

HTML/CSS

CSS 발전 흐름 및 박스모델과 레이아웃

김태민

저번에 우리는

WEB은 무엇을 위해 만들어졌죠?

Website

정적페이지 Static Page

단순 정보 전달용

단순한 상호작용

HTML, CSS, Javascript로

만들 수 있음

Web App

동적페이지 Dynamic Page

기술 복합 가치 전달

복잡한 상호작용

PHP, Ruby, Java, Python,

NodeJS, Rails, Django ...

CSS란?

Cascading Style Sheets의 약자

HTML 문서의 스타일(색상, 레이아웃, 폰트 등)을 정의
HTML(구조)과 분리하여 웹 페이지의 디자인과 시각적 표현 담당

유지보수 용이: 스타일 일괄 변경 가능

사용자 경험 개선: 반응형 디자인 지원

재사용성: 동일 스타일 여러 페이지에 적용



CSS 기초

CSS 기본 문법 및 적용 방법 (내부, 외부, 인라인)

기본 선택자 (*, tag, .class, #id)

속성 선택자 ([attribute])

후손/자식 선택자 (, >)

가상 클래스 및 가상 요소 (:hover, :first-child, ::after)

실습 환경

실습환경

OS : Window / Mac

브라우저 : Chrome

에디터 : VS Code <https://code.visualstudio.com/>

VS Code 익스텐션 : ESLint, Prettier, HTML CSS Support, HTML to CSS autocompletion, Auto Rename Tag, Auto Close Tag, htmltagwrap

EXTENSIONS: MARKETPLACE

HTML CSS Support

**HTML CSS Support** 4ms
CSS Intellisense for HTML
ecmel

**PostCSS Lang...** 1.4M 2.5
Syntax highlighting for mode...
csstools

**HTML/CSS/Jav...** 519K 4
HTML/CSS/JavaScript/Jade/P...
Eno Yao

**HTML-Essentials** 45K 5
Web Design essential extensi...
Gyduhn

**CSS Var Complete** 82K 5
Intellisense support for CSS V...
Shub

**Html And Css In...** 15K 4
This An Extension Which Gen...
Aditya Krishna

**HTML Detection ...** 2K 5
Simple format magic (HDNS) ...
Andra Ryanda

**IntelliSense for CSS c...** 4K
CSS class name completion f...
kang-sw

**Live HTML Previe...** 2K 5
Edit and preview HTML docu...
Invision

**Quanta CSS Intellisense** 3
Autocomplete and hover sup...

Extension: HTML CSS Support X



HTML CSS Support

ecmel  ecmel.dev | 25,921,535 | ★★★★★☆ (106) | Sponsor

CSS Intellisense for HTML

Disable

Uninstall

☒ Auto Update

DETAILS

FEATURES

CHANGELOG

Visual Studio Code CSS Intellisense for HTML

Tests passing

HTML `id` and `class` attribute completion for Visual Studio Code.

Features

- HTML `id` and `class` attribute completion
- Supports completion from in file defined styles
- Supports specifying remote and local style sheets
- Supports any language for completion
- Supports go to definition for selectors
- Validates class attributes on demand

Supported Languages

Supported languages can be configured with the `css.enabledLanguages` global setting. By default `html` is enabled:

```
{  "css.enabledLanguages": ["html"]}
```

Marketplace

Identifier	ecmel.vscode-html-css
Version	2.0.13
Published	2016-07-15, 06:02:33
Last Released	2025-01-24, 16:17:31

Categories

Programming Languages

Resources


[Marketplace](#)
[Issues](#)
[Repository](#)
[License](#)
[ecmel](#)

오즈코딩스쿨 x 고용노동부 1인 창업가 개발부트캠프


실습환경

EXTENSIONS: MARKETPLACE


Auto Rename Tag

 **Auto Rename Tag** 31ms
Auto rename paired HTML/XML tag
Jun Han


⚙️

 **Auto Rename Tag** 173K ★ 5
Auto rename paired HTML/XML tag
prasadbobby


Install

 **Auto Complete Tag** 2.9M ★ 4
Extension Packs to add close tag and re...
Jun Han


Install

 **React Essentials** 19K ★ 5
Andrés Gutiérrez


Install

 **Frontend Essentials Extens...** 37K ★ 5
Frontend Development Essentials Extens...
SeyyedKhandon


Install

 **Package ++** 23K ★ 5
Productivity extensions to make your co...
package ++


Install

 **VSCode Essentials** 28K ★ 5
Andrés Gutiérrez


Install

 **VSCode Pro - Frontend De...** 14K ★ 5
An Opinionated collection/pack of esse...
SeyyedKhandon


Install

 **DevOps Extension Pack** 5K ★ 1
The most common extensions for DevO...
walkme

Install

 **TypeScriptJS Essentials Extensi...** 4K

Extension: Auto Rename Tag


 **Auto Rename Tag**
Jun Han | 22,320,548 | ★★★★★☆ (206)
Auto rename paired HTML/XML tag
Disable Uninstall Auto Update ⚙️

DETAILS

FEATURES


CHANGELOG

Auto Rename Tag


 Build Status

Automatically rename paired HTML/XML tag, same as Visual Studio IDE does.

Sponsors



Increase your coding productivity with Tabnine's AI code completions! Tabnine is a free powerful Artificial Intelligence assistant designed to help you code faster, reduce mistakes, and discover best coding practices - without ever leaving the comfort of VS Code.
Tabnine is trusted by more than a million developers worldwide. [Get it now.](#)



Eliminate context switching and costly distractions. Create and merge PRs and perform code reviews from inside your IDE while using jump-to-definition, your keybindings, and other IDE favorites.
[Learn more](#)

Marketplace

Identifier	formulahendry.auto-rename-tag
Version	0.1.10
Published	2016-07-03, 17:34:28
Last Released	2022-02-08, 20:22:48

Categories

Programming Languages

Resources

Marketplace

Issues

Repository

License

Jun Han

오즈코딩스쿨 x 고용노동부 1인 창업가 개발부트캠프

실습환경

The screenshot displays the Visual Studio Code interface with the Extensions Marketplace open. The 'Auto Close Tag' extension by Jun Han is selected and its details are shown in the main panel. The extension's description states it automatically adds HTML/XML closing tags, similar to Visual Studio IDE or Sublime Text. The main panel also features a 'Usages' section with a code editor snippet showing an HTML tag being closed. The right sidebar provides marketplace information such as the identifier, version, publication date, and categories.

EXTENSIONS: MARKETPLACE

Auto close Tag

- Auto Close Tag** (5ms)
Automatically add HTML/XML close tag, ...
Jun Han
- Auto Complete Tag** (2.9M) ★ 4
Extension Packs to add close tag and re...
Jun Han
- Auto Close FTL Tag** (1K)
Auto Close Tag for ftl file.
Orca Team
- Auto Close Empty Tag** (3K)
Properly close HTML empty tags and co...
Emmanuel Paul Elom
- Auto Self Close Tag** (587) ★ 5
Automatically adds closing tags for HT...
Zeeshan Ahmad
- Auto Closing Tags** (4K)
Automatically close tags for multiple lan...
Codegyan
- jsx-self-closing-tag** (2K) ★ 5
help you auto close the self-closing tag ...
YangYongAn
- React Essentials** (19K) ★ 5
Andrés Gutiérrez
- VSCode Essentials** (28K) ★ 5
Andrés Gutiérrez
- C/C++ AutoClosingTag** (1K)
Auto Close the closing tag ; after each a...
hamzasadiq

Auto Close Tag
Jun Han | 15,360,817 | ★★★★★ (102)
Automatically add HTML/XML close tag, same as Visual Studio IDE or Sublime Text

Disable | Uninstall | Auto Update

Usages
After typing in the closing bracket of the opening tag, the closing tag will be inserted automatically.

test.html D:\Try
1 <html|

Marketplace

Identifier	formulahendry.auto-close-tag
Version	0.5.15
Published	2016-06-28, 23:06:44
Last Released	2023-12-19, 13:21:57

Categories
Programming Languages

Resources
Marketplace
Issues
Repository
License
Jun Han

실습환경

EXTENSIONS: MARKETPLACE

htmltagwrap

htmltagwrap

Wraps selected code with HTML tags

Brad Gashler

Extension: htmltagwrap

</>

HTML TagWrap

</>

htmltagwrap

Brad Gashler

Wraps selected code with HTML tags

836,863

★★★★☆ (65)

Disable

Uninstall

Auto Update

Alt + w

DETAILS

FEATURES

CHANGELOG

htmltagwrap

This VS Code extension wraps your text selection in HTML tags. It can wrap inline selections and selections that span multiple lines (works with both single selections and multiple selections at once).

To use, select one or many chunks of code and press "Alt + W" ("Option + W" for Mac).

demoFile.html

```
1 <html>
2   <head>
3   </head>
4   <body>
5     <span class="inline">Wrap inline</span> or block selections.
6     Multiple selections and empty selections also work.
7   </body>
8 </html>
```

Download it on the Visual Studio Marketplace

How to Use It

- Select one or more blocks of text or strings of text.
- Press Alt + W or Option + W for Mac.

Marketplace

Identifier

bradgashler.htmltagwrap

Version

1.0.0

Published

2015-12-10, 06:01:10

Last Released

2023-08-02, 13:48:15

Categories

Other

Resources

Marketplace

Issues

Repository

License

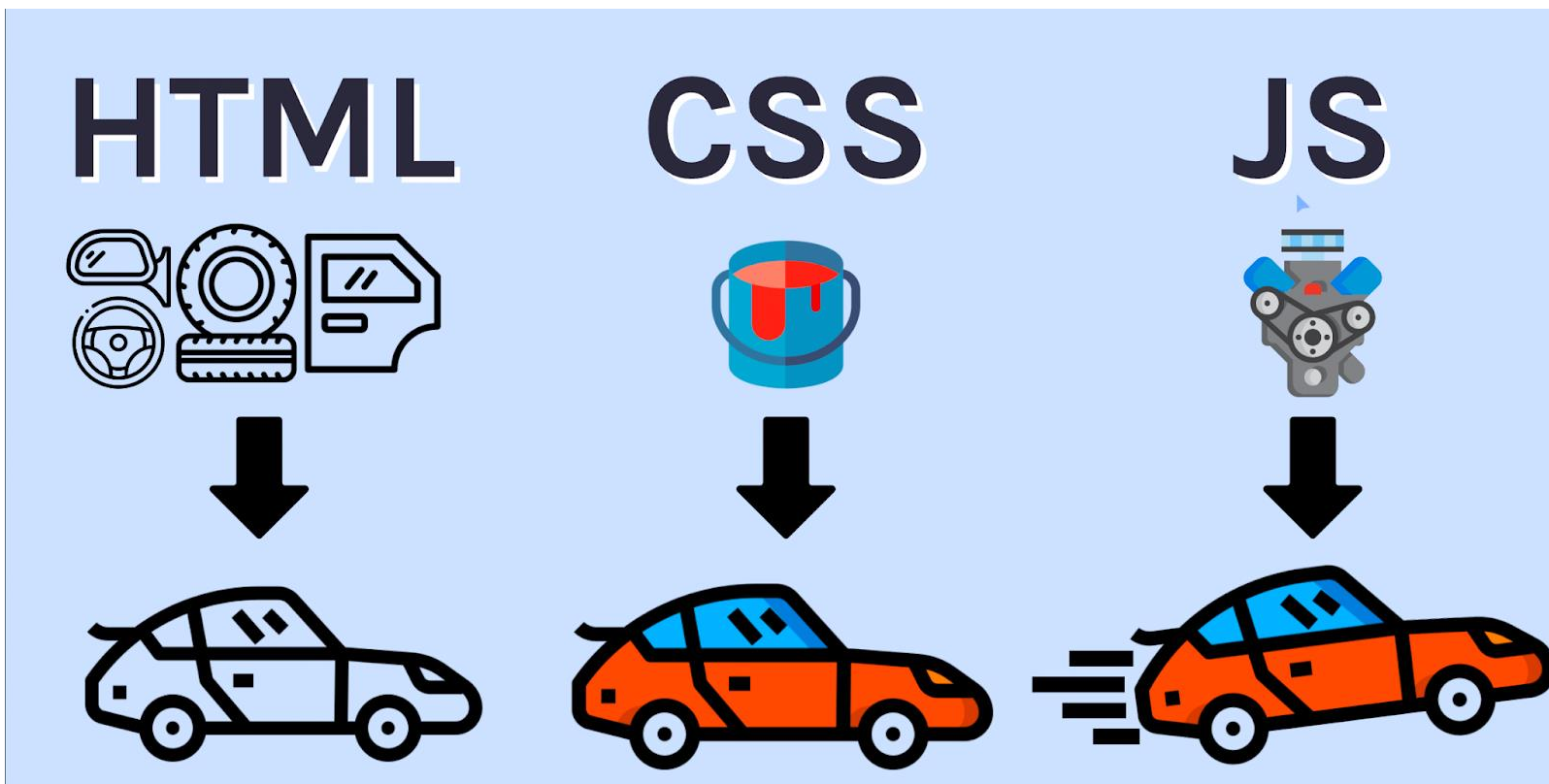
Brad Gashler

오즈코딩스쿨 x 고용노동부 1인 창업가 개발부트캠프

CSS 발전흐름

현대 WEB의 핵심요소 3가지

HTML, CSS, Javascript



CSS Level 1 (CSS1, 1996)

발표: W3C(World Wide Web Consortium)에서 공식 권고안 발표

주요 기능:

- 기본 스타일 속성: 글꼴(font), 색상(color), 여백(margin), 테두리(border)
- 선택자: 태그, 클래스(.class), ID(#id)
- 캐스케이딩: 스타일 우선순위 규칙 도입

한계:

- 레이아웃 제어 부족(표나 프레임에 의존)
- 브라우저 호환성 문제(넷스케이프 vs 인터넷 익스플로러)

영향: 웹 디자인의 기초 확립, HTML과 스타일 분리 시작

CSS Level 2 (CSS2, 1998)

주요 개선:

- 고급 선택자: 후손(), 자식(>), 가상 클래스(:hover, :first-child)
- 레이아웃 기능: position(absolute, relative), float
- 미디어 쿼리 초기 개념(인쇄용 스타일 등)
- 박스 모델 명확화(width, padding, margin)

문제점:

- 복잡성 증가로 브라우저 구현 불완전
- 인터넷 익스플로러의 비표준 구현으로 개발자 어려움

영향: 현대 웹 레이아웃의 기초 마련

CSS의 정체기와 도전 (2000년대 초)

상황:

- CSS2의 복잡성과 브라우저 호환성 문제로 개발자 혼란
- 플래시(Flash)와 테이블 레이아웃이 여전히 인기

2000년대 초반:

- CSS2.1로 수정 및 안정화 작업 진행
- 2011년 CSS2.1 공식 권고안 발표 (오류 수정, 모호성 제거)

변화:

- 웹 표준 운동 확산 (웹 표준 프로젝트, WaSP)
- 파이어폭스, 크롬 등 표준 준수 브라우저 등장

웹브라우저로 파이어폭스(**Firefox**)나 크롬(**Chrome**)을 사용한 직원들이 인터넷 익스플로러(**Internet Explorer**)나 사파리(**Safari**) 사용자들보다 재직 기간이 **15%나 더 길었으며...**

CSS3와 현대적 발전 (2000년대 말~2010년대)

주요 기능:

- 선택자 모듈: 속성 선택자([attr]), 가상 요소(::before, ::after)
- 레이아웃: Flexbox, Grid, 반응형 디자인을 위한 미디어 쿼리
- 시각효과/애니메이션 : border-radius, box-shadow, gradient, transform, @keyframes, transition

영향:

- 모바일 웹 급성장(스마트폰 보급)으로 반응형 디자인 필수
- 플래시 사용 감소, HTML5+CSS3 중심의 웹 개발 표준화

표준화:

- 브라우저 호환성 대폭 개선(크롬, 사파리, 엣지)

CSS 현재

CSS는 "Living Standard"로 지속 업데이트
접근성(ARIA)과 성능 최적화(예: Lazy Loading) 강조

새로운 기능:

- CSS Grid와 Subgrid로 복잡한 레이아웃 구현
- Container Queries: 컴포넌트 단위 반응형 디자인
- Custom Properties(--var): 변수로 스타일 재사용
- Houdini API: CSS 엔진 확장으로 커스텀 스타일링 가능

CSS 미래

미래 전망:

- AI 기반 디자인 도구와 CSS 통합
- VR/AR 웹 콘텐츠를 위한 3D 스타일링 확장
- WebAssembly와 결합한 고성능 웹 애플리케이션 지원

CSS는 웹의 시각적 표현과 **사용자 경험 혁신**의 핵심

박스모델과 레이아웃

박스모델과 레이아웃

박스 모델 개념 (content, padding, border, margin)

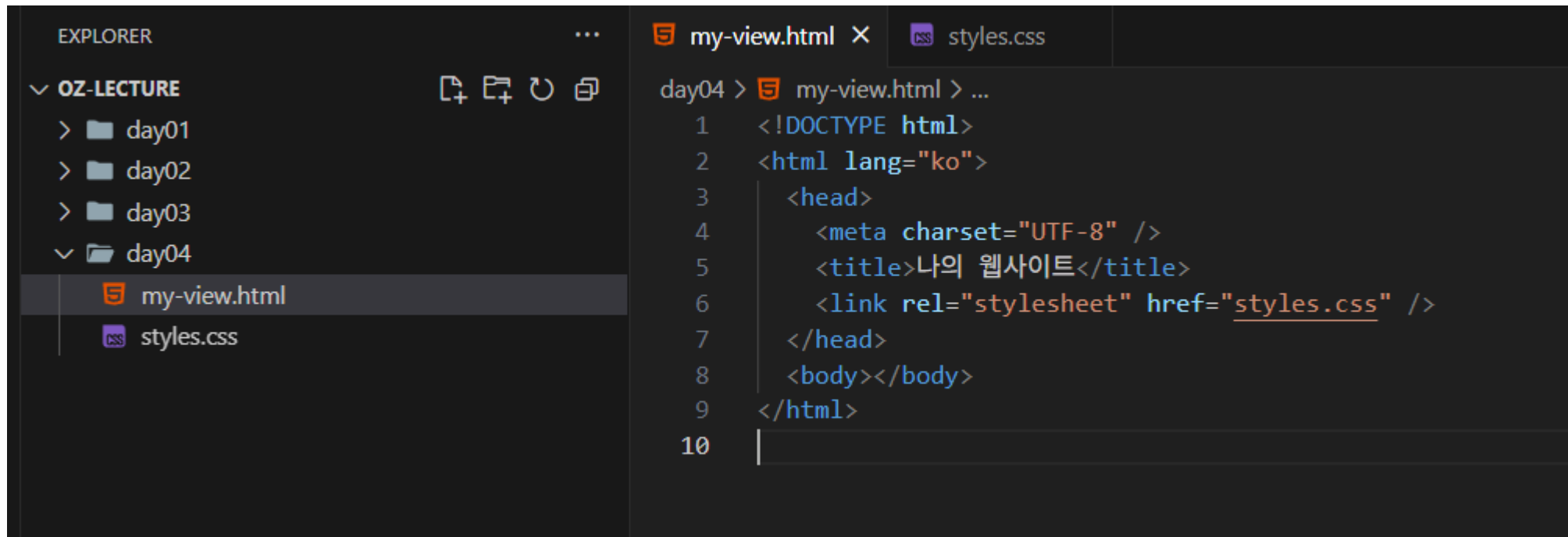
display 속성 (block, inline, inline-block, none)

position 속성 (static, relative, absolute, fixed, sticky)

float와 clear

flexbox와 grid 레이아웃 개념

Day04 설정



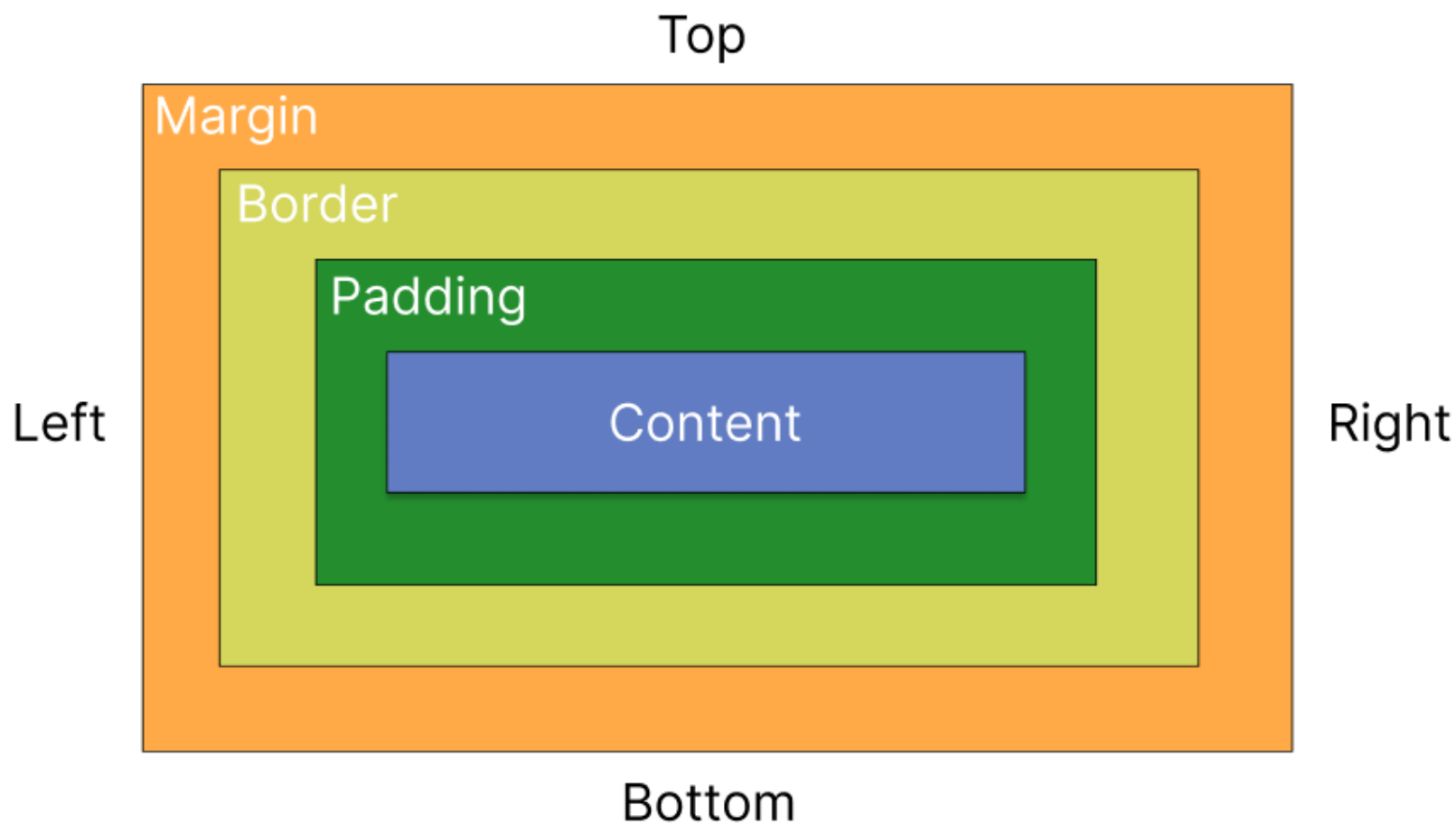
박스모델 개념

모든 HTML 요소는 박스 형태로 구성되며, 박스 모델은 요소의 크기와 간격을 정의

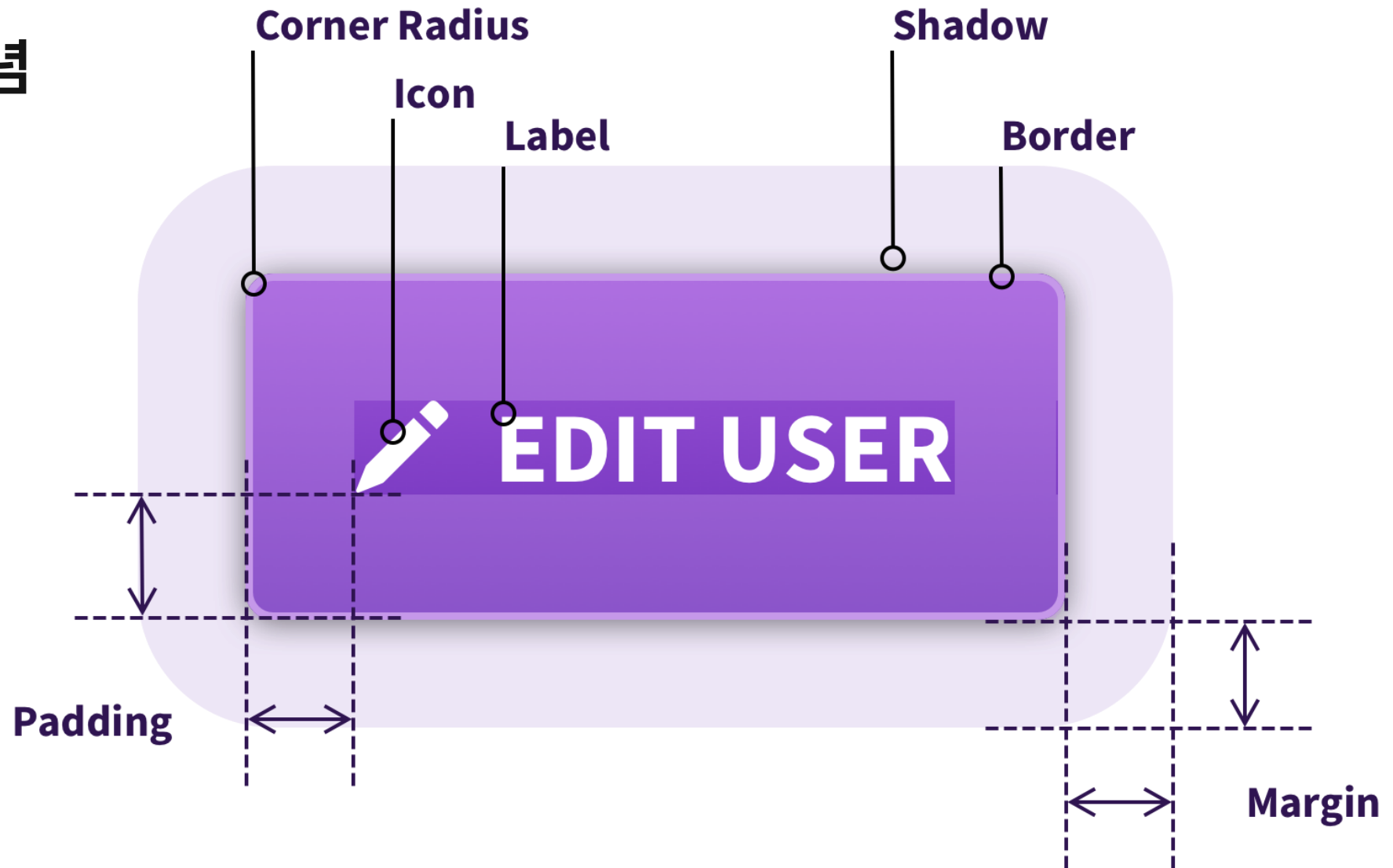
구성 요소:

- Content: 실제 콘텐츠(텍스트, 이미지 등)
- Padding: 콘텐츠와 테두리 간의 내부 여백
- Border: 콘텐츠와 패딩을 감싸는 테두리
- Margin: 박스 외부의 여백, 다른 요소와의 간격

박스모델 개념



박스모델 개념



박스모델과 레이아웃

```
my-view.html × styles.css
day04 > my-view.html > ...
1 <!DOCTYPE html>
2 <html lang="ko">
3 <head>
4   <meta charset="UTF-8" />
5   <title>나의 웹사이트</title>
6   <link rel="stylesheet" href="styles.css" />
7 </head>
8 <body>
9 <main>
10   <h1 class="title">제목입니다</h1>
11   <div class="box">
12     <p>본문 내용입니다.</p>
13     <p>본문 내용입니다.</p>
14     <p>본문 내용입니다.</p>
15     <p>본문 내용입니다.</p>
16     <p>본문 내용입니다.</p>
17     <button id="test-btn" type="button">테스트버튼</button>
18   </div>
19 </main>
20 </body>
21 </html>
```

```
my-view.html styles.css ×
day04 > styles.css > ...
1 * {
2   padding: 0;
3   margin: 0;
4 }
5 main {
6   padding: 30px;
7 }
8 .title {
9   font-size: 64px;
10  font-weight: bold;
11  margin: 10px 0px;
12 }
13 .box {
14   padding: 24px;
15   width: 300px;
16   border: solid 1px;
17   text-align: center;
18 }
19 #test-btn {
20   margin: 10px;
21   padding: 10px 24px;
22   font-size: 12px;
23   font-weight: bold;
24 }
25
```

Display 속성

HTML 요소의 표시 방식과 레이아웃 동작을 정의하는 CSS 속성

역할: 요소가 화면에 어떻게 배치되고, 다른 요소와 어떻게 상호작용하는지 결정

주요 값: block, inline, inline-block, flex, grid, none 등

block과 flex의 차이점과 활용 방식

Display: Block

특징:

- 요소가 새 줄에서 시작, 부모 컨테이너의 전체 너비 차지
- width, height 설정 가능

대표 요소: <div>, <p>, <h1>~<h6>, <section>

활용 사례:

- 섹션 구분(헤더, 푸터, 메인 콘텐츠)
- 단일 콘텐츠 블록(문단, 이미지 컨테이너)
- 고정된 레이아웃 구조

Display: Inline

특징:

- width와 height 속성 무시 (콘텐츠 크기에 의존)
- margin과 padding은 수평으로는 적용되지만, 수직으로는 레이아웃에 영향 최소화
- 기본적으로 텍스트 흐름을 따름

대표 요소: , <a>, ,

제한점:

- 고정 크기 설정 불가
- 수직 여백이 레이아웃을 깨뜨리지 않음 (다른 요소와 겹칠 수 있음)

Display: Inline-Block

요소가 한 줄에 나란히 표시 (줄 바꿈 없이)

width, height, margin, padding 설정 가능

대표 요소:

- 기본적으로 inline-block을 사용하는 요소는 없음.
- 버튼(<button>)이나 입력 필드(<input>)에 적용 시 유용

활용 사례:

- 나란히 배치된 버튼 그룹
- 이미지 갤러리 또는 카드 리스트
- 인라인 요소에 블록 속성(예: 고정 크기) 적용 필요 시

주의: 요소 간 공백(Whitespace) 발생 가능
(폰트 크기/줄 간격 영향)

Display: Flex

flex를 부모 요소에 적용, 자식 요소(Flex 아이TEM)를 유연하게 배치
주축(Main Axis)과 교차축(Cross Axis)을 기준으로 정렬

주요 속성:

- flex-direction: row(기본), column, row-reverse
- justify-content: 주축 정렬 (flex-start, center, space-between)
- align-items: 교차축 정렬 (stretch, center, baseline)
- flex-wrap: 줄 바꿈 여부 (nowrap, wrap)

활용 사례:

- 반응형 네비게이션 메뉴
- 카드 리스트, 갤러리 레이아웃 / 요소 간 균등 배분(예: 버튼 그룹, 푸터 링크)

자식 요소:

- flex: 크기 비율 (flex: 1)
- order: 순서 변경

Display: Grid

2차원 레이아웃(행과 열)으로 복잡한 구조를 정밀하게 배치

부모 요소:

- `grid-template-columns`: 열 크기 정의 (예: `1fr 1fr`)
- `grid-template-rows`: 행 크기 정의
- `gap`: 셀 간 간격

자식 요소:

- `grid-column`: 열 범위 지정 (예: `1 / 3`)
- `grid-row`: 행 범위 지정

Display: None

요소를 화면에서 완전히 숨김, 레이아웃 공간도 차지하지 않음

JavaScript로 동적 표시/숨김 제어에 자주 사용

대안: visibility: hidden은 공간을 차지하며 숨김

활용 사례:

- 모달 창, 드롭다운 메뉴 등 사용자 상호작용 시 표시/숨김
- 조건부 콘텐츠 렌더링(예: 에러 메시지 토글)
- 반응형 디자인에서 특정 화면 크기에서 요소 제거

Block vs Flex

항목	Block	Flex
기본 동작	새 줄에서 시작, 부모 너비 100% 차지	자식 요소를 유연하게 배치 (행/열)
레이아웃	수직 쌓임, 고정된 블록 형태	1차원 레이아웃, 정렬 및 크기 조정 가능
자식 요소 제어	자식 요소의 배치 직접 제어 불가	justify-content, align-items로 정렬
주요 속성	width, height, margin, padding	flex-direction, flex-wrap, gap
유연성	고정된 구조에 적합	반응형, 동적 레이아웃에 적합

Block: 정적인 구조, 콘텐츠 블록 단위로 쌓을 때 적합
Flex: 동적인 배치, 요소 간 간격/정렬 조정이 필요한 경우 유리

Block vs Flex

항목	Block	Flex
기본 동작	새 줄에서 시작, 부모 너비 100% 차지	자식 요소를 유연하게 배치 (행/열)
레이아웃	수직 쌓임, 고정된 블록 형태	1차원 레이아웃, 정렬 및 크기 조정 가능
자식 요소 제어	자식 요소의 배치 직접 제어 불가	justify-content, align-items로 정렬
주요 속성	width, height, margin, padding	flex-direction, flex-wrap, gap
유연성	고정된 구조에 적합	반응형, 동적 레이아웃에 적합

Display 실습

https://www.notion.so/display-html-217caf5650aa80c1b1bfd52d04e0c97d?source=copy_link

Position 속성

CSS의 position 속성은 HTML 요소의 위치를 제어하며, 요소가 문서 흐름에서 어떻게 배치되는지 결정

역할: 요소를 원하는 위치에 정밀하게 배치하거나, 스크롤 및 뷰포트에 따라 동작을 조정

주요 값: static, relative, absolute, fixed, sticky

참고: position 속성은 top, right, bottom, left 속성과 함께 사용해 위치 조정

Position: Static

기본값, 요소가 문서의 정상 흐름(Normal Flow)에 따라 배치

특징:

- top, right, bottom, left 속성 무시
- 요소는 HTML 코드 순서대로 배치
- z축(레이어) 조정 불가

활용 사례:

- 특별한 위치 조정이 필요 없는 기본 레이아웃
- 대부분의 요소가 기본적으로 static 사용

Position: Relative

요소가 원래 위치를 기준으로 상대적으로 이동, 문서 흐름 유지

특징:

- top, right, bottom, left로 원래 위치에서 이동
- 원래 공간은 유지(다른 요소에 영향 없음)
- z축 조정 가능 (z-index)

활용 사례:

- 요소를 미세하게 위치 조정
- absolute 자식 요소의 기준점 역할

Position: Absolute

요소가 가장 가까운 position이 static이 아닌 부모 요소를 기준으로 배치, 문서 흐름에서 제거

특징:

- top, right, bottom, left로 기준점(부모 또는 뷰포트)에서 위치 지정
- 원래 공간 차지하지 않음 (다른 요소가 그 자리 채움)
- z축 조정 가능 (z-index)

활용 사례:

- 팝업, 툴팁, 오버레이 등 특정 위치에 고정 배치
- 부모 요소 내 정밀한 레이아웃 구성

Position: Sticky

요소가 문서 흐름을 따르다가 특정 스크롤 위치에서 고정(뷰포트 기준)

특징:

- top, right, bottom, left 중 하나 필수 지정
- 부모 요소 내에서 동작, 부모를 벗어나면 일반 흐름 복귀
- z축 조정 가능 (z-index)

활용 사례:

- 섹션 헤더가 스크롤 시 상단에 고정
- 사이드바의 특정 항목 고정

Position 실습

<https://www.notion.so/position-html-217caf5650aa80e985c8dd65833dd89d>

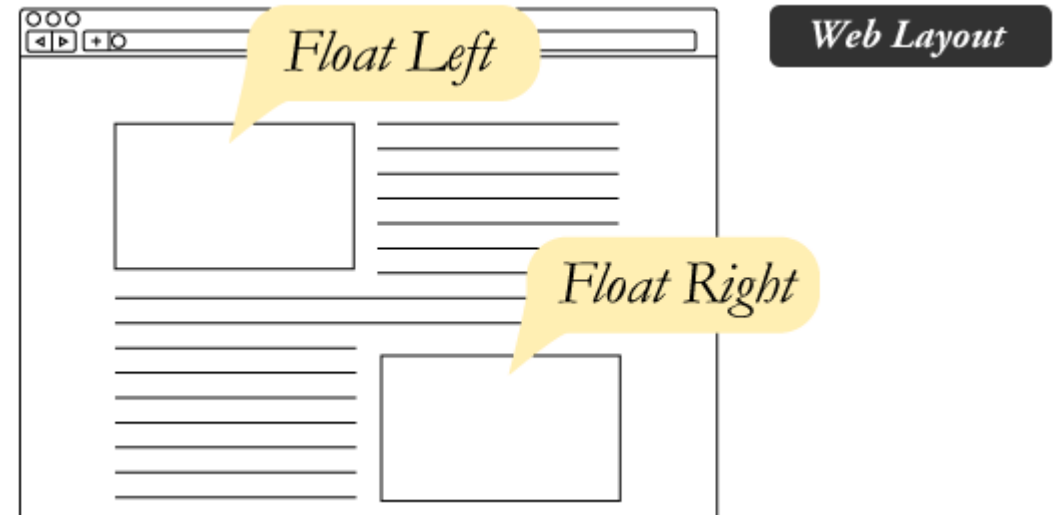
Float와 Clear

Float:

- CSS 속성으로, 요소를 문서 흐름에서 띄워 왼쪽 또는 오른쪽으로 배치, 주변 콘텐츠가 이를 감싸도록 함
- 이미지와 텍스트를 나란히 배치하거나, 간단한 레이아웃 구

Clear:

- Float된 요소의 영향을 제거하여 정상적인 문서 흐름 복원
- Float로 인해 어긋난 레이아웃을 정리
- 현대에는 Flexbox와 Grid가 주로 사용되지만, float는 여전



Float

주요 값:

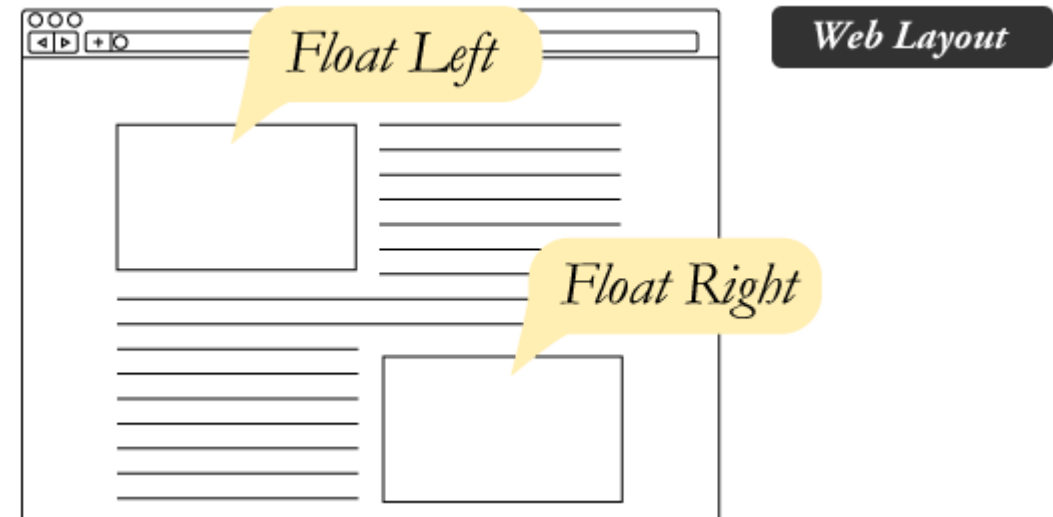
- left: 요소를 왼쪽으로 띄움, 뒤따르는 콘텐츠가 오른쪽에 배치
- right: 요소를 오른쪽으로 띄움, 뒤따르는 콘텐츠가 왼쪽에 배치
- none: 기본값, 띄우지 않음

특징:

- Float된 요소는 문서 흐름에서 부분적으로 제거됨
- width가 콘텐츠 크기에 따라 자동 조정(블록 요소의 경우)

활용 사례:

- 이미지와 텍스트를 나란히 배치 (예: 기사 레이아웃)



Float 부모요소 높이 붕괴 문제

문제: Float된 자식 요소로 인해 부모 요소가 높이를 잃음

원인: Float된 요소는 문서 흐름에서 벗어나 부모의 높이 계산에 포함되지 않음

해결 방법: Clearfix 부모에 가상 요소 추가

결과: 부모 요소가 Float된 자식의 높이를 포함, 정상적인 레이아웃 유지



Clear

주요 값:

- left: 왼쪽 Float 영향 제거
- right: 오른쪽 Float 영향 제거
- both: 양쪽 Float 영향 제거
- none: 기본값, Clear 적용 안 함

특징:

- Float된 요소 이후의 요소에 적용, 정상 문서 흐름 복원

활용 사례:

- Float된 이미지/박스 이후 콘텐츠를 새 줄로 강제 이동

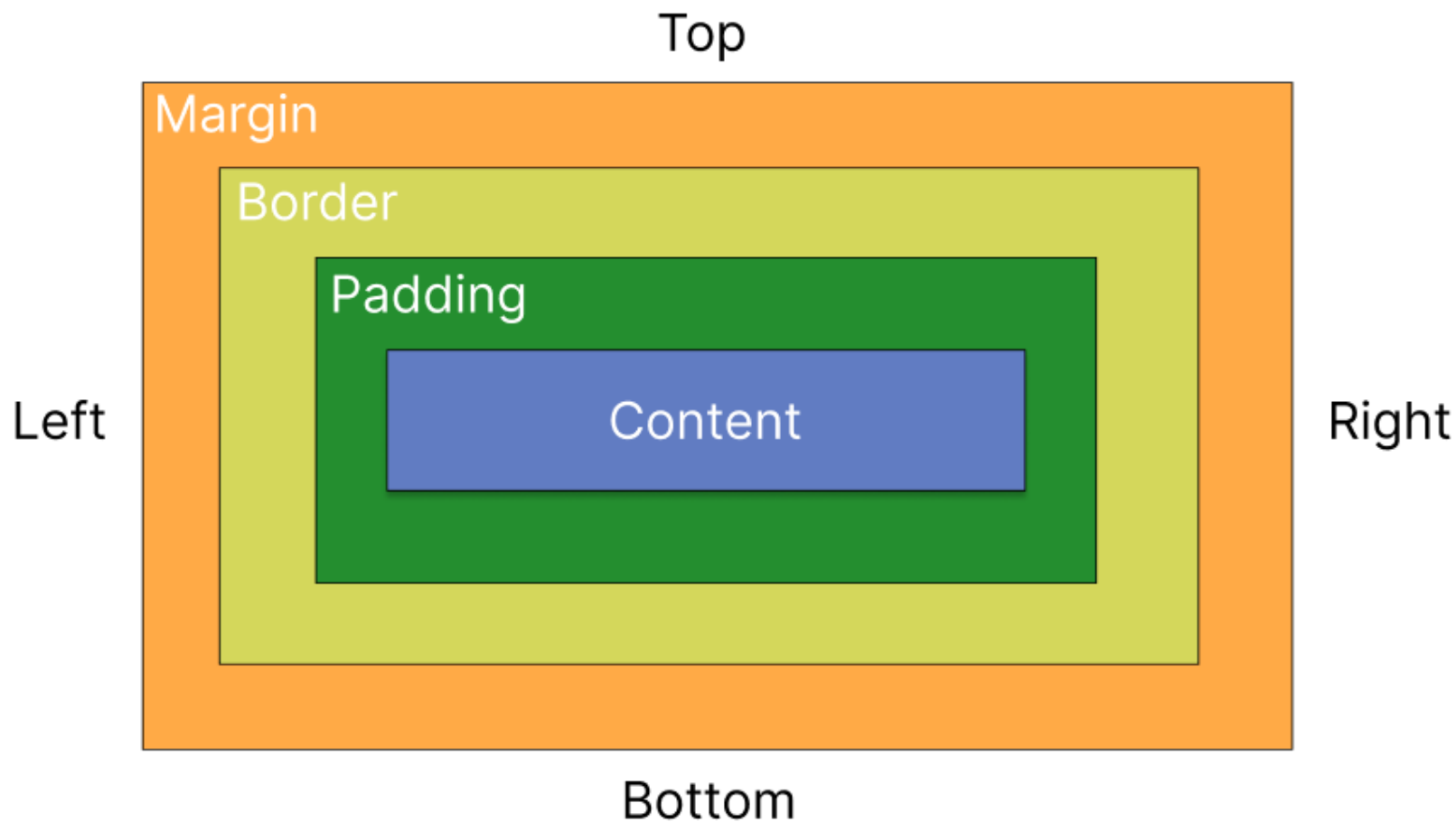


Float, Clear 실습

<https://www.notion.so/float-html-217caf5650aa80039879e1d08074293d>

오늘 우리는

박스모델 개념



Display 속성

HTML 요소의 표시 방식과 레이아웃 동작을 정의하는 CSS 속성

역할: 요소가 화면에 어떻게 배치되고, 다른 요소와 어떻게 상호작용하는지 결정

주요 값: block, inline, inline-block, flex, grid, none 등

block과 flex의 차이점과 활용 방식

Position 속성

CSS의 position 속성은 HTML 요소의 위치를 제어하며, 요소가 문서 흐름에서 어떻게 배치되는지 결정

역할: 요소를 원하는 위치에 정밀하게 배치하거나, 스크롤 및 뷰포트에 따라 동작을 조정

주요 값: static, relative, absolute, fixed, sticky

참고: position 속성은 top, right, bottom, left 속성과 함께 사용해 위치 조정

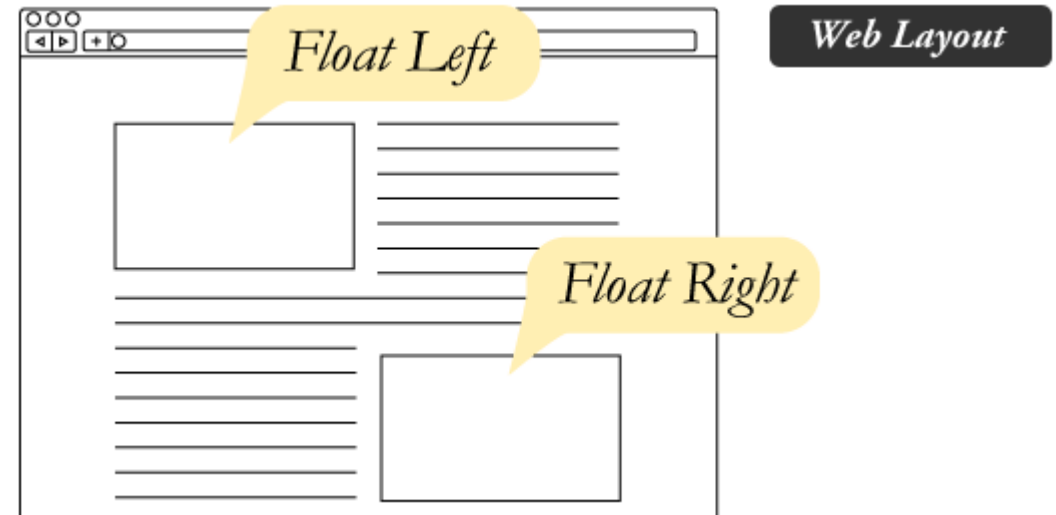
Float와 Clear

Float:

- CSS 속성으로, 요소를 문서 흐름에서 띄워 왼쪽 또는 오른쪽으로 배치, 주변 콘텐츠가 이를 감싸도록 함
- 이미지와 텍스트를 나란히 배치하거나, 간단한 레이아웃 구

Clear:

- Float된 요소의 영향을 제거하여 정상적인 문서 흐름 복원
- Float로 인해 어긋난 레이아웃을 정리
- 현대에는 Flexbox와 Grid가 주로 사용되지만, float는 여전



감사합니다