Console

The `console` object provides a set of methods to help you emit logs, warnings, and debug code.

All standard methods of the `console` API are present on the `console` object in Workers.

However, some methods are no ops — they can be called, and do not emit an error, but do not do anything. This ensures compatibility with libraries which may use these APIs.

The table below enumerates each method, and the extent to which it is supported in Workers.

All methods noted as "□ supported" have the following behavior:

- They will be written to the console in local dev (`npx wrangler@latest dev`)
- They will appear in real-time logs when tailing logs in the dashboard or running `wrangler tail`
- They will create entries in the `logs` field of Tail Worker events and Workers Trace Events.

  You can use Logpush to send Workers Trace Event Logs to a supported destination.

All methods noted as "◫ partial support" have the following behavior:

- In both production and local development the method can be safely called, but will do nothing (no op)
- In the Workers Playground, Quick Editor in the Workers dashboard, and remote preview mode (`wrangler dev --remote`) calling the method will behave as expected, print to the console, etc.

Refer to Logs for more information about debugging and adding logs to Workers.

Method	Behavior
`console.debug()`	□ supported
`console.error()`	□ supported
`console.info()`	□ supported
`console.log()`	□ supported
`console.warn()`	□ supported
`console.clear()`	◫ partial support
`console.count()`	◫ partial support
`console.group()`	◫ partial support

Method	Behavior
`console.table()`	□□ partial support
`console.trace()`	□□ partial support
`console.assert()`	○ no op
`console.countReset()`	○ no op
`console.dir()`	○ no op
`console.dirxml()`	○ no op
`console.groupCollapsed()`	○ no op
`console.groupEnd`	○ no op
`console.profile()`	○ no op
`console.profileEnd()`	○ no op
`console.time()`	○ no op
`console.timeEnd()`	○ no op
`console.timeLog()`	○ no op
`console.timeStamp()`	○ no op
`console.createTask()`	□□ Will throw an exception in production, but works in local dev, Quick Editor, and remote preview

---

□□□ #2
□ □□□ □□□ 2026-11-01 05:49:36 UTC
□ □□□ □□□ 2026-11-01 05:49:38 UTC