

[Home](#) > [puppeteer](#) > [Frame](#) > [waitForFunction](#)

## Frame.waitForFunction() method

**Signature:**

```
waitForFunction(pageFunction: Function | string, options?:  
FrameWaitForFunctionOptions, ...args: SerializableOrJSHandle[]): Promise<JSHandle>;
```

### Parameters

Parameter	Type	Description
pageFunction	Function   string	the function to evaluate in the frame context.
options	<a href="#">FrameWaitForFunctionOptions</a>	options to configure the polling method and timeout.
args	<a href="#">SerializableOrJSHandle</a> []	arguments to pass to the <code>pageFunction</code> .

**Returns:**

Promise<[JSHandle](#)>

the promise which resolve when the `pageFunction` returns a truthy value.

### Remarks

### Example

The `waitForFunction` can be used to observe viewport size change:

```
const puppeteer = require('puppeteer');  
  
(async () => {  
  . const browser = await puppeteer.launch();  
  . const page = await browser.newPage();  
  . const watchDog = page.mainFrame().waitForFunction('window.innerWidth < 100');  
  . page.setViewport({width: 50, height: 50});  
  . await watchDog;  
  . await browser.close();  
}) ();
```

To pass arguments from Node.js to the predicate of `page.waitForFunction` function:

```
const selector = '.foo';  
await frame.waitForFunction(  
  selector => !!document.querySelector(selector),  
  {}, // empty options object  
  selector  
);
```