

CSS 빠른 참조

선택자, 레이아웃, flexbox, grid, 애니메이션, 반응형 디자인

선택자	
기본 선택자	
*	전체 — 모든 요소
div	타입 — 모든 <div> 요소
.class	클래스 — 해당 클래스를 가진 요소
#id	ID — 해당 id를 가진 요소
[attr]	속성 — 속성이 있는 요소
[attr="val"]	속성 값 일치
결합자	
A B	자손 (모든 깊이)
A > B	직접 자식만
A + B	인접 형제 (다음)
A ~ B	일반 형제 (이후 모두)
가상 클래스	
:hover	요소 위에 마우스 올림
:focus	요소에 포커스
:first-child	부모의 첫 번째 자식
:nth-child(n)	n번째 자식 (1부터, `odd`, `even`, `2n+1`)
:not(sel)	부모 — 매칭 제외
:has(sel)	부모 선택자 (매칭 포함)
가상 요소	
::before	요소 앞에 콘텐츠 삽입
::after	요소 뒤에 콘텐츠 삽입
::placeholder	입력 placeholder 텍스트 스타일
::selection	선택된 텍스트 스타일

박스 모델	
박스 사이징	
/* Include padding/border in width */	
*, ::before, ::after {	
box-sizing: border-box;	
}	
속성	
margin	테두리 바깥 공간
border	padding 주변 테두리
padding	테두리 안쪽 공간
width / height	콘텐츠 크기
outline	margin 바깥 링 (공간 없음)
단축 표기	
margin: 10px; /* all sides */	
margin: 10px 20px; /* vertical horizontal */	
margin: 10px 20px 30px; /* top horiz bottom */	
margin: 10px 20px 30px 40px; /* T R B L */	

Flexbox	
컨테이너	
.flex {	
display: flex;	
justify-content: center; /* main axis */	
align-items: center; /* cross axis */	
gap: 1rem;	
}	
컨테이너 속성	
flex-direction	`row` `column` `row-reverse` `column-reverse`
flex-wrap	`nowrap` `wrap` `wrap-reverse`
justify-content	`flex-start` `center` `space-between` `space-around` `space-evenly`
align-items	`stretch` `center` `flex-start` `flex-end` `baseline`
align-content	다중 줄 교차 정렬
gap	flex 아이템 사이 공간
아이템 속성	
flex: 1	남은 공간 채우기
flex: 0 0 200px	고정 너비, 늘림/줄임 없음
align-self	한 아이템에 대해 align-items 재정의
order	시각적 순서 변경 (기본값 0)

Grid	
컨테이너	
.grid {	
display: grid;	
grid-template-columns: repeat(3, 1fr);	
grid-template-rows: auto 1fr auto;	
gap: 1rem;	
}	
컨테이너 속성	
grid-template-columns	열 트랙 정의: `1fr 2fr`, `repeat(3, 1fr)`
grid-template-rows	행 트랙 정의
grid-template-areas	명령된 영역: `header header` "nav main"
gap	행 및 열 간격
justify-items	인라인 방향으로 아이템 정렬 (수평)
align-items	블록 방향으로 아이템 정렬 (수직)
아이템 배치	
.item {	
grid-column: 1 / 3; /* span cols 1-2 */	
grid-row: 1 / -1; /* span all rows */	
grid-area: header; /* named area */	
}	

타이포그래피	
폰트 속성	
font-family	폰트 스타: `Inter, sans-serif`
font-size	크기: `1rem`, `16px`, `clamp(1rem, 2vw, 2rem)`
font-weight	`normal` (400) `bold` (700) `100` - `900`
font-style	`normal` `italic` `oblique`

line-height	줄 간격: `1.5` (단위 없음 권장)
letter-spacing	자간: `0.05em`
텍스트 속성	
text-align	`left` `center` `right` `justify`
text-decoration	`none` `underline` `line-through`
text-transform	`uppercase` `lowercase` `capitalize`
text-overflow	`ellipsis` (overflow: hidden과 함께)
white-space	`nowrap` `pre` `pre-wrap`
word-break	`break-all` `break-word`

색상 & 배경	
색상 형식	
color: #fff660; /* hex */	
color: rgb(255, 102, 0); /* rgb */	
color: hsl(24, 100%, 50%); /* hsl */	
color: oklch(70% 0.15 50); /* oklch */	
배경 속성	
background-color	단색 채우기: `#0f0f0`
background-image	`url(img.jpg)` 또는 그라디언트
background-size	`cover` `contain` `100px 200px`
background-position	`center` `top right` `50% 50%`
background-repeat	`no-repeat` `repeat-x` `repeat-y`
그라디언트	
background: linear-gradient(to right, #f00, #00f);	
background: radial-gradient(circle, #fff, #000);	
background: conic-gradient(red, yellow, green);	

트랜지션 & 애니메이션	
트랜지션	
.btn {	
transition: background 0.3s ease, transform 0.2s;	
}	
.btn:hover {	background: #0056b3;
}	transform: scale(1.05);
트랜지션 속성	
transition-property	애니메이션할 속성
transition-duration	길이: `0.3s`, `300ms`
transition-timing-function	`ease` `linear` `ease-in-out` `cubic-bezier()`
transition-delay	시작 전 대기 시간
키프레임 애니메이션	
@keyframes spin {	
from { transform: rotate(0deg); }	
to { transform: rotate(360deg); }	
}	
.icon { animation: spin 1s linear infinite; }	
애니메이션 속성	
animation-name	@keyframes 이름 참조
animation-duration	한 주기의 길이
animation-iteration-count	`1` `3` `infinite`
animation-direction	`normal` `alternate` `reverse`
animation-fill-mode	`forwards` `backwards` `both`

반응형 디자인	
미디어 쿼리	
@media (max-width: 768px) {	
.sidebar { display: none; }	
}	
@media (prefers-color-scheme: dark) {	
body { background: #lalala; color: #eee; }	
}	
일반 브레이크포인트	
max-width: 480px	모바일 폰
max-width: 768px	태블릿
max-width: 1024px	소형 노트북
max-width: 1280px	데스크톱
뷰포트 단위	
vw / vh	뷰포트 너비 / 높이의 %
dvh	동적 뷰포트 높이 (모바일 안전)
svh / lvh	소형 / 대형 뷰포트 높이
cqi	컨테이너 쿼리 인라인 크기
컨테이너 쿼리	
.card-wrapper { container-type: inline-size; }	
@container (min-width: 400px) {	
.card { flex-direction: row; }	
}	

포지셔닝	
position 값	
static	기본값 — 일반 문서 흐름
relative	인라인 위치에서 요소의 공간 유지
absolute	가장 가까운 위치 지정 조상에 배치
fixed	뷰포트에 배치; 스크롤 시 고정
sticky	스크롤에 따라 relative/fixed 전환
중양 정렬 패턴	
/* Flex center */	
display: flex;	
justify-content: center;	
align-items: center;	
/* Grid center */	
display: grid;	
place-items: center;	
스태킹	
z-index	스택 순서 (높을수록 위); `position` 필요
isolation: isolate	새 스택링 컨텍스트 생성

커스텀 속성	
정의 & 사용	
:root {	
--color-primary: #3b82f6;	
--spacing-md: 1rem;	
}	
.btn {	
background: var(--color-primary);	
padding: var(--spacing-md);	
}	
대체값	
color: var(--accent, #fff660);	
/* uses #fff660 if --accent is not defined */	
--bg: #1a1a2e;	
--text: #e0e0e0;	
}	
body { background: var(--bg); color: var(--text); }	