

CHROME DEVTOLS 빠른 참조

Elements, Console, Network, Performance, 디버깅

Elements	
검사 & 편집	
Right-click -> Inspect	요소의 Elements 패널 열기
Double-click tag/attribute	HTML 인라인 편집
Delete key	선택한 노드 삭제
Ctrl+Z	DOM 변경 실행 취소
H key	선택한 요소 표시/숨기기 토글
Drag node	DOM 트리에서 요소 이동
Styles 패널	
element.style {}	요소에 인라인 스타일 추가
Click property value	CSS 값 라이브 편집
Checkbox next to rule	CSS 속성 켜기/끄기
hover button	요소 의사클래스 강제 적용 (hover, focus)
.cls button	CSS 클래스 추가/제거
Color swatch	색상 선택기 열기
Computed tab	최종 계산된 CSS 값 보기

Console	
Console API	
console.log("info");	console.warn("warning");
console.error("error");	console.table(arrayOrObj);
console.group("label");	console.groupEnd();
console.time("t");	/*...*/ console.timeEnd("t");

Console 유틸리티	
\$0	Elements에서 현재 선택된 요소
\$('sel') / \$\$ ('sel')	querySelector / querySelectorAll 단축
copy(obj)	객체를 문자열로 클립보드에 복사
clear()	콘솔 출력 지우기
monitor(fn)	함수 in 호출 로깅
monitorEvents(el, 'click')	요소의 이벤트 로깅
(keys(obj) / values(obj))	객체 키 / 값
\$	마지막 평가 표현식의 결과

필터링	
Log levels dropdown	verbose/info/warn/error로 필터
Filter text box	콘솔 출력 검색
-url:extension	소스 URL로 메시지 제외
context: dropdown	iframe/worker 컨텍스트로 필터

Network	
Network 패널 기능	
Filter bar	유형별 필터: XHR, JS, CSS, Img, Doc, WS
Search (Ctrl+F)	모든 요청/응답 본문 검색
Preserve log	페이지 이동 후에도 로그 유지
Disable cache	DevTools 열린 동안 브라우저 캐시 우회
Throttling dropdown	Slow 3G, Fast 3G, Offline 시뮬레이션
Block request URL	요청 URL릭 -> Block URL

요청 세부 탭	
Headers	요청/응답 헤더, 상태 코드
Payload	POST 본문, 쿼리 파라미터
Preview	서식 있는 응답 (JSON, HTML, 이미지)
Response	원시 응답 본문
Timing	DNS, 연결, TLS, TTFB, 다운로드
Initiator	요청을 트리거한 스택 트레이스
Cookies	전송/수신된 쿠키

복사 & 내보내기	
Right-click -> Copy as cURL	요청을 cURL 명령으로 복사
Copy as fetch	fetch() JavaScript로 복사
Export HAR	모든 요청을 HAR 파일로 내보내기

Copy response	
	응답 본문을 클립보드에 복사

Sources	
중단점	
Click line number	라인 중단점 토글
Right-click -> Conditional	표현식이 참일 때만 중단
Right-click -> Logpoint	실행 중단 없이 로깅
DOM breakpoint	서브트리/속성/제거 시 중단
XHR/Fetch breakpoint	URL에 문자열 포함 시 중단
Event listener breakpoint	특정 이벤트 유형 시 중단

디버거 컨트롤	
F8 / Ctrl+V	실행 재개 / 일시 중지
F10	다음 호출 시 호출 건너뛰기
F11	할수 호출 안으로 들어가기
Shift+F11	현재 할수에서 나오기
Ctrl+Shift+P -> "never pause"	모든 중단점 비활성화

디버거 패널	
Watch	표현식 값 모니터링
Scope	로컬, 글로벌, 지역 변수
Call Stack	함수 호출 체인
Snippets	재사용 가능한 JS 코드 저장 및 실행

Performance	
녹화	
Record button (Ctrl+E)	성능 녹화 시작/중지
Reload button	페이지 로드 후 성능 녹화
Screenshots checkbox	시각적 타임라인 캡처
CPU throttle dropdown	CPU 4x/6x 속도 저하 시뮬레이션
Network throttle	녹화 중 느린 네트워크 시뮬레이션

플래임 차트 읽기	
Main track	JavaScript 실행 (플래임 차트)
Network track	네트워크 요청 타임라인
Frames track	FPS 및 프레임 지속 시간

Timings track	FCP, LCP, DCL, Load 마커
Yellow bars	JavaScript (스크립팅)
Purple bars	레이아웃 / 렌더링
Green bars	페인팅 / 합성
Bottom-Up & Summary	
Summary tab	시간 범용, 스크리닝, 렌더링 등
Bottom-Up tab	가장 비용이 큰 함수 먼저
Call Tree tab	루트에서 리프까지 호출 계층
Event Log tab	시간순 이벤트 목록

Application	
스토리지	
Local Storage	오리진별 키-값 쌍 보기/편집
Session Storage	세션 범위 스토리지 보기/편집
IndexedDB	객체 스토어 및 레코드 검사
Cookies	도메인별 쿠키 보기/편집/삭제
Cache Storage	Service Worker 캐시 검사
Clear storage	선택한 스토리지 유형 일괄 삭제
Service Workers & Manifest	
Service Workers	등록 상태, push/sync 보기
Update on reload	리로드 시마다 SW 강제 업데이트
Bypass for network	SW 건너뛰고 네트워크로 직접 요청
Manifest	헬프 매니페스트 세부 정보 보기
Offline checkbox	오프라인 모드 시뮬레이션

Lighthouse	
감사 실행	
Mode: Navigation	전체 페이지 로드 감사
Mode: Timespan	시간 경과에 따른 사용자 상호작용 감사
Mode: Snapshot	현재 페이지 상태 감사
Categories	성능, 접근성, 모범 사례, SEO
Device	모바일 또는 데스크톱 시뮬레이션
핵심 지표	
FCP (First Contentful Paint)	첫 번째 콘텐츠 표시까지 걸린 시간
LCP (Largest Contentful Paint)	가장 큰 시각 요소 표시까지 걸린 시간
TBT (Total Blocking Time)	작업의 총 블로킹 시간 합
CLS (Cumulative Layout Shift)	시각적 안정성 점수
SI (Speed Index)	콘텐츠가 시각적으로 채워지는 속도
INP (Interaction to Next Paint)	사용자 입력에 대한 반응성

단축키	
DevTools 열기	
F12 / Ctrl+Shift+I	DevTools 열기/닫기 토글
Ctrl+Shift+J	Console 패널 열기
Ctrl+Shift+C	Elements + 검사 모드 열기
Ctrl+Shift+M	기기 톨바(반응형) 토글

탐색 & 검색	
Ctrl+Shift+P	명령 메뉴 (모든 작업 실행)
Ctrl+P	파일 열기 (Go to File)
Ctrl+Shift+F	모든 소스 검색
Ctrl+G	Sources에서 라인 번호 이동
Ctrl+[/ Ctrl+]	패널 간 전환

편집 & Console	
Ctrl+Shift+D	DevTools 도킹 위치 토글
Ctrl+L (Console)	콘솔 출력 지우기
Shift+Enter (Console)	여러 줄 입력
Esc	콘솔 토글러 토글
Ctrl+K (Console)	콘솔 지우기

모바일 디바그	
기기 톨바	
Ctrl+Shift+M	기기 톨바 토글
Device dropdown	사전 설정 폰/태블릿 크기
Responsive mode	뷰포트 자유 조정
DPR dropdown	기기 픽셀 비율 변경
Throttling	모바일 CPU + 네트워크 시뮬레이션
Show media queries	CSS 중단점 시각화

원격 디버깅 (Android)	
1. Enable USB debugging	기기에서 설정 -> 개발자 옵션
2. Connect USB	Android 기기를 컴퓨터에 연결
3. chrome://inspect	데스크톱 Chrome에서 열기
4. Click Inspect	모바일 페이지용 DevTools 열기

센서 재정의	
Geolocation	GPS 좌표 재정의
Orientation	기기 방향 시뮬레이션
Touch	터치 이벤트 시뮬레이션
Idle detection	유휴 감지 API 재정의

일반 패턴	
네트워크 문제 디버그	
// In Console: monitor all fetch requests	
const origFetch = window.fetch;	
window.fetch = (...args) => {	
console.log('fetch:', args);	
return origFetch(...args);	
};	
성능 워크플로우	
1. Lighthouse audit	상위 성능 문제 파악
2. Performance recording	플래임 차트에서 각 작업 찾기

3. Coverage tab	미사용 CSS/JS 탐색 (Ctrl+Shift+P -> Coverage)
4. Network waterfall	블로킹 리소스 파악
5. Rendering tab	페인트/레이아웃 시각화 (Ctrl+Shift+P -> Rendering)

유용한 Console 스니펫	
// List all event listeners on element	
getEventListeners(\$0);	
// Monitor layout shifts	
new PerformanceObserver(l => l.getEntries().forEach(e => console.log('CLS:', e))).observe({ type: 'layout-shift', buffered: true });	