

Chrome Extension Messaging Cheat Sheet

1. Content Script → Background Script

javascript

```
// In content script
chrome.runtime.sendMessage({
  action: "getData",
  data: "some info"
}, (response) => {
  console.log("Background responded:", response);
});

// In background script
chrome.runtime.onMessage.addListener((message, sender, sendResponse) => {
  if (message.action === "getData") {
    sendResponse({status: "received", data: "processed"});
  }
});
```

2. Popup → Background Script

javascript

```
// In popup.js
chrome.runtime.sendMessage({type: "popupMessage", value: "hello"});

// In background.js (same listener as above)
```

3. Background → Content Script

javascript

```
// In background script
chrome.tabs.query({active: true, currentWindow: true}, (tabs) => {
  chrome.tabs.sendMessage(tabs[0].id, {
    action: "updateUI",
    data: newData
  });
});

// In content script
chrome.runtime.onMessage.addListener((message, sender, sendResponse) => {
  if (message.action === "updateUI") {
    // Update the page
  }
});
```

4. Popup → Content Script (via Background)

javascript

```
// In popup.js
chrome.tabs.query({active: true, currentWindow: true}, (tabs) => {
  chrome.tabs.sendMessage(tabs[0].id, {
    from: "popup",
    action: "doSomething"
  });
});

// In content script (same listener)
```

5. Long-lived Connections

javascript

```
// Establish connection
let port = chrome.runtime.connect({name: "content-script"});

// Send message
port.postMessage({action: "ping"});

// Listen for messages
port.onMessage.addListener((msg) => {
  console.log("Received:", msg);
});

// In background
chrome.runtime.onConnect.addListener((port) => {
  if (port.name === "content-script") {
    port.onMessage.addListener((msg) => {
      port.postMessage({response: "pong"});
    });
  }
});
```

Key Points:

- **chrome.runtime.sendMessage()** - Send to background/popup
- **chrome.tabs.sendMessage()** - Send to content script
- **chrome.runtime.onMessage** - Listen for messages
- Use **sendResponse** for immediate responses
- Use **connections** for ongoing communication