

WEB STUDY

2 WEEK

[2 week 목표]

- HTML과 CSS를 함께 상호작용해서 사용할 수 있다.
- CSS의 기본적인 컨셉과 개념을 이해한다.

[CONTENTS]

0. 과제 피드백

1. CSS 개요

2. 선택자

3. Block vs Inline

4. Box Model

0. 과제 피드백

HTML - class 속성

```
<div class="side-bar"></div>
```

1. CSS 개요

CSS (Cascading Style Sheets)



CSS의 컨셉은 웹 디자인 뿐만 아니라 데스크탑 앱, 모바일 앱 등 소프트웨어 개발에서 중요한 디자인 컨셉입니다.

HTML + CSS

1. 태그의 style 속성에 값으로 지정하기 (우선적으로 적용)

```
<div id="box-1" style="color: ■ red">박스1</div>
```

2. <style> CSS </style>

```
<style>
  div {
    color: ■ blue;
  }
</style>
```

3. styles.css 파일을 따로 만들어서 link 태그로 연결하기

```
<link rel="stylesheet" href="._css/styles.css" />
```

2. 선택자

CSS의 선택자

CSS는 HTML의 태그를 선택하고 선택한 태그에 스타일을 지정합니다.

```
Selector {
```

```
    속성 : 값;
```

```
}
```

CSS의 선택자

tagname { ... }

.classname { ... }

#idname { ... }

[attribute] { ... }

* { ... }

CSS의 가상 선택자

선택자 뒤에 : 가상 이벤트를 붙이면 특정 이벤트마다 적용 할 스타일을 설정 할 수 있으며, 이를 가상 선택자라고 합니다.

- `:link` 방문 전 링크 상태
- `:visited` 방문 후 링크 상태
- `:hover` 마우스 오버했을 때 링크 상태
- `:active` 클릭했을 때 링크 상태
- `:focus` 포커스 되었을 때 (<input>)
- `:first` 첫번째 요소
- `:last` 마지막 요소
- `:first-child` 첫번째 자식
- `:last-child` 마지막 자식
- `:nth-child(3)` 3 번째 자식
- `:nth-child(2n+1)` 홀수 번째 자식

3. Block vs Inline

Block vs Inline

display: block;

- 대표적인 태그 : <div>

1. 자신이 있는 한 줄 만큼의 공간을 차지합니다.
2. 줄바꿈을 합니다.
3. 가상의 박스라고 생각하면 됩니다.

<h1~6>, <header>, <main>, <section>,
<footer>, <p>, <form>, , , ...

display: inline;

- 대표적인 태그 :

1. 자신의 Text 크기 만큼만 공간을 차지합니다.
2. 줄바꿈을 하지 않습니다.
3. width / height 적용 불가
4. margin / padding-top(bottom) 적용 불가

<a>, <input>, <video>, <textarea>, <button>,
, <i>, <sub>, <sup>, ...

display: inline-block ??

- 줄바꿈이 이루어지지 않습니다.
- block처럼 width와 height를 지정할 수 있습니다.
- 만약 width와 height를 지정하지 않는 경우, inline처럼 동작합니다.

3. Box Model

Border

<https://developer.mozilla.org/ko/docs/Web/CSS/border>

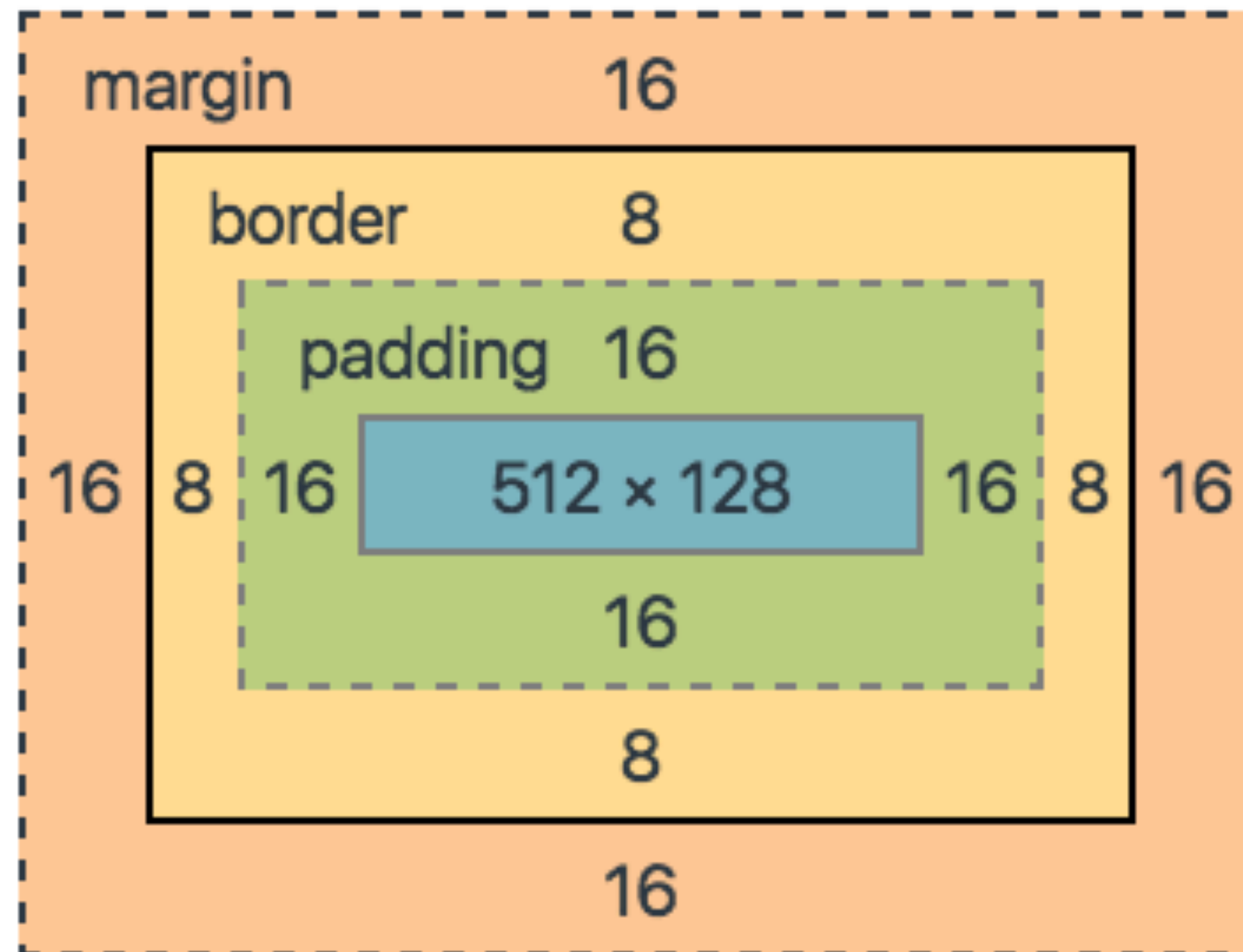
Margin

<https://developer.mozilla.org/ko/docs/Web/CSS/margin>

Padding

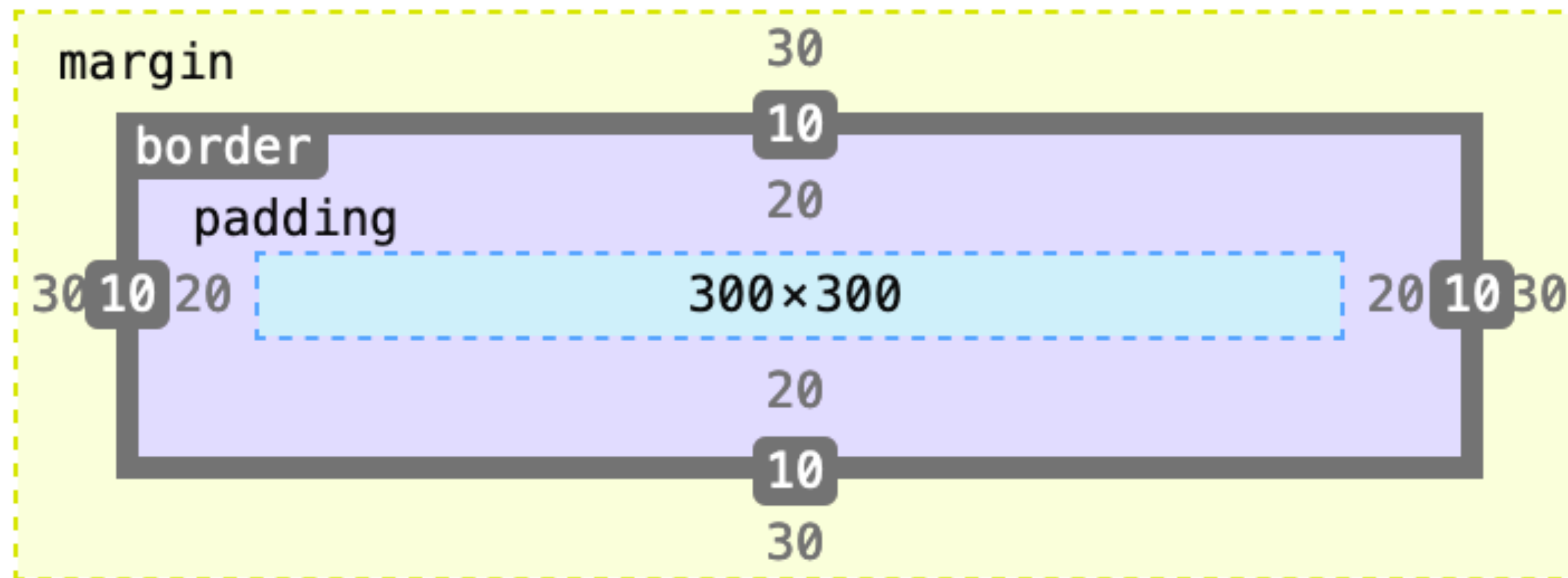
<https://developer.mozilla.org/ko/docs/Web/CSS/padding>

Box Model



box-sizing : content-box; (기본 값)

```
div {  
  box-sizing: content-box;  
  width: 300px;  
  height: 300px;  
  padding: 20px;  
  border: 10px solid black;  
  margin: 30px;  
}
```

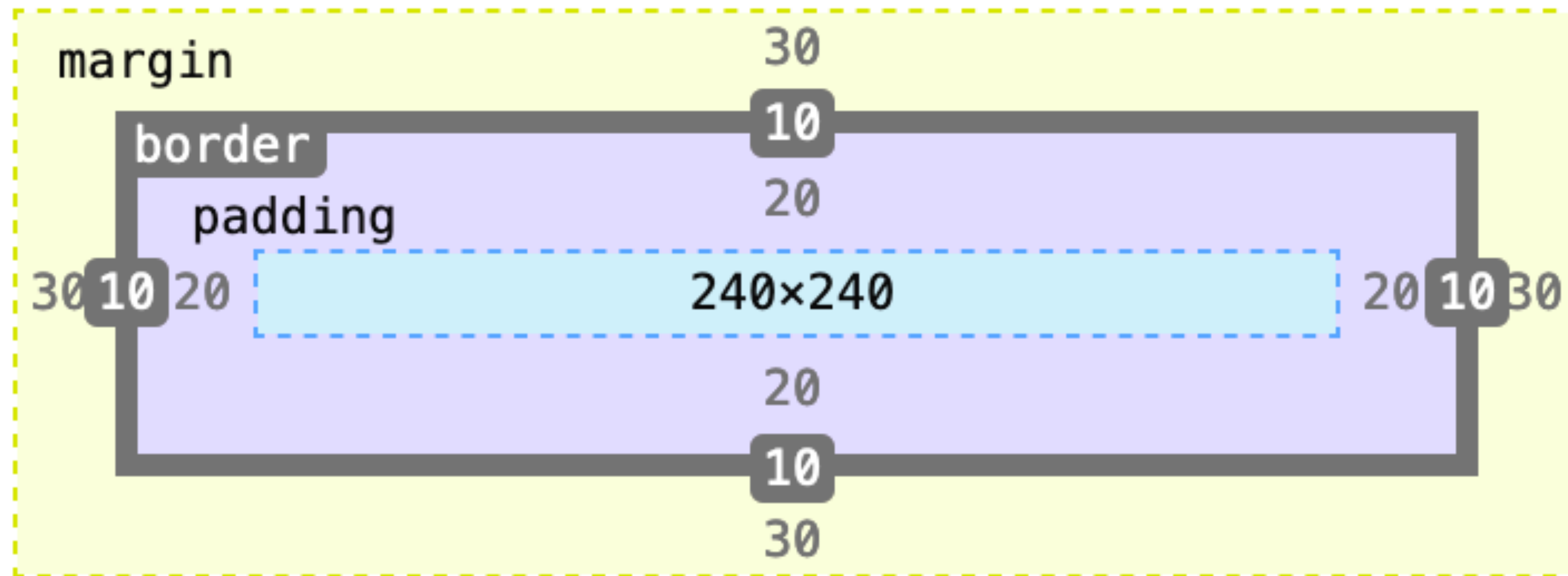


360x360

static

box-sizing : border-box;

```
div {  
  box-sizing: border-box;  
  width: 300px;  
  height: 300px;  
  padding: 20px;  
  border: 10px solid black;  
  margin: 30px;  
}
```



300x300

static

다음주에는 CSS Layout에 대해 학습합니다.

웹에서 레이아웃을 디자인하는 여러가지 방법이 있습니다.

1. HTML에서 <table>태그를 사용하여 디자인 하는 방법
2. CSS에서 “display: inline-block;”을 이용하여 디자인하는 방법
3. CSS에서 float 속성을 이용하여 디자인하는 방법
4. CSS에서 “display: flex;”을 이용하여 디자인하는 방법
5. CSS에서 “display: grid;”을 이용하여 디자인하는 방법

flex와 grid를 사용하여 디자인하는 방법이 가장 가독성이 높고 효율적인 디자인 방법입니다.

과제

1. 복습용 이론 영상 보기 (영상 두 개)

<https://youtu.be/gGebK7IWnCk>

<https://youtu.be/Zny5Vxqk6Mk>

2. 저번 주에 개발한 웹 문서에 **CSS**를 사용하여 디자인한 후 코드와 링크를 보내주세요.

- 기존의 주제 변경해도 됩니다.
- css 파일을 따로 만들고 <link> 태그를 사용하여 개발해주세요.

3. 구글 홈 화면 사이트를 “똑같이” 개발해서 코드와 링크를 보내주세요.

- 다음주에 제가 직접 라이브 코딩으로 하나하나 만듭니다. 그 전에 여러분만의 스타일로 개발해서 보내주시면 됩니다.
- 구글링을 하셔서 개발해 주시면 됩니다. (코드를 반드시 이해하고 사용해야 합니다.)
- 막막할 수 있지만 최대한 똑같이 만들어야 합니다.
- Bootstrap, TailwindCSS 등 CSS 라이브러리는 사용하면 안됩니다.
- css 파일을 따로 만들고 <link> 태그를 사용하여 개발해주세요.



🔍

⌨️

Google 검색

I'm Feeling Lucky

Google 제공 서비스: [English](#)

참고 강의

- **CSS** 단위에 대하여 (10분) : <https://youtu.be/7Z3t1OWOpHo>
- **HTML + CSS** 유료 강의 : <https://nomadcoders.co/kokoa-clone>

풀스택 혹은 프론트엔드 개발로 직업을 삼으실 분들에게 추천드립니다.

3주 후에 CSS 스터디가 끝나면 수강하시는 것을 추천드립니다.

