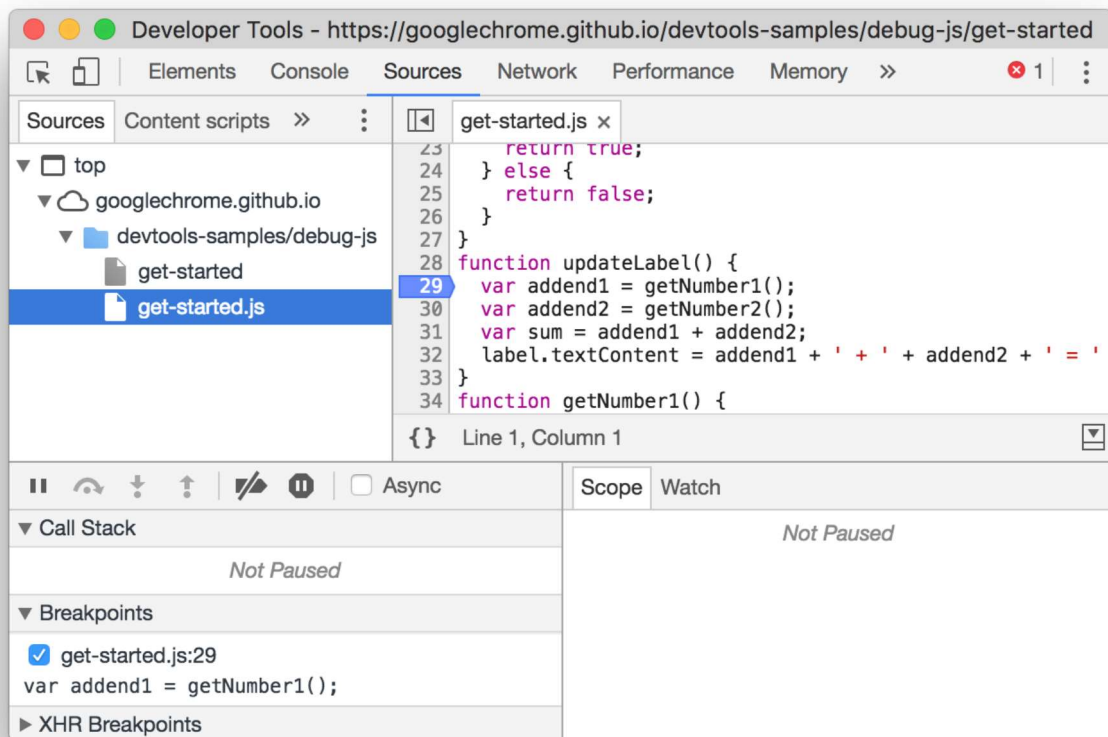


图1：第29行设置的一行代码断点

## 代码中的代码行断点

Call `debugger` from your code to pause on that line. This is equivalent to a line-of-code breakpoint (#loc), except that the breakpoint is set in your code, not in the DevTools UI.

```
console.log('a');  
console.log('b');
```

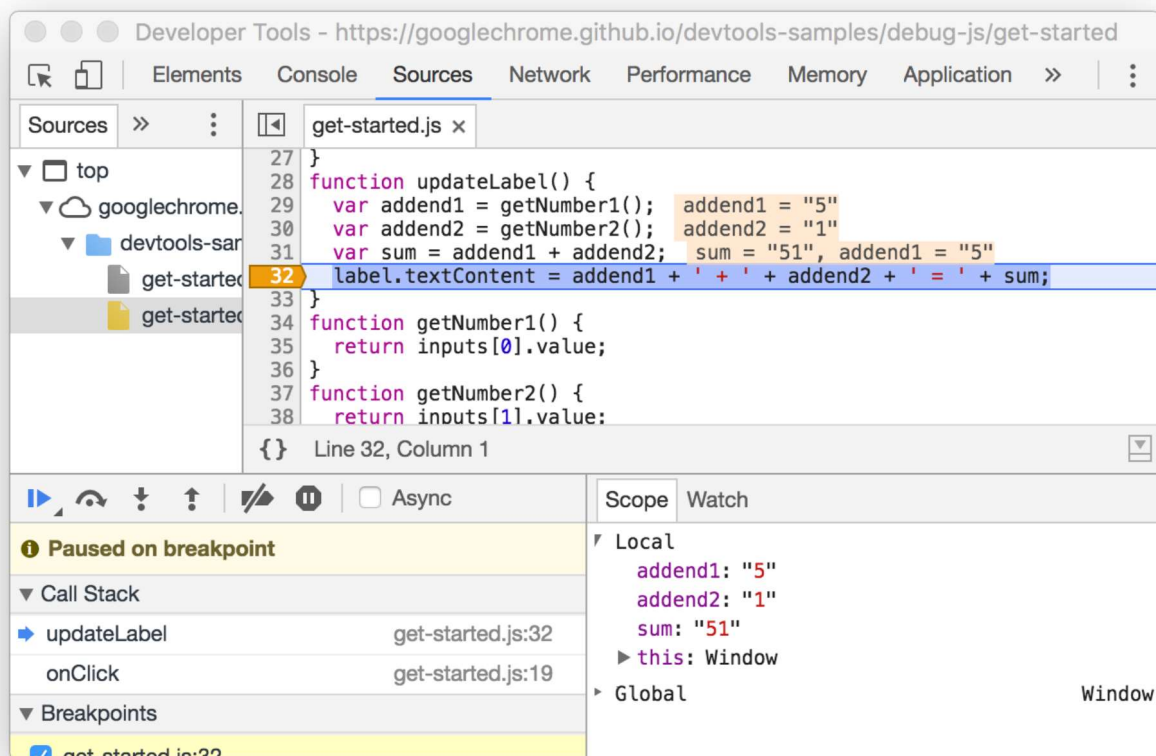
```
debugger;  
console.log('c');
```

## Conditional line-of-code breakpoints

Use a conditional line-of-code breakpoint when you know the exact region of code that you need to investigate, but you want to pause only when some other condition is true.

To set a conditional line-of-code breakpoint:

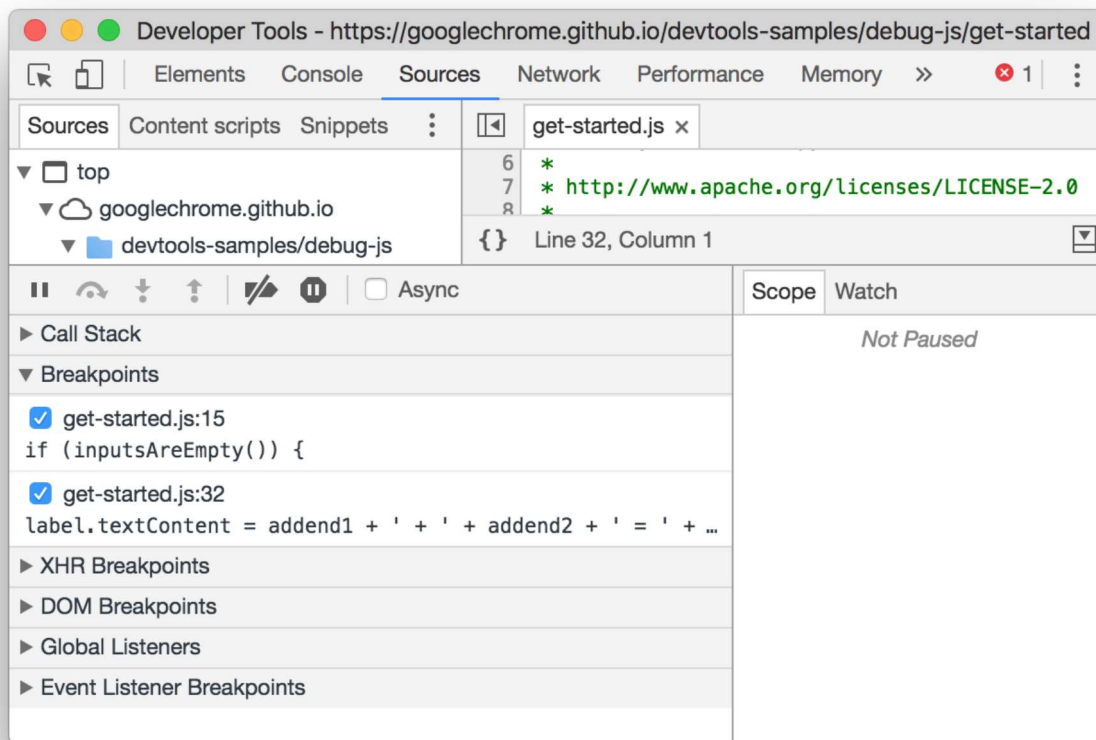
1. Click the **Sources** tab.
2. Open the file containing the line of code you want to break on.
3. Go the line of code.
4. To the left of the line of code is the line number column. Right-click it.
5. Select **Add conditional breakpoint**. A dialog displays underneath the line of code.
6. Enter your condition in the dialog.
7. Press **Enter** to activate the breakpoint. An orange icon appears on top of the line number column.



**Figure 2:** A conditional line-of-code breakpoint set on line 32

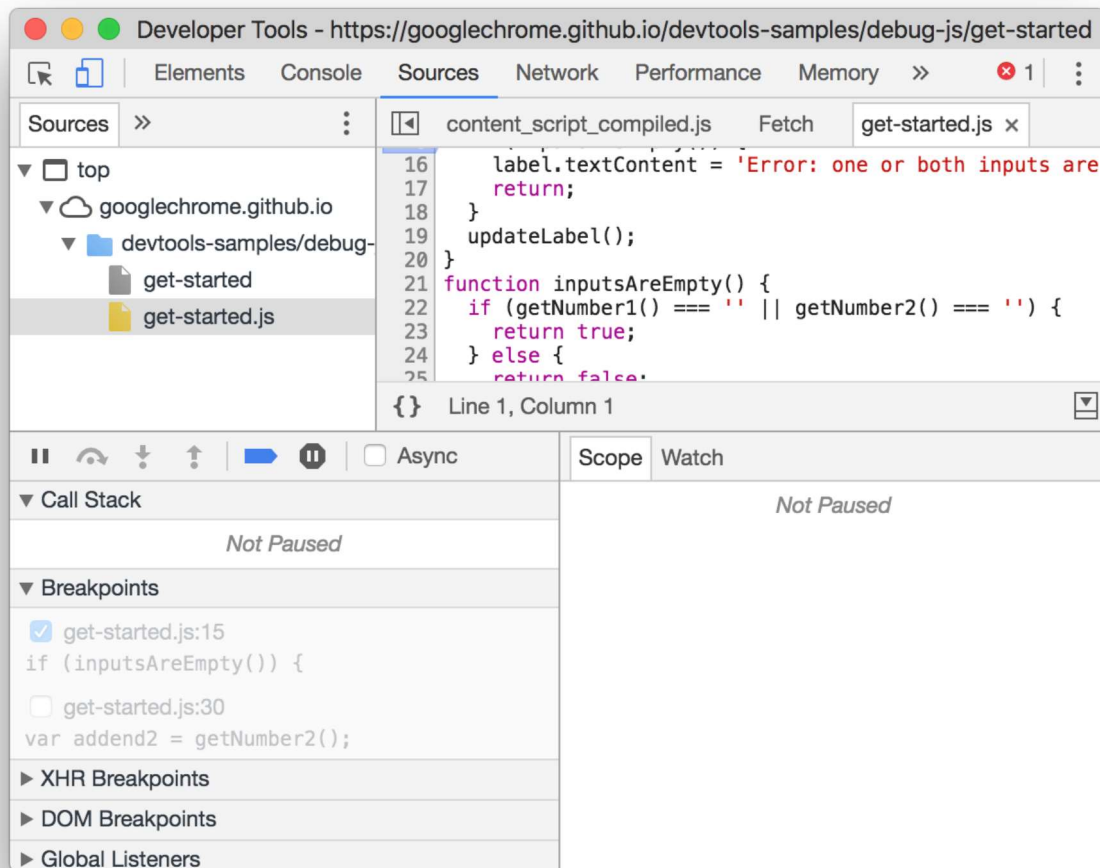
## Manage line-of-code breakpoints

Use the **Breakpoints** pane to disable or remove line-of-code breakpoints from a single location.



**Figure 3:** The **Breakpoints** pane showing two line-of-code breakpoints: one on line 15 of `get-started.js`, another on line 32

- Check the checkbox next to an entry to disable that breakpoint.
- Right-click an entry to remove that breakpoint.
- Right-click anywhere in the **Breakpoints** pane to deactivate all breakpoints, disable all breakpoints, or remove all breakpoints. Disabling all breakpoints is equivalent to unchecking each one. Deactivating all breakpoints instructs DevTools to ignore all line-of-code breakpoints, but to also maintain preserve their enabled state so that they are in the same state as before when you reactivate them.



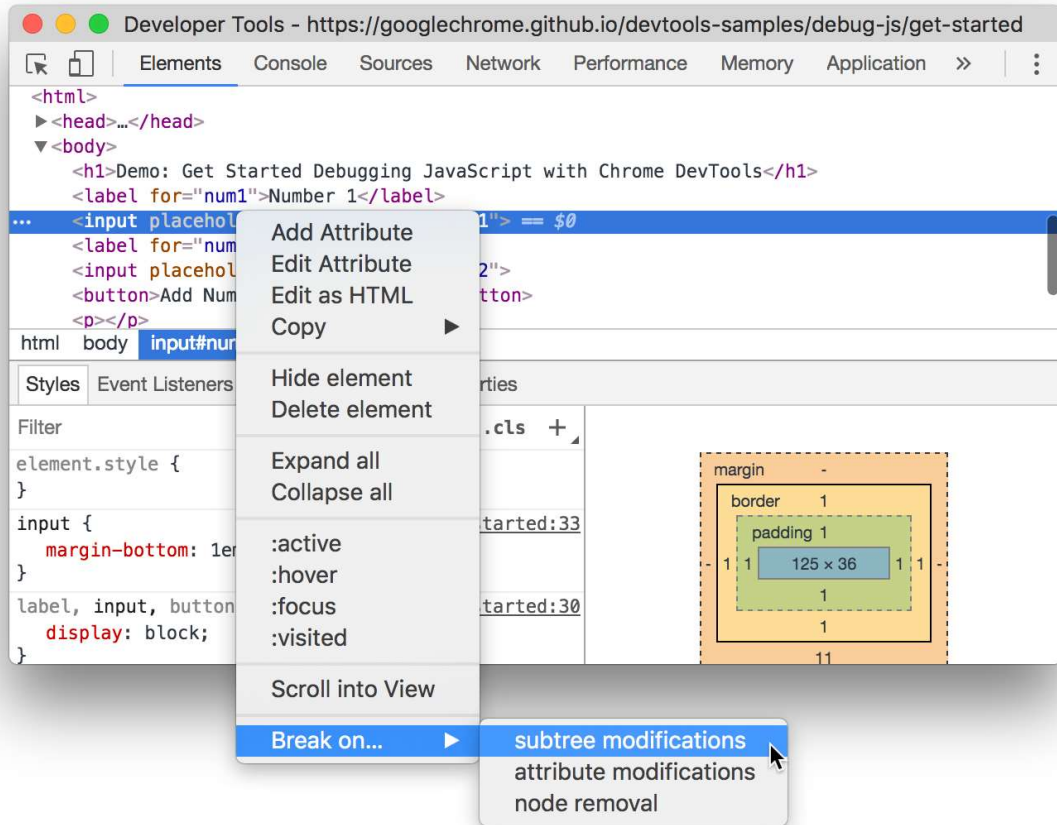
**Figure 4:** Deactivated breakpoints in the **Breakpoints** pane are disabled and transparent

## DOM change breakpoints

Use a DOM change breakpoint when you want to pause on the code that changes a DOM node or its children.

To set a DOM change breakpoint:

1. Click the **Elements** tab.
2. Go to the element that you want to set the breakpoint on.
3. Right-click the element.
4. Hover over **Break on** then select **Subtree modifications**, **Attribute modifications**, or **Node removal**.



**Figure 5:** The context menu for creating a DOM change breakpoint

## Types of DOM change breakpoints

- **Subtree modifications.** Triggered when a child of the currently-selected node is removed or added, or the contents of a child are changed. Not triggered on child node attribute changes, or on any changes to the currently-selected node.
- **Attributes modifications:** Triggered when an attribute is added or removed on the currently-selected node, or when an attribute value changes.
- **Node Removal:** Triggered when the currently-selected node is removed.

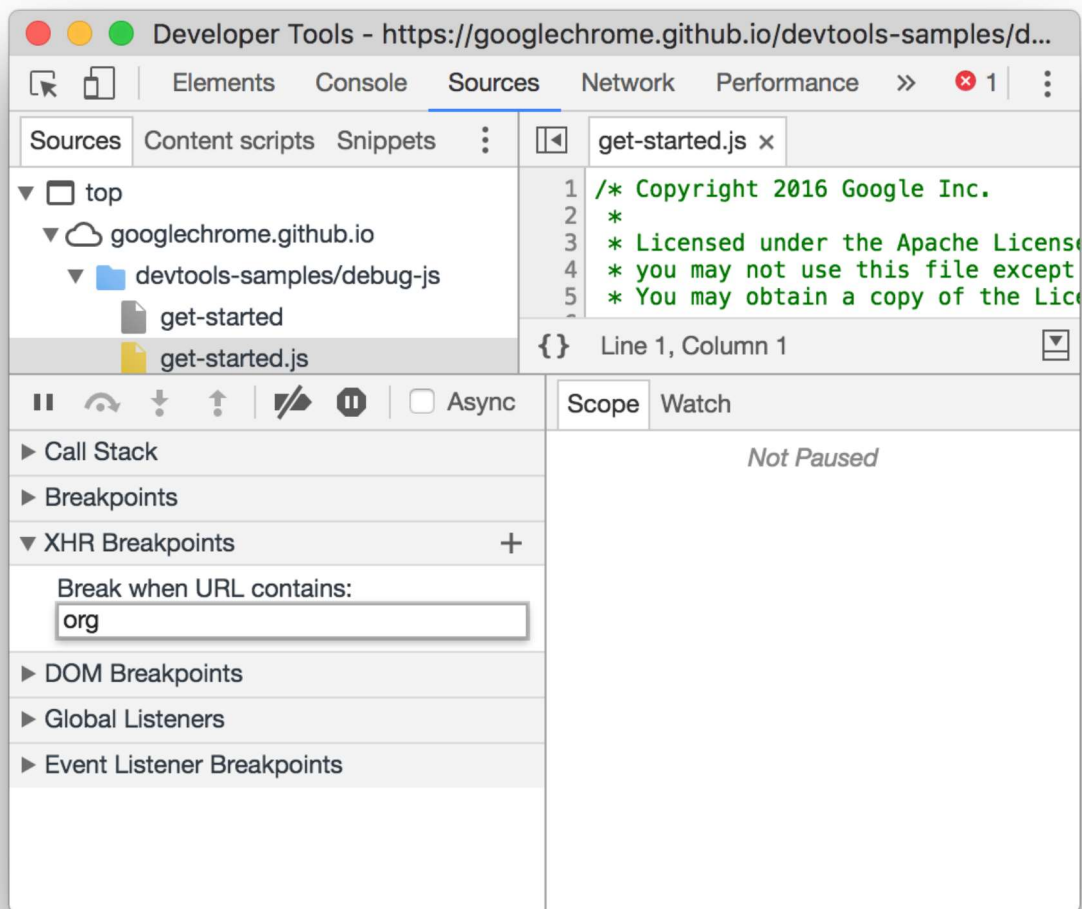
## XHR breakpoints

Use an XHR breakpoint when you want to break when the request URL of an XHR contains a specified string. DevTools pauses on the line of code where the XHR calls `send()`.

**Note:** This feature also works with [Fetch](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API) ([https://developer.mozilla.org/en-US/docs/Web/API/Fetch\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API)) requests.

To set an XHR breakpoint:

1. Click the **Sources** tab.
2. Expand the **XHR Breakpoints** pane.
3. Click **Add breakpoint**.
4. Enter the string which you want to break on. DevTools pauses when this string is present anywhere in an XHR's request URL.
5. Press `Enter` to confirm.



**Figure 6:** Creating an XHR breakpoint in the **XHR Breakpoints** for any request that contains `org` in the URL

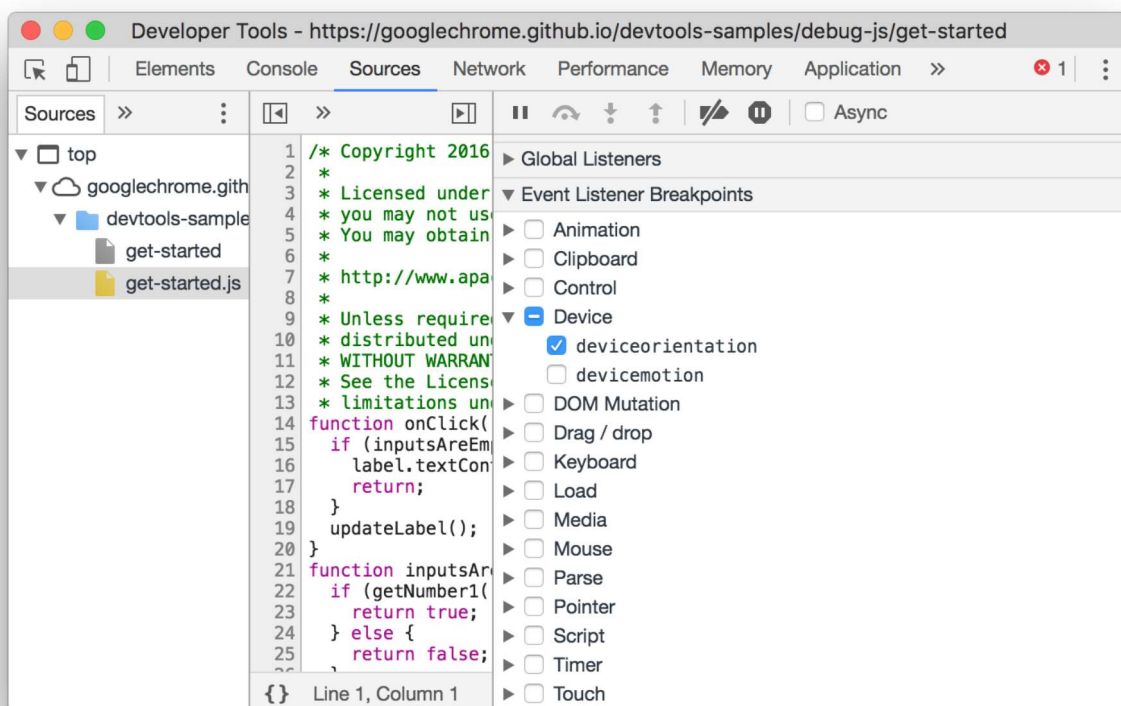
## Event listener breakpoints

---



Use event listener breakpoints when you want to pause on the event listener code that runs after an event is fired. You can select specific events, such as `click`, or categories of events, such as all mouse events.


1. Click the **Sources** tab.
2. Expand the **Event Listener Breakpoints** pane. DevTools shows a list of event categories, such as **Animation**.
3. Check one of these categories to pause whenever any event from that category is fired, or expand the category and check a specific event.



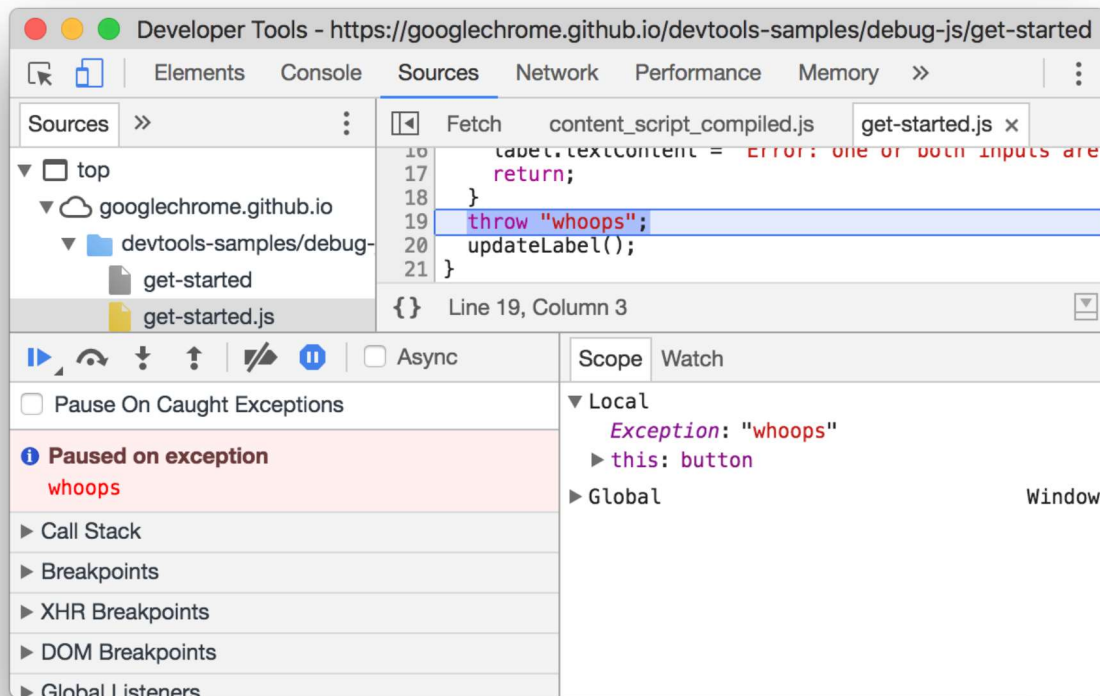
**Figure 7:** Creating an event listener breakpoint for `deviceorientation`

## Exception breakpoints

Use exception breakpoints when you want to pause on the line of code that's throwing a caught or uncaught exception.

1. Click the **Sources** tab.
2. Click **Pause on exceptions** . It turns blue when enabled.
3. (Optional) Check the **Pause On Caught Exceptions** checkbox if you also want to pause on caught exceptions, in addition to uncaught ones.





**Figure 7:** Paused on an uncaught exception

## Function breakpoints

Call `debug(functionName)`, where `functionName` is the function you want to debug, when you want to pause whenever a specific function is called. You can insert `debug()` into your code (like a `console.log()` statement) or call it from the DevTools Console. `debug()` is equivalent to setting a line-of-code breakpoint (`#loc`) on the first line of the function.

```
function sum(a, b) {
  let result = a + b; // DevTools pauses on this line.
  return result;
}
debug(sum); // Pass the function object, not a string.
sum();
```

## Make sure the target function is in scope

DevTools throws a `ReferenceError` if the function you want to debug is not in scope.

```
(function () {  
  function hey() {  
    console.log('hey');  
  }  
  function yo() {  
    console.log('yo');  
  }  
  debug(yo); // This works.  
  yo();  
})();  
debug(hey); // This doesn't work. hey() is out of scope.
```

Ensuring the target function is in scope can be tricky if you're calling `debug()` from the DevTools Console. Here's one strategy:

1. Set a line-of-code breakpoint (#loc) somewhere where the function is scope.
2. Trigger the breakpoint.
3. Call `debug()` in the DevTools Console while the code is still paused on your line-of-code breakpoint.

---

*Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](http://creativecommons.org/licenses/by/3.0/) (<http://creativecommons.org/licenses/by/3.0/>), and code samples are licensed under the [Apache 2.0 License](http://www.apache.org/licenses/LICENSE-2.0) (<http://www.apache.org/licenses/LICENSE-2.0>). For details, see our [Site Policies](https://developers.google.com/terms/site-policies) (<https://developers.google.com/terms/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.*

*Last updated 九月 26, 2017.*



### Chromium Blog

The latest news on the Chromium blog.



### GitHub

Fork our code samples and other open-source projects.



### Twitter

Connect with @ChromiumDev on Twitter.



### Videos

Check out our videos.



### Events

Attend a developer event and  
get hacking.