

# **CSS LAYOUTING**

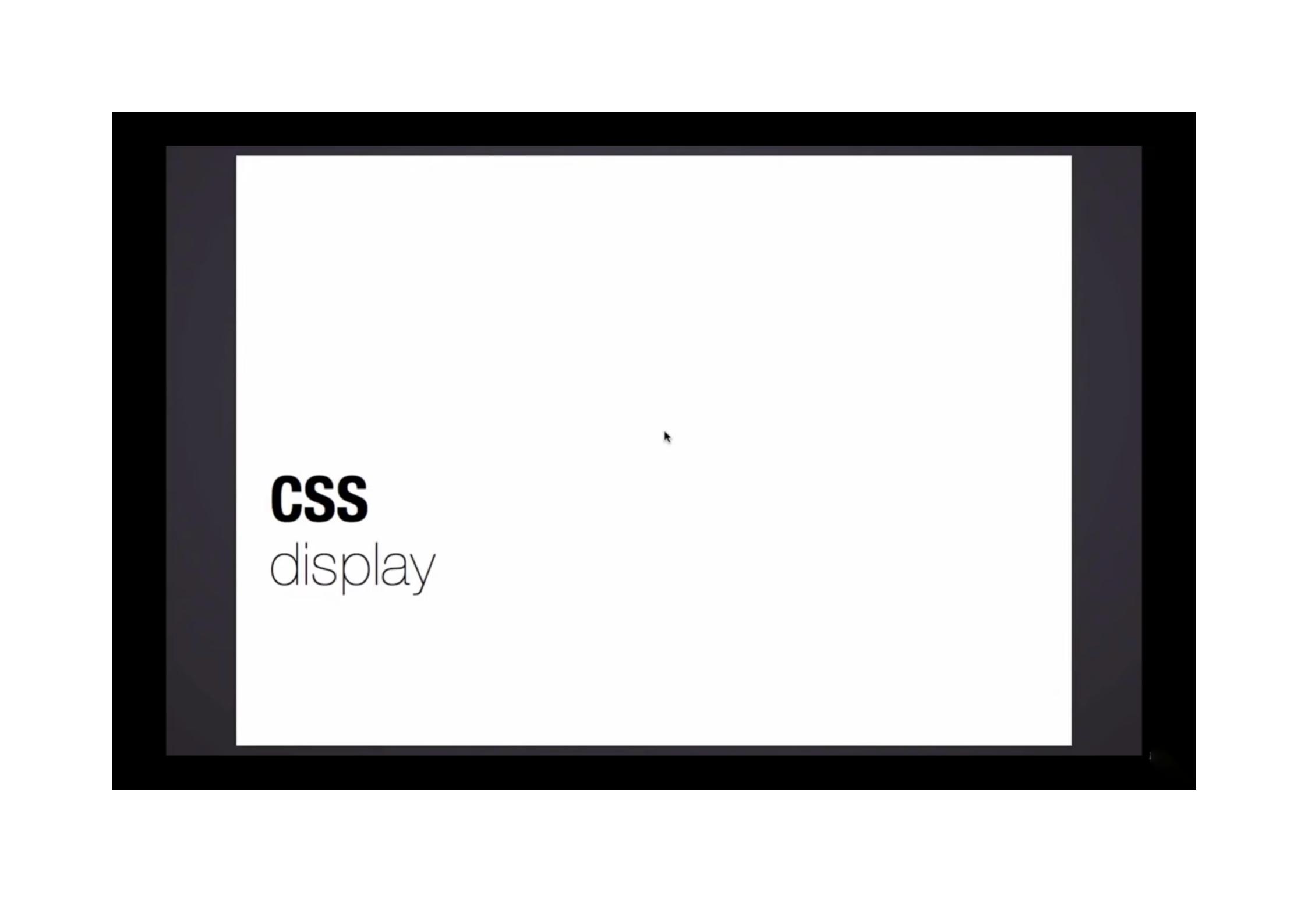
## **PENGATURAN TATA LETAK MENGGUNAKAN CSS**



# **DISPLAY**

## **MEMAHAMI PROPERTY DISPLAY PADA CSS**





**CSS**  
display

**<div> & <span>**

“

tag pada HTML digunakan untuk memberikan 'maksud' / 'arti' pada sebuah konten (**p** untuk paragraf, **h1** untuk heading utama, dll).

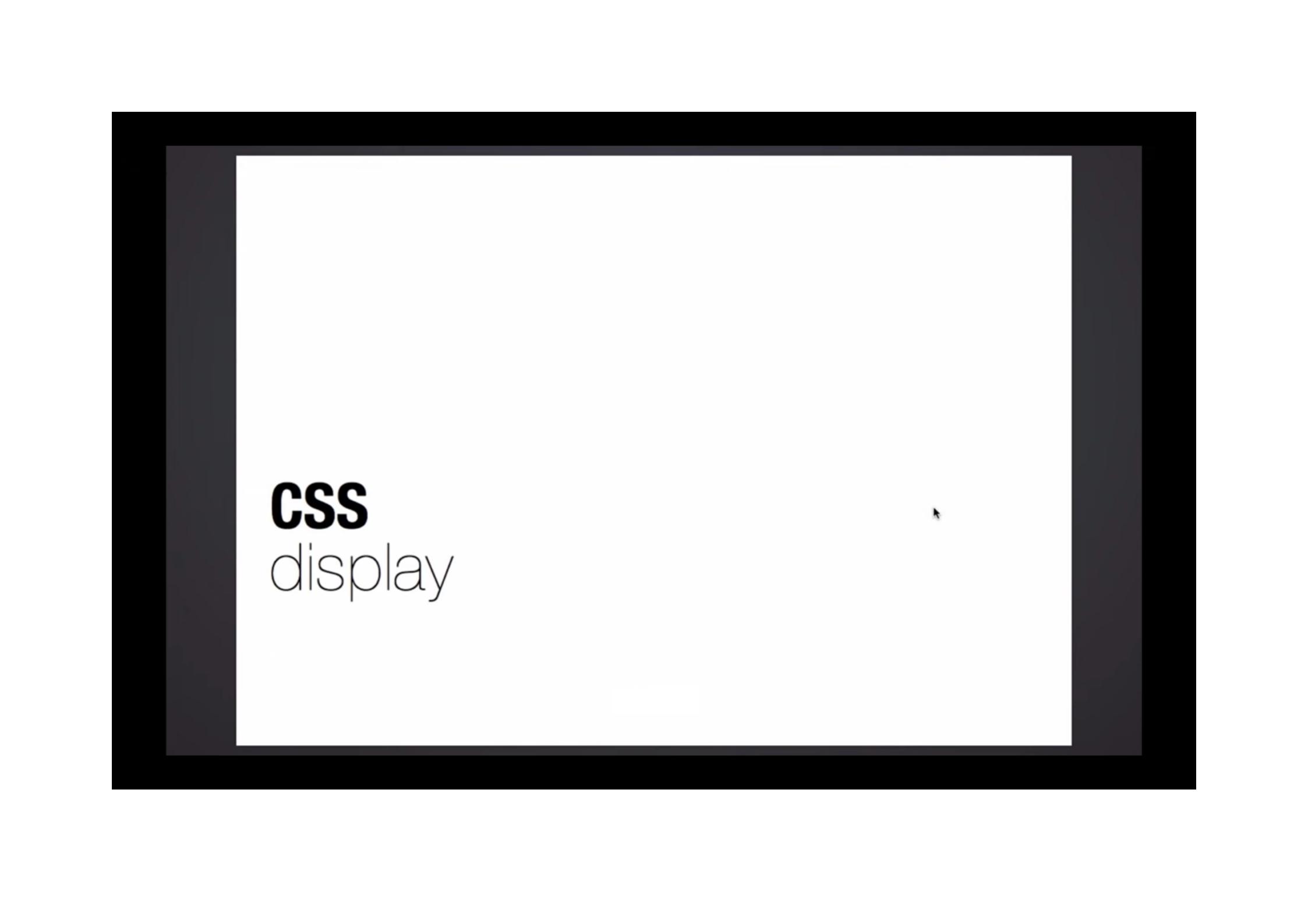
Tag **<div>** dan **<span>** tidak memiliki 'arti' apapun, keduanya digunakan untuk mengelompokkan tag-tag HTML dan memberikan informasi terhadap tag-tag tersebut.

```
<div></div>
```

```
div {  
    display: block;  
}
```

```
<span></span>
```

```
span {  
    display: inline;  
}
```



**CSS**  
display

“

setiap tag pada HTML berada di dalam sebuah **kotak**. properti **display** pada CSS mengatur perilaku dari kotak tersebut.

r

“

setiap tag pada HTML memiliki nilai default untuk properti **display**. Tetapi kita juga dapat mengubah perilaku dari tag tersebut dengan mengganti value-nya.

**display**

I

inline

inline-block

block

none

## inline

---

- Elemen HTML yang secara default tidak menambahkan baris baru ketika dibuat
- Lebar dan tinggi elemennya sebesar konten yang ada di dalamnya
- Kita tidak dapat mengatur tinggi dan lebar dari elemen inline
- *Margin dan padding hanya mempengaruhi elemen secara horizontal, tidak vertikal*

## **elemen inline**

b, strong, i, em,  
a, span, sub, sup,  
button, input, label,  
select, textarea,

## inline-block

---

- Tidak ada elemen yang secara default memiliki properti `display: inline-block;`
- Kita harus ubah secara manual properti tersebut
- Perilaku dasarnya sama dengan elemen inline
- Perbedaanya, elemen inline-block dapat kita atur tinggi dan lebar-nya

## **elemen block**

**h1-h6, p, ol, ul, li**

**form, hr, div,**

**...**

I

## block

---

- Elemen HTML yang secara default menambahkan baris baru ketika dibuat
- Jika tidak diatur lebar-nya, maka lebar default dari elemen block akan memenuhi lebar dari browser / parent-nya
- Kita dapat mengatur tinggi dan lebar dari elemen block
- Di dalam elemen block, kita dapat menyimpan tag dengan elemen inline, inline-block, atau bahkan elemen block lagi

none

---

Digunakan untuk menghilangkan sebuah elemen <sup>i</sup>

mengubah **display**

# **CSS LAYOUTING**

## **#2 DIMENSI**

# **CSS**

## Dimensi & Overflow

**CSS**  
dimensi

# **width & height**

I

px

%

in, cm, mm, pt, pc

**CSS**  
overflow

**overflow**

I

visible

auto

hidden

scroll

# CSS LAYOUTING BOX MODEL

**“**

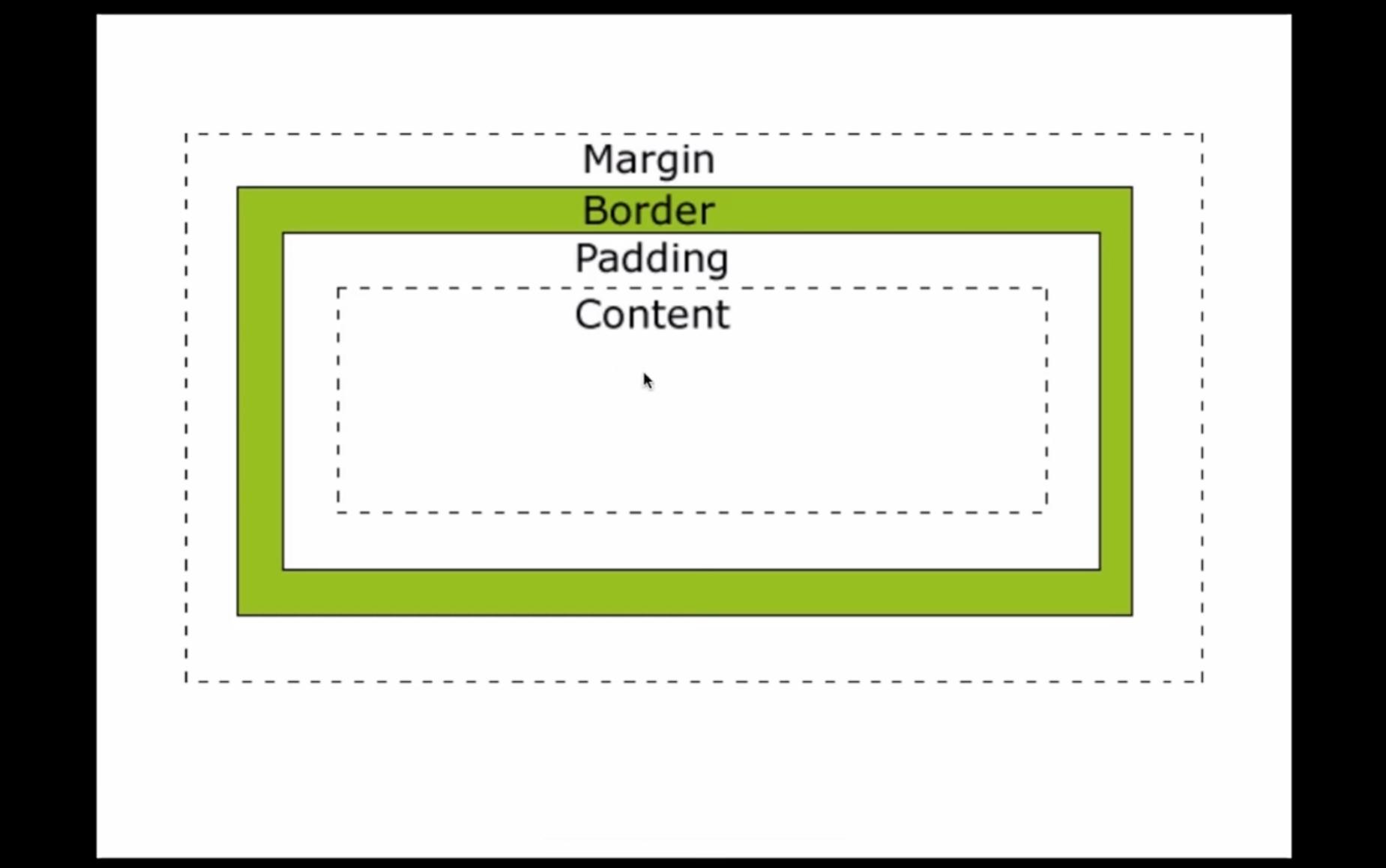
- setiap elemen di halaman berada di dalam sebuah box (kotak)
- kita bisa mengatur ukuran dan posisi kotak tersebut
- kita bisa memberi warna / gambar sebagai background kotak tersebut

“

CSS box model mendefinisikan ‘kotak’ yang dihasilkan oleh sebuah elemen, lalu menampilkannya sesuai dengan format visualnya.

“

CSS box model terdiri dari **margin**, **border**,  
**padding** dan **content**.



The diagram illustrates the CSS box model, which defines the structure of an element's content area. It consists of four concentric rectangular layers: **Margin**, **Border**, **Padding**, and **Content**. The **Content** layer is the innermost white rectangle. The **Padding** layer is the thin white border around the content. The **Border** layer is the thick green border around the padding. The **Margin** layer is the outermost dashed black border. A small black cursor icon is positioned in the center of the content area.

Margin

Border

Padding

Content

## CSS Box Model

---

### **margin**

area transparan di sekitar kotak (di luar border)

### **padding**

area transparan di dalam kotak (antara content dan border)

### **border**

batas disekeliling content dan padding

### **content**

konten sebenarnya di dalam box, bisa berupa teks atau gambar

properti untuk margin, padding & border

---

**margin-top**

**padding-top**

**border-top**

**margin-right**

**padding-right**

**border-right**

**margin-bottom**

**padding-bottom**

**border-bottom**

**margin-left**

**padding-left**

**border-left**

**margin**

**padding**

**border**

margin

# **margin**

cara pakai

overlapping margin

negatif margin

auto

shorthand

**margin-top**

**margin-right**

**margin-bottom**

**margin-left**



**margin**

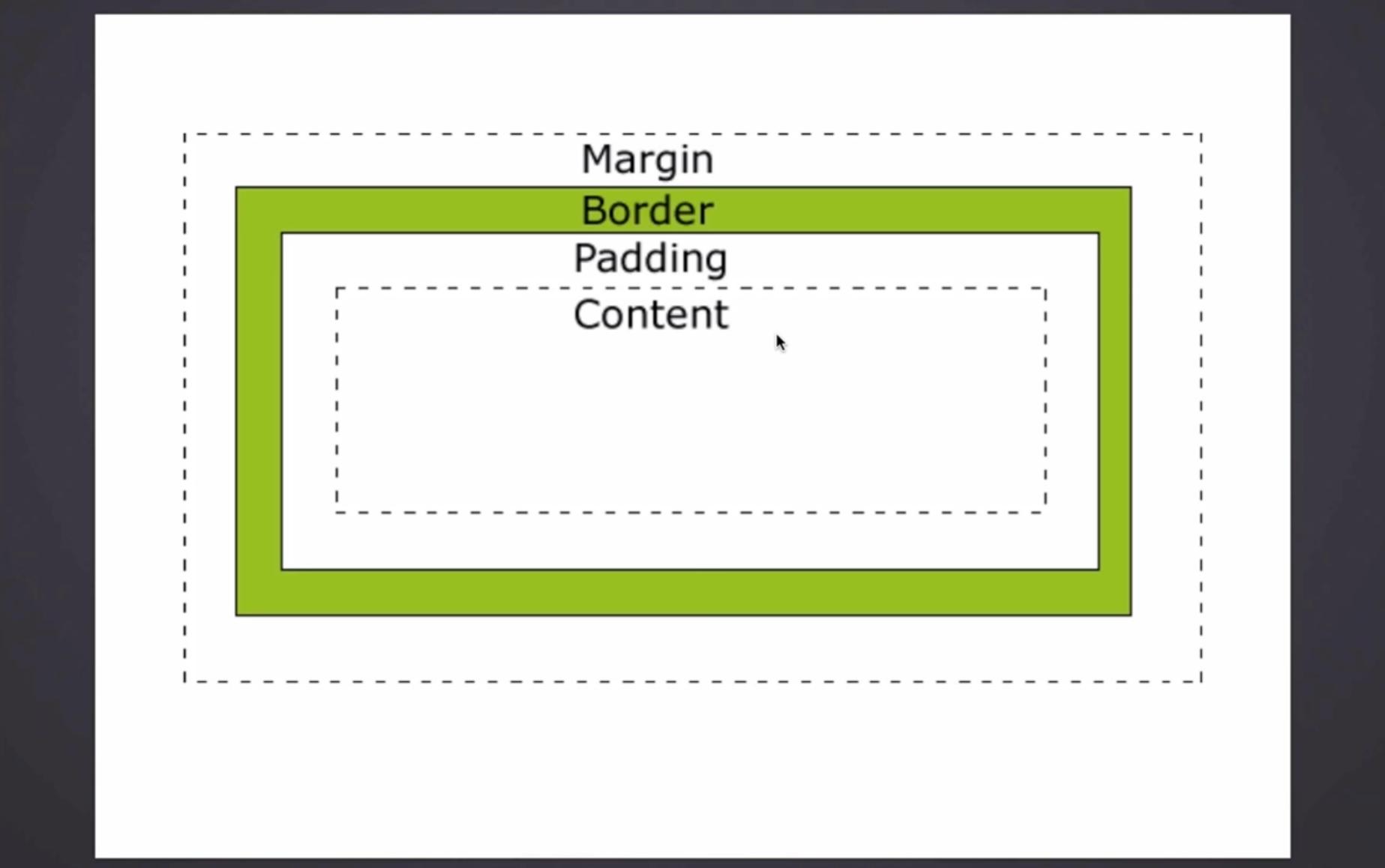
cara pakai

overlapping margin

negatif margin

auto

shorthand



The diagram illustrates the CSS box model, which defines the structure of an element's content area. It consists of four concentric rectangular layers: **Margin**, **Border**, **Padding**, and **Content**. The **Content** layer is the innermost white rectangle. The **Padding** layer is the thin white border around the content. The **Border** layer is the thick green border around the padding. The **Margin** layer is the outermost dashed black border.

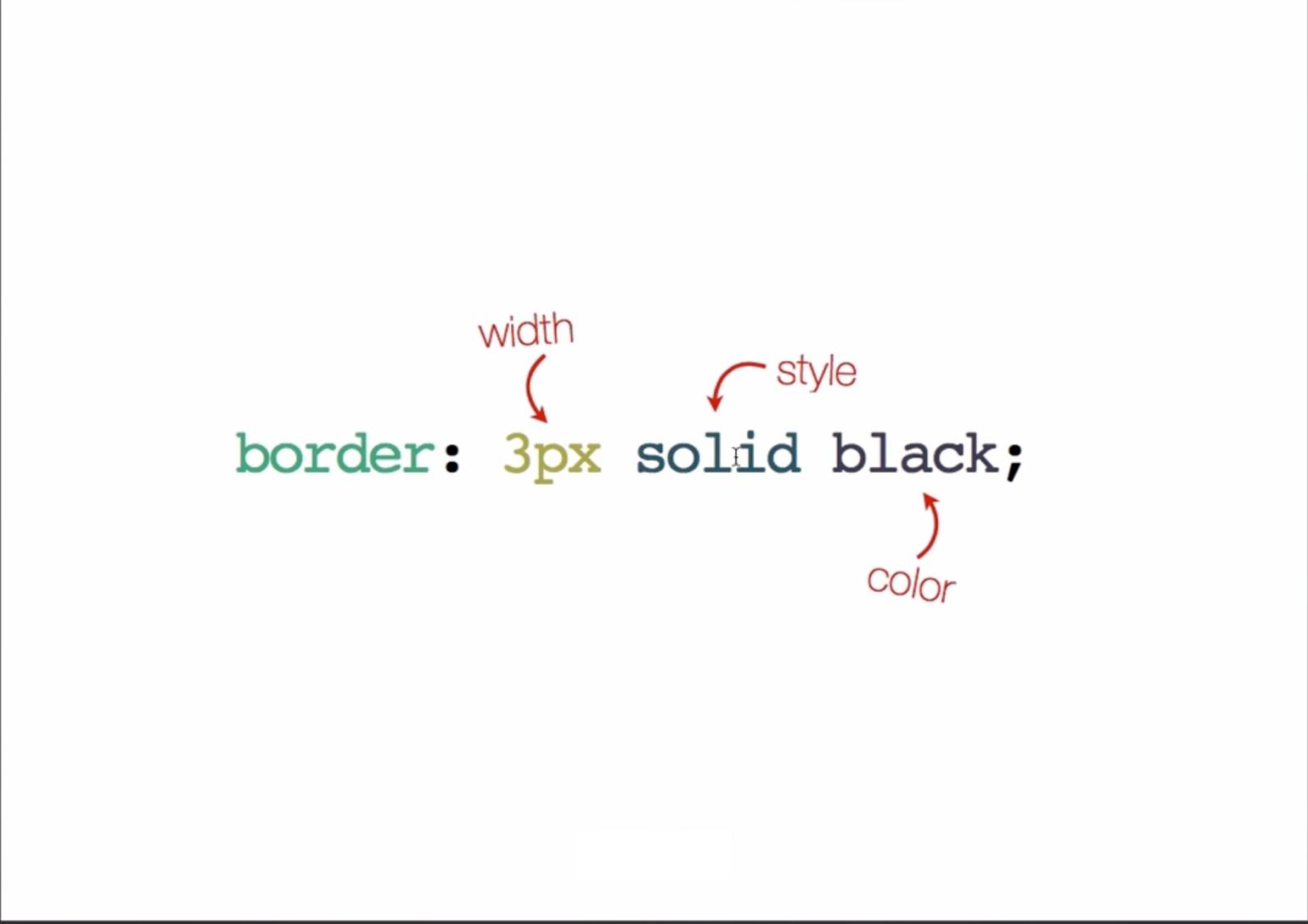
Margin

Border

Padding

Content

```
border: width style color;
```



```
border: 3px solid black;
```

width

style

)

color

The image shows a white rectangular box centered on a dark gray background. Above the box, there is a green line of CSS code: "border: 3px solid black;". Four red arrows point from the words "width", "style", ")", and "color" to the corresponding parts of the code: the "3px", the "solid black", the closing parenthesis ")", and the "border" keyword respectively.

# **style**

**solid**

**dotted**

**dashed**

**double**

# **css** reset

**css** float

“

Float merupakan properti pada CSS untuk mengatur posisi sebuah elemen. Sebuah elemen dapat dipaksa untuk berada di sebelah kiri atau kanan dari parent/pembungkusnya dengan menggunakan properti ini.

**float**

I  
none

left

right



**CSS** position

# **position**

static

relative

absolute

fixed

## CSS Position

---

- **static** adalah nilai default dari tiap-tiap elemen ketika tidak diberi properti position
- Menggunakan position selain **static** (non-static), akan membuat sebuah elemen menjadi seolah-olah **berbeda dimensi** dari elemen lainnya
- Elemen yang diberi position selain **static** dapat menggunakan properti **top**, **left**, **bottom** dan **right** untuk mengatur posisinya

## absolute

---

- Ketika kita menggerakkan elemen dengan posisi **absolute** (menggunakan properti **top**, **left**, **bottom** dan **right**), ruang yang ditempati oleh elemen tersebut dianggap **tidak ada**
- Ketika kita menggerakkan elemen dengan posisi absolute, elemen akan bergerak relatif terhadap posisi dari elemen parent-nya, **selama elemen parent-nya memiliki posisi yang juga non-static**
- Jika kita memberi properti **top: 0;** dan **left: 0;** maka elemen akan berada di ujung kiri atas dari elemen parent-nya

position: **fixed**;