

JAVASCRIPT CONSOLE

? The browser console is an interactive command line where you can execute JavaScript code.

Most browsers offer keyboard shortcuts for easy access.

For Google Chrome:

mac command + option + j

windows / linux ctrl + shift + j

Not using Chrome? Check the interwebs. ☺

The `console` object offers a variety of methods for interacting with the browser's debugging console.

Popular methods include:

log

Any type of value can be logged.

```
input console.log("cactus", 42, true) output cactus 42 true
```

Print values to the console, followed by a newline.

> Need to print an array or object? Use: `console.table(...)`

> Want to emphasize your logs? Add custom styles: `const styles = 'color: orange;'`

> Prefer string substitution? No problem:

```
console.log('%c Wow!', styles)
```

```
input console.log("Hello, %s the cactus.", "Fred") output Hello, Fred the cactus.
```

Go ahead, try it!



The console is a great place for debugging. You can run arbitrary code on the fly with no overhead - just type & hit enter.

error

```
input console.error("Something went wrong...") output Something went wrong...
```

Print an error message to the console, followed by a newline.

* By default, errors are highlighted in red.

warning

```
input console.warn("I have a bad feeling...") output I have a bad feeling...
```

Print a warning message to the console, followed by a newline.

* By default, warnings are highlighted in yellow.

timing

Start a "stopwatch" to identify bottlenecks in your code.

```
input console.time("myTimer");
// do stuff
console.timeEnd("myTimer");
output myTimer: 10ms
```

grouping

> Printing multiple messages? Group relevant messages for improved readability:

```
console.group("ex")
//messages
console.groupEnd("ex")
```

clearing console.clear()

> Made a mess? You can clear the console or refresh the page.

testing

```
console.assert(cactus.name === "Fred")
```

← Alert an error if a condition is not met.