

Home > puppeteer > Page > waitForSelector

## Page.waitForSelector() method

Wait for the **selector** to appear in page. If at the moment of calling the method the **selector** already exists, the method will return immediately. If the **selector** doesn't appear after the **timeout** milliseconds of waiting, the function will throw.

This method works across navigations:

```
const puppeteer = require('puppeteer');
(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  let currentURL;
  page
    .waitForSelector('img')
    .then(() => console.log('First URL with image: ' + currentURL));
  for (currentURL of [
    'https://example.com',
    'https://google.com',
    'https://bbc.com',
  ]) {
    await page.goto(currentURL);
  }
  await browser.close();
})();
```

Signature:

```
waitForSelector(selector: string, options?: {
  visible?: boolean;
  hidden?: boolean;
  timeout?: number;
}): Promise<ElementHandle | null>;
```

## Parameters

| Parameter | Type   | Description                          |
|-----------|--|--------------------------------------|
| selector  | string   | A selector of an element to wait for |
| options   | { visible?: boolean; hidden?: boolean; timeout?: number; } | Optional waiting parameters          |

Returns:

Promise<ElementHandle | null>

Promise which resolves when element specified by selector string is added to DOM. Resolves to **null** if waiting for hidden: **true** and selector is not found in DOM.

## Remarks

The optional Parameter in Arguments **options** are :

- **Visible:** A boolean wait for element to be present in DOM and to be visible, i.e. to not have **display: none** or **visibility: hidden** CSS properties. Defaults to **false**.
- **hidden:** wait for element to not be found in the DOM or to be hidden, i.e. have **display: none** or **visibility: hidden** CSS properties. Defaults to **false**.
- **timeout:** maximum time to wait for in milliseconds. Defaults to 30000 (30 seconds). Pass 0 to disable timeout. The default value can be changed by using the Page.setDefaultTimeout() method.