Browser Dialogs and Message Boxes

JavaScript applications raise modal dialogs and browser-native message boxes during playback. There are two sides to handle, and they are not the same thing:

- **Dialogs your script triggers** — `alert`, `confirm`, `prompt` and `print`, called from a script or from a CustomJS action.
- **Dialogs the application raises** — handled inside the scenario so playback can continue past them.

> **These calls block browser execution until the dialog is dismissed.** Never leave them in a production or CI script: the run will hang waiting for a person.

API reference

API	Signature	What it does
`alert`	`alert(String: message)`	Triggers a native alert. Handy as a debugging or logging call — use it to inspect what a variable holds during runtime.
`confirm`	`confirm(String: message)`	Triggers a native confirm. Call it from *CustomJS in action*, together with the confirm API.
`prompt`	`prompt(String: prompt, String: defaultValue)`	Triggers a native prompt. Call it from *CustomJS in action*.
`print`	`print()`	Triggers the native print dialog. Call it from *CustomJS in action*.

Example — alert as a runtime check

```
navigateTo("http://www.apple.com");
click(link("ac-gn-link ac-gn-link-iphone"));
click(heading4("typography-headline-reduced"));

var var1 = isVisible(heading4("typography-headline-reduced"));
alert(var1);

if (var1 == true)
{
    click(link("ac-gn-link ac-gn-link-watch"));
}
```

An `alert` left in a script stops playback until someone closes it. Remove it before the script is promoted.

Handling a message box raised by the application

Type	Behaviour
Alert	Informational dialog, usually dismissed with OK.
Confirm	Dialog that expects an accept or dismiss decision — the branch can change the page state.
Prompt	Dialog that may require controlled text before it can be accepted.

Validation workflow

- Identify the exact dialog text and the moment it appears during playback.
- Use the matching dialog function in the scenario script.
- Replay the scenario on the target browser and node to confirm the dialog is handled consistently.
- Assert the expected application result after the dialog is accepted or dismissed — the dialog itself is not the outcome.

> Confirm the dialog is generated by the application, not by the browser as a permission or security prompt. Permission prompts are browser-profile settings, not script steps, and `print` / navigation dialogs are outside the DOM.

Legacy content

Older Test Designer examples for message boxes were written against previous browser behaviour. Validate any legacy snippet in the current Scenario Editor runtime before reusing it, because native dialog handling depends on the browser and node version in use.

Related: Conditional Functions, Assert Functions.

Revision #2
Created 2026-11-20 17:07:08 UTC by AI_Mananger
Updated 2026-11-20 17:09:59 UTC by AI_Mananger