

Référence rapide Chrome DevTools

Éléments, Console, Réseau, Performances, débogage

Éléments

Inspection et édition

Clic droit → Inspecter	Ouvrir le panneau Éléments pour l'élément
Double-clic sur tag/attribut	Modifier le HTML en ligne
Touche Suppr	Supprimer le nœud sélectionné
Ctrl+Z	Annuler le changement DOM
Touche H	Basculer la visibilité de l'élément sélectionné
Glisser un nœud	Déplacer l'élément dans l'arbre DOM

Panneau Styles

element.style {}	Ajouter des styles inline à l'élément
Cliquer sur une valeur	Modifier la valeur CSS en direct
Case à cocher d'une règle	Activer/désactiver une propriété CSS
Bouton :hov	Forcer les pseudo-états (:hover, :focus)
Bouton .cls	Ajouter/supprimer des classes CSS
Échantillon de couleur	Ouvrir le sélecteur de couleur
Onglet Calculé	Voir les valeurs CSS calculées finales

Console

API Console

<code>console.log("info");</code>	<code>console.warn("warning");</code>
<code>console.error("error");</code>	<code>console.table(arrayObj);</code>
<code>console.group("label");</code>	<code>console.groupEnd();</code>
<code>console.time("t");</code>	<code>/*...*/ console.timeEnd("t");</code>

Utilitaires Console

\$0	Élément actuellement sélectionné dans Éléments
\$('.sel') / \$\$('sel')	Raccourci querySelector / querySelectorAll
copy(obj)	Copier l'objet en chaîne vers le presse-papiers
clear()	Effacer la sortie de la console
monitor(fn)	Journaliser les appels à la fonction fn
monitorEvents(el, 'click')	Journaliser les événements sur l'élément
keys(obj) / values(obj)	Clés / valeurs de l'objet
\$_	Résultat de la dernière expression évaluée

Filtrage

Liste déroulante niveaux de log	Filtrer par verbose/info/warn/error
Zone de filtre texte	Rechercher dans la sortie de la console
-url:extension	Exclure les messages par URL source
Liste déroulante context:	Filtrer par contexte iframe/worker

Réseau

Fonctionnalités du panneau Réseau

Barre de filtre	Filtrer par type : XHR, JS, CSS, Img, Doc, WS
Recherche (Ctrl+F)	Rechercher dans tous les corps de requête/réponse
Conserver le journal	Garder le journal lors de la navigation
Désactiver le cache	Contourner le cache navigateur avec DevTools ouvert
Liste déroulante limitation	Simuler 3G lent, 3G rapide, Hors ligne
Bloquer l'URL de requête	Clic droit → Bloquer URL

Onglets des détails de requête

En-têtes	En-têtes de requête/réponse, code de statut
Charge utile	Corps POST, paramètres de requête
Aperçu	Réponse formatée (JSON, HTML, image)
Réponse	Corps de réponse brut
Chronologie	DNS, connexion, TLS, TTFB, téléchargement
Initiateur	Trace de pile déclenchant la requête
Cookies	Cookies envoyés/reçus

Copier et exporter

Clic droit → Copier en cURL	Copier la requête comme commande cURL
Copier en fetch	Copier en JavaScript fetch()
Exporter HAR	Exporter toutes les requêtes en fichier HAR
Copier la réponse	Copier le corps de réponse dans le presse-papiers

Sources

Points d'arrêt

Clic sur numéro de ligne	Activer/désactiver un point d'arrêt
Clic droit → Conditionnel	S'arrêter uniquement si l'expression est vraie
Clic droit → Logpoint	Journaliser sans interrompre l'exécution
Point d'arrêt DOM	S'arrêter sur modification sous-arbre/attribut/suppression
Point d'arrêt XHR/Fetch	S'arrêter quand l'URL contient une chaîne
Point d'arrêt sur événement	S'arrêter sur des types d'événements spécifiques

Contrôles du débogueur

F8 / Ctrl+\	Reprendre / suspendre l'exécution
F10	Passer par-dessus l'appel de fonction
F11	Entrer dans l'appel de fonction
Shift+F11	Sortir de la fonction courante
Ctrl+Shift+P → "never pause"	Désactiver tous les points d'arrêt

Panneaux de débogage

Watch	Surveiller les valeurs d'expressions
Scope	Variables locales, fermetures, globales
Call Stack	Chaîne d'appels de fonctions
Snippets	Enregistrer et exécuter du code JS réutilisable

Performances

Enregistrement

Bouton Enregistrer (Ctrl+E)	Démarrer/arrêter l'enregistrement des performances
Bouton Recharger	Enregistrer les performances au chargement de page
Case Captures d'écran	Capturer la chronologie visuelle
Liste déroulante CPU	Simuler un ralentissement CPU 4x/6x
Limitation réseau	Simuler un réseau lent pendant l'enregistrement

Lire le graphe en flamme

Piste principale	Exécution JavaScript (graphe en flamme)
Piste Réseau	Chronologie des requêtes réseau
Piste Frames	FPS et durées de frames
Piste Timings	Marqueurs FCP, LCP, DCL, Load
Barres jaunes	JavaScript (scripting)
Barres violettes	Mise en page / rendu
Barres vertes	Peinture / composition

Bottom-Up et Résumé

Onglet Résumé	Répartition du temps : scripting, rendu, etc.
Onglet Bottom-Up	Fonctions les plus coûteuses en premier
Onglet Call Tree	Hiérarchie d'appels de la racine aux feuilles
Onglet Event Log	Liste chronologique des événements

Application

Stockage

Local Storage	Afficher/modifier les paires clé-valeur par origine
Session Storage	Afficher/modifier le stockage de session
IndexedDB	Inspecter les magasins d'objets et les enregistrements
Cookies	Afficher/modifier/supprimer les cookies par domaine
Cache Storage	Inspecter les caches Service Worker
Effacer le stockage	Effacer en masse les types de stockage sélectionnés

Service Workers et Manifest

Service Workers	Voir l'enregistrement, le statut, push/sync
Mettre à jour au rechargement	Forcer la mise à jour du SW à chaque rechargement
Contourner pour le réseau Manifest	Ignorer le SW et aller au réseau Voir les détails du manifeste de l'application web
Case Hors ligne	Simuler le mode hors ligne

Lighthouse

Lancer des audits

Mode : Navigation	Audit complet du chargement de page
Mode : Timespan	Auditer les interactions utilisateur dans le temps
Mode : Snapshot	Auditer l'état actuel de la page
Catégories	Performances, Accessibilité, Bonnes pratiques, SEO
Appareil	Simulation mobile ou bureau

Métriques clés

FCP (First Contentful Paint)	Temps jusqu'au premier contenu visible
LCP (Largest Contentful Paint)	Temps jusqu'au plus grand élément visible
TBT (Total Blocking Time)	Somme du temps de blocage des tâches longues
CLS (Cumulative Layout Shift)	Score de stabilité visuelle
SI (Speed Index)	Rapidité de remplissage visuel du contenu
INP (Interaction to Next Paint)	Réactivité aux entrées utilisateur

Référence rapide Chrome DevTools

Raccourcis

Ouvrir DevTools

F12 / Ctrl+Shift+I	Ouvrir/fermer DevTools
Ctrl+Shift+J	Ouvrir le panneau Console
Ctrl+Shift+C	Ouvrir Éléments + mode inspection
Ctrl+Shift+M	Basculer la barre d'outils d'appareil (responsive)

Navigation et recherche

Ctrl+Shift+P	Menu de commandes (toute action)
Ctrl+P	Ouvrir un fichier (Aller au fichier)
Ctrl+Shift+F	Rechercher dans toutes les sources
Ctrl+G	Aller au numéro de ligne dans Sources
Ctrl+[/ Ctrl+] 	Basculer entre les panneaux

Édition et Console

Ctrl+Shift+D	Basculer la position d'ancrage de DevTools
Ctrl+L (Console)	Effacer la sortie de la console
Shift+Entrée (Console)	Saisie multi-ligne
Échap	Basculer le tiroir console
Ctrl+K (Console)	Effacer la console

Débugage mobile

Barre d'outils d'appareil

Ctrl+Shift+M	Basculer la barre d'outils d'appareil
Liste déroulante appareil	Dimensions prédéfinies téléphone/tablette
Mode responsive	Redimensionner librement le viewport
Liste déroulante DPR	Changer le ratio de pixels
Limitation	Simuler CPU + réseau mobile
Afficher les media queries	Visualiser les points de rupture CSS

Débugage à distance (Android)

1. Activer le débogage USB	Paramètres → Options développeur sur l'appareil
2. Connecter USB	Connecter l'appareil Android à l'ordinateur
3. chrome://inspect	Ouvrir dans Chrome bureau
4. Cliquer Inspecter	Ouvrir DevTools pour la page mobile

Nécessite Chrome sur le bureau et l'appareil Android

Remplacements des capteurs

Géolocalisation	Remplacer les coordonnées GPS
Orientation	Simuler l'orientation de l'appareil
Tactile	Simuler les événements tactiles
Détection d'inactivité	Remplacer l'API de détection d'inactivité

Patterns courants

Déboguer les problèmes réseau

```
// Dans la Console : surveiller toutes les requêtes fetch
const origFetch = window.fetch;
window.fetch = (...args) => {
  console.log('fetch:', args);
  return origFetch(...args);
};
```

Workflow de performances

1. Audit Lighthouse	Identifier les principaux problèmes de performance
2. Enregistrement de performance	Trouver les tâches longues dans le graphe en flamme
3. Onglet Couverture	Trouver le CSS/JS inutilisé (Ctrl+Shift+P → Coverage)
4. Cascade réseau	Identifier les ressources bloquantes
5. Onglet Rendu	Visualiser la peinture/mise en page (Ctrl+Shift+P → Rendering)

Extraits de console utiles

```
// Lister tous les écouteurs d'événements sur un élément
getEventListeners($0);
```

```
// Surveiller les décalages de mise en page
new PerformanceObserver(l => l.getEntries().forEach(
  e => console.log('CLS:', e)
)).observe({ type: 'layout-shift', buffered: true });
```