JSHandle

JSHandle represents an in-page JavaScript object. JSHandles can be created with the page.evaluate_handle() method.

Sync Async

```
window_handle = page.evaluate_handle("window")
# ...
```

JSHandle prevents the referenced JavaScript object being garbage collected unless the handle is exposed with js_handle.dispose(). JSHandles are auto-disposed when their origin frame gets navigated or the parent context gets destroyed.

JSHandle instances can be used as an argument in page.eval_on_selector(), page.evaluate() and page.evaluate_handle() methods.

Methods

dispose

Added in: v1.8

The `jsHandle.dispose` method stops referencing the element handle.

Usage

```
js_handle.dispose()
```

evaluate

Added in: v1.8

Returns the return value of `expression`.

This method passes this handle as the first argument to `expression`.

If `expression` returns a Promise, then `handle.evaluate` would wait for the promise to resolve and return its value.

Usage

Sync **Async**

```
tweet_handle = page.query_selector(".tweet .retweets")
assert tweet_handle.evaluate("node => node.innerText") == "10 retweets"
```

Arguments

- `expression` str

  JavaScript expression to be evaluated in the browser context. If the expression evaluates to a function, the function is automatically invoked.

- `arg` EvaluationArgument *(optional)*

  Optional argument to pass to `expression`.

Returns

- Serializable

evaluate_handle

Added in: v1.8

Returns the return value of `expression` as a JSHandle.

This method passes this handle as the first argument to `expression`.

The only difference between `jsHandle.evaluate` and `jsHandle.evaluateHandle` is that `jsHandle.evaluateHandle` returns JSHandle.

If the function passed to the `jsHandle.evaluateHandle` returns a Promise, then `jsHandle.evaluateHandle` would wait for the promise to resolve and return its value.

See page.evaluate_handle() for more details.

Usage

```
js_handle.evaluate_handle(expression)
js_handle.evaluate_handle(expression, **kwargs)
```

Arguments

- `expression` str

  JavaScript expression to be evaluated in the browser context. If the expression evaluates to a function, the function is automatically invoked.

- `arg` EvaluationArgument *(optional)*

  Optional argument to pass to `expression`.

Returns

- JSHandle

get_properties

Added in: v1.8

The method returns a map with **own property names** as keys and JSHandle instances for the property values.

Usage

Sync **Async**

```
handle = page.evaluate_handle("({ window, document })")
properties = handle.get_properties()
window_handle = properties.get("window")
```

```
document_handle = properties.get("document")
handle.dispose()
```

Returns

- [Map][str, JSHandle]

get_property

Added in: v1.8

Fetches a single property from the referenced object.

Usage

```
js_handle.get_property(property_name)
```

Arguments

- `property_name` str

  property to get

Returns

- JSHandle

json_value

Added in: v1.8

Returns a JSON representation of the object. If the object has a `toJSON` function, it **will not be called**.

> **NOTE**
>
> The method will return an empty JSON object if the referenced object is not stringifiable. It will throw an error if the object has circular references.

Usage

```
js_handle.json_value()
```

Returns

- Serializable

Properties

as_element

Added in: v1.8

Returns either `null` or the object handle itself, if the object handle is an instance of ElementHandle.

Usage

```
js_handle.as_element()
```

Returns

- NoneType|ElementHandle