

[Home](#) > [puppeteer](#) > [JSHandle](#)

## JSHandle class

Represents an in-page JavaScript object. JSHandles can be created with the [page.evaluateHandle](#) method.

### Signature:

```
export declare class JSHandle<HandleObjectType = unknown>
```

## Remarks

The constructor for this class is marked as internal. Third-party code should not call the constructor directly or create subclasses that extend the `JSHandle` class.

## Example

```
const windowHandle = await page.evaluateHandle(() => window);
```

JSHandle prevents the referenced JavaScript object from being garbage-collected unless the handle is [disposed](#). JSHandles are auto-disposed when their origin frame gets navigated or the parent context gets destroyed.

JSHandle instances can be used as arguments for [Page.\\$eval\(\)](#), [Page.evaluate\(\)](#), and [Page.evaluateHandle\(\)](#).

## Methods

Method	Modifiers	Description
<a href="#">asElement()</a>		
<a href="#">dispose()</a>		Stops referencing the element handle, and resolves when the object handle is successfully disposed of.
<a href="#">evaluate(pageFunction, args)</a>		This method passes this handle as the first argument to <code>pageFunction</code> . If <code>pageFunction</code> returns a Promise, then <code>handle.evaluate</code> would wait for the promise to resolve and return its value.
<a href="#">evaluateHandle(pageFunction, args)</a>		This method passes this handle as the first argument to <code>pageFunction</code> .
<a href="#">executionContext()</a>		Returns the execution context the handle belongs to.
<a href="#">getProperties()</a>		The method returns a map with property names as keys and JSHandle instances for the property values.
<a href="#">getProperty(propertyName)</a>		Fetches a single property from the referenced object.
<a href="#">jsonValue()</a>		
<a href="#">toString()</a>		Returns a string representation of the JSHandle.