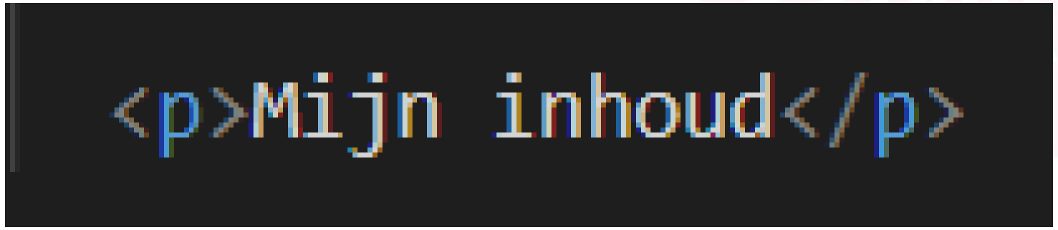
# Front-end development

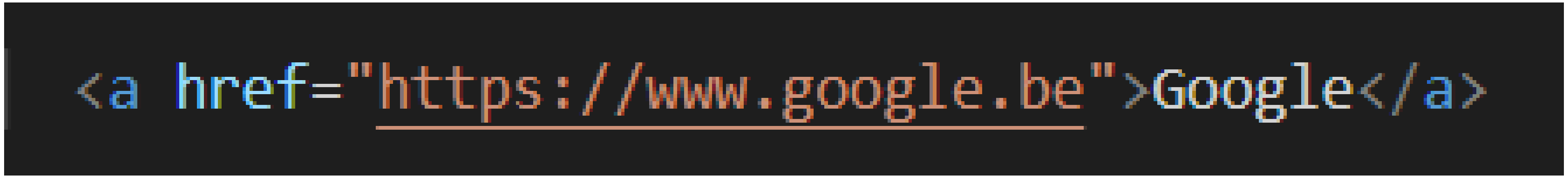
Front-end development – les 4

# HERHALING - HTML

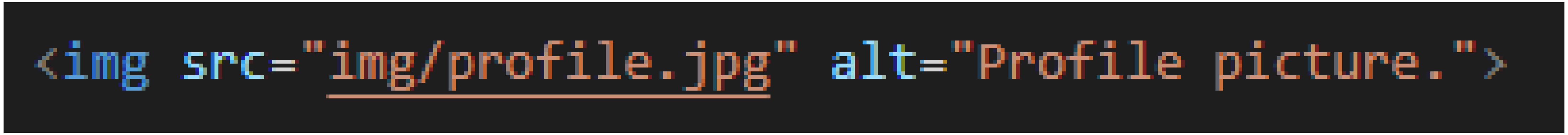
```
<!DOCTYPE html>
<html lang=“en”>
        <head>
        <body>
```



```
<p>Mijn inhoud</p>
```



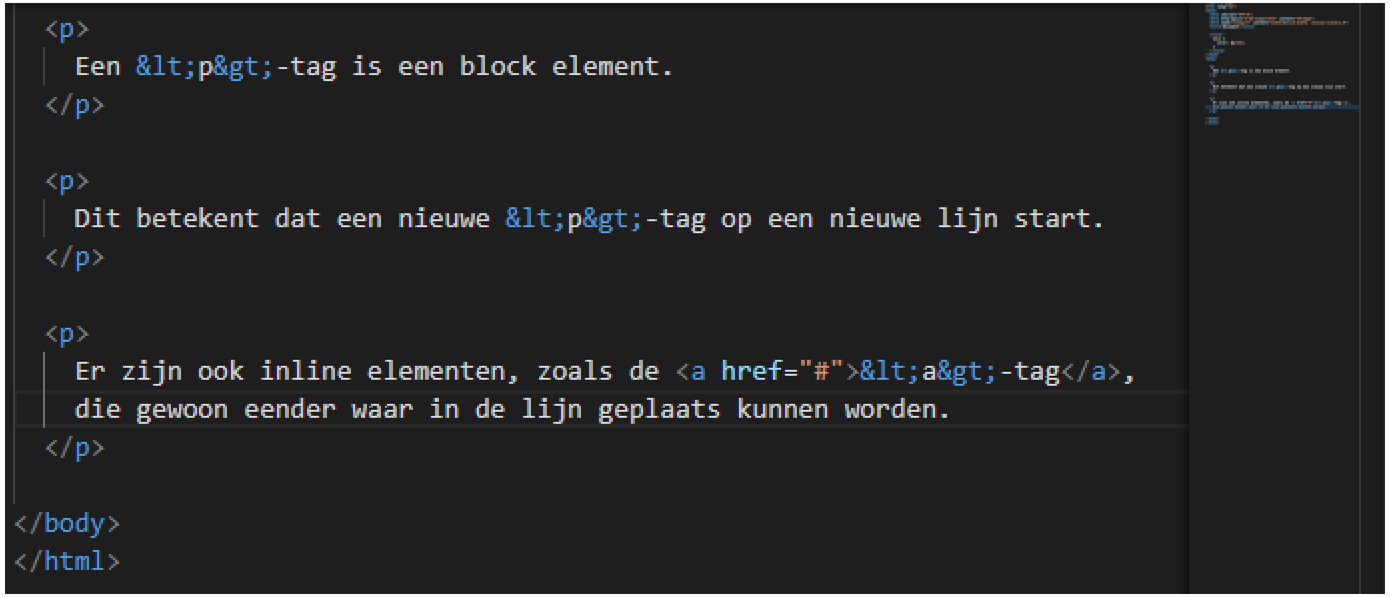
```
<a href="https://www.google.be">Google</a>
```



```
<img src="img/profile.jpg" alt="Profile picture.">
```

# INLINE VS BLOCK

- block: start op nieuwe regel
- inline: start eender waar op de huidige lijn

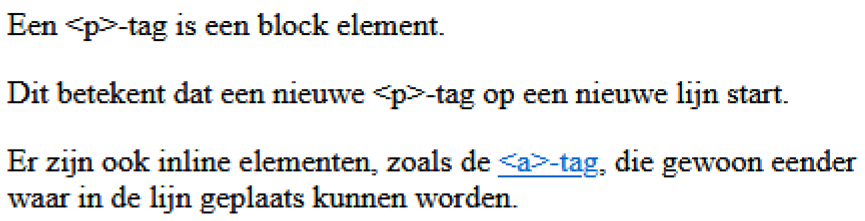


```
  <p>
    Een &lt;p&gt;-tag is een block element.
  </p>

  <p>
    Dit betekent dat een nieuwe &lt;p&gt;-tag op een nieuwe lijn start.
  </p>

  <p>
    Er zijn ook inline elementen, zoals de <a href="#">&lt;a&gt;-tag</a>,
    die gewoon eender waar in de lijn geplaats kunnen worden.
  </p>

</body>
</html>
```



Een <p>-tag is een block element.

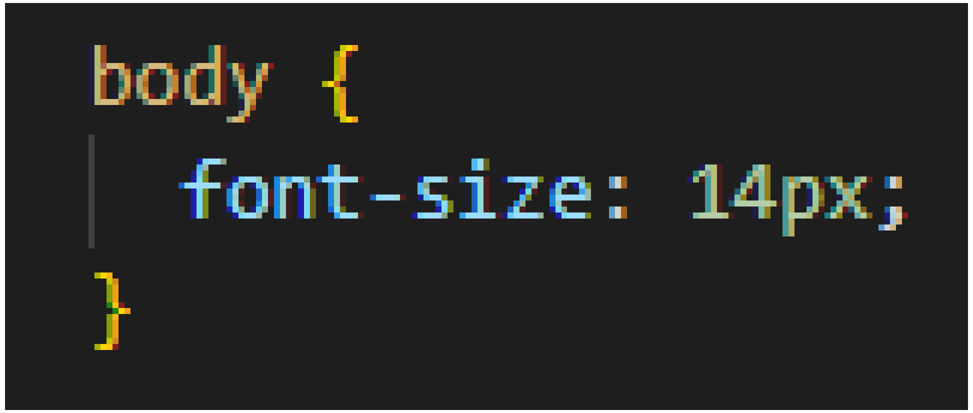
Dit betekent dat een nieuwe <p>-tag op een nieuwe lijn start.

Er zijn ook inline elementen, zoals de <a>-tag, die gewoon eender waar in de lijn geplaats kunnen worden.

# STYLING - CSS

- Cascading Stylesheets

- Selector
- Declaration
- Property
- Value



```
body {
  font-size: 14px;
}
```

# STYLING - CSS

- 3 manieren:
  - Inline
  - Embedded
  - External stylesheet

- Selectors: tag, class, id, attribute, pseudo-classes
- Overerving

# STYLING

- Tekstkleur: color
- Achtergrond-kleur: background-color
- Achtergrond-afbeelding: background-image
- Lettertype: font-family

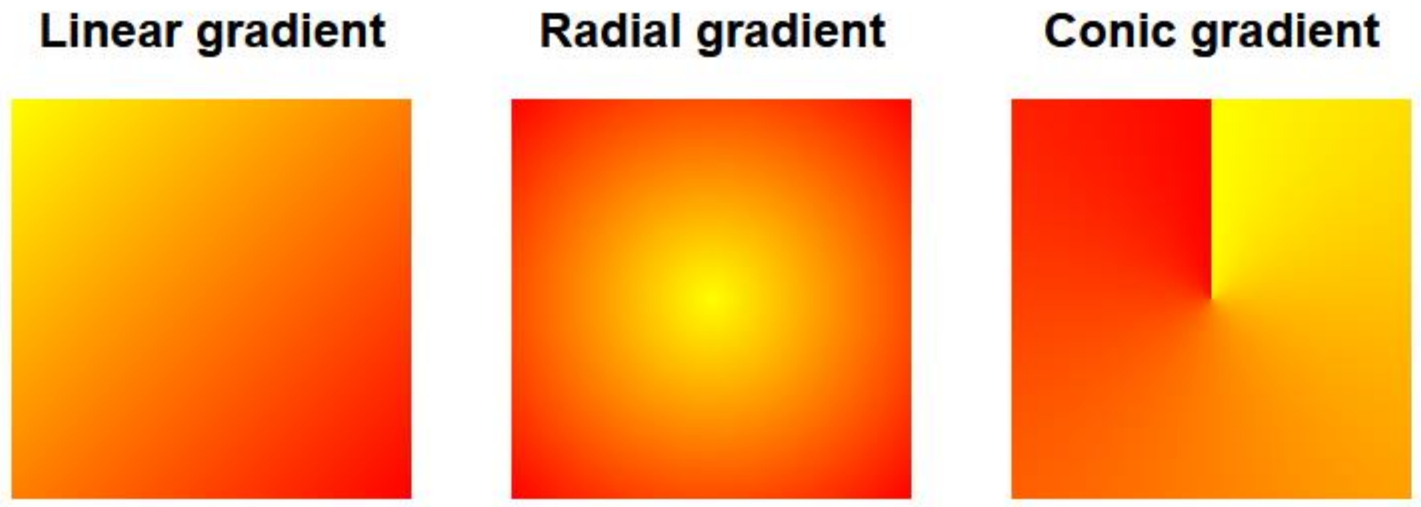
- Units: absolute (px, pt, …) relative (%, em, rem, vh, …)

# SPECIFICITY

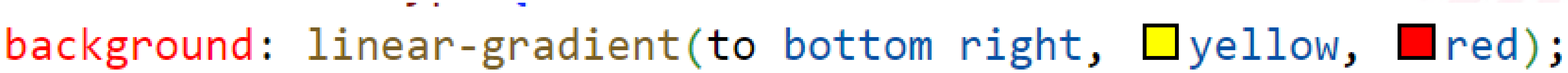
- ID, Class, Type: x,x,x
- ID: Only element Ids (#example)
- Class: classes, attribute selectors and pseudo classes
- Type: Tag names

- Add 1 to column for each selector

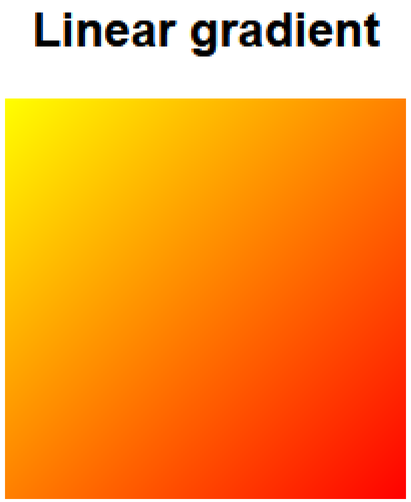
# STYLING - GRADIENTS



# STYLING – LINEAR GRADIENT

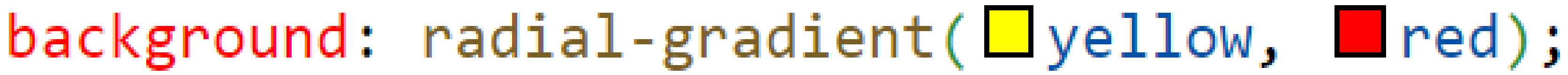


```
background: linear-gradient(to bottom right, 🟨yellow, 🟥red);
```

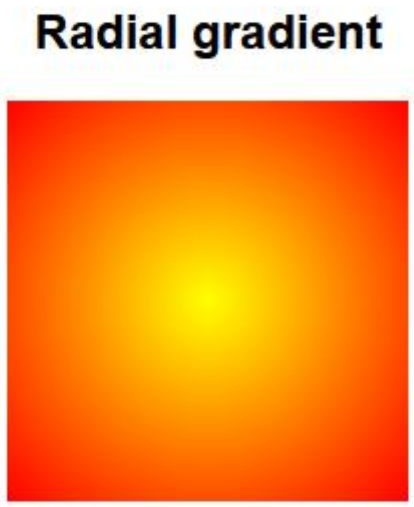


**Linear gradient**

# STYLING – RADIAL GRADIENT

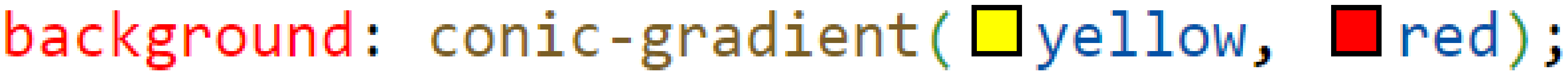


```
background: radial-gradient(🟨yellow, 🟥red);
```



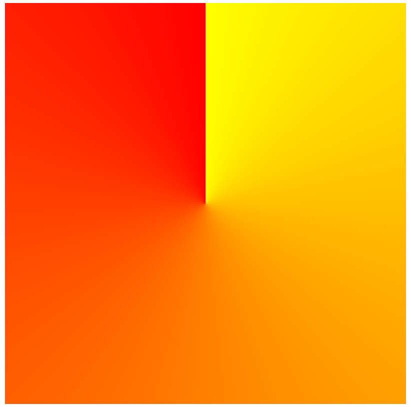
Radial gradient

# STYLING – CONIC GRADIENT

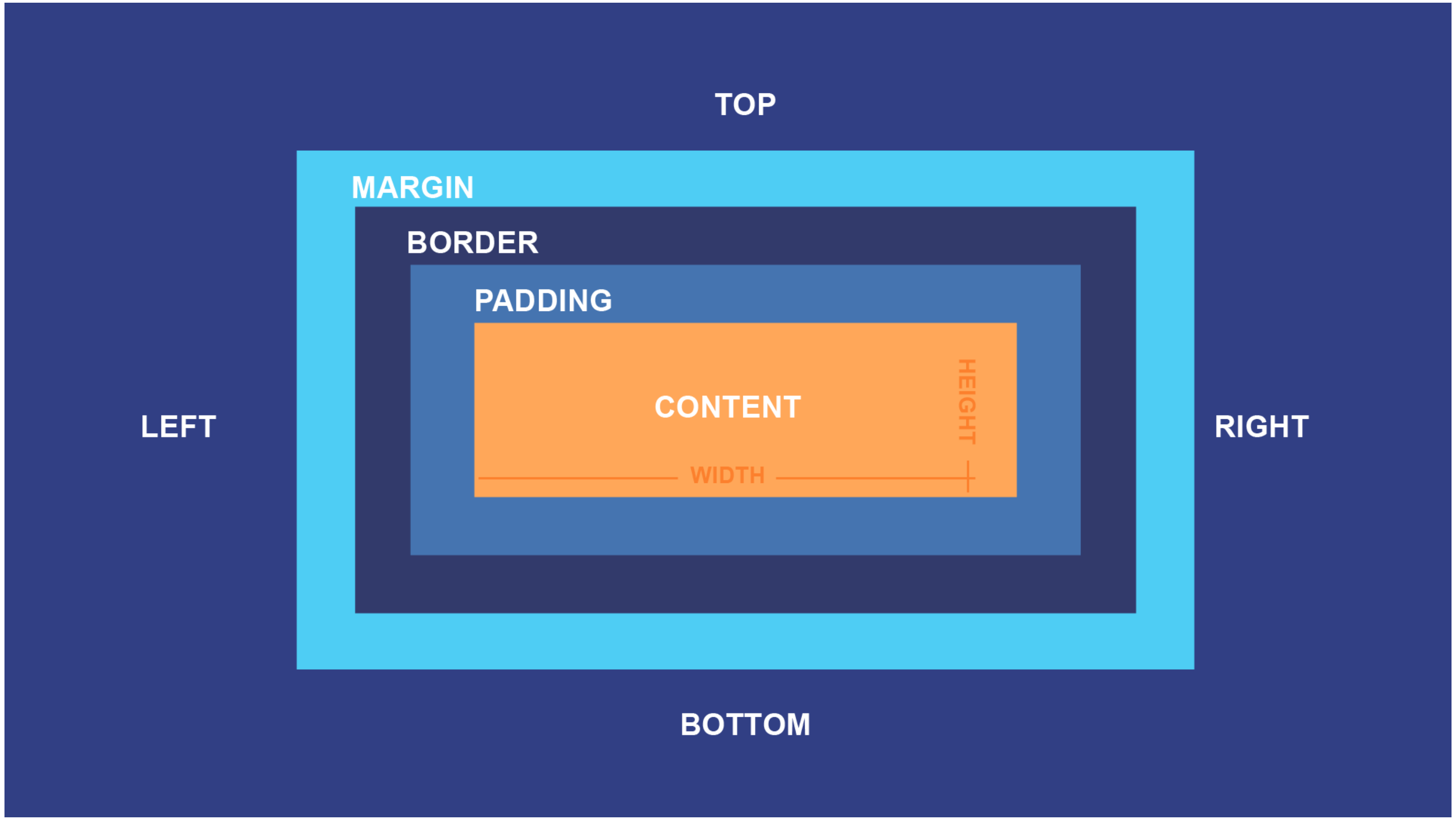


```
background: conic-gradient(🟨yellow, 🟥red);
```

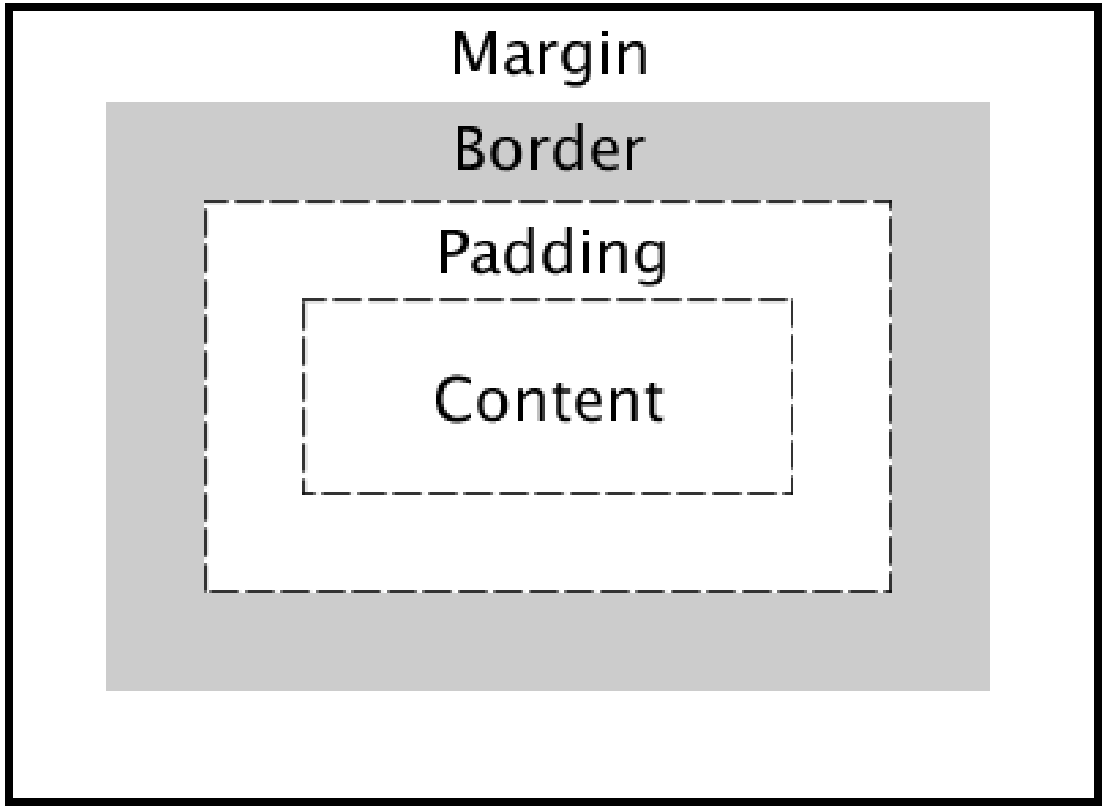
**Conic gradient**



# BOX MODEL

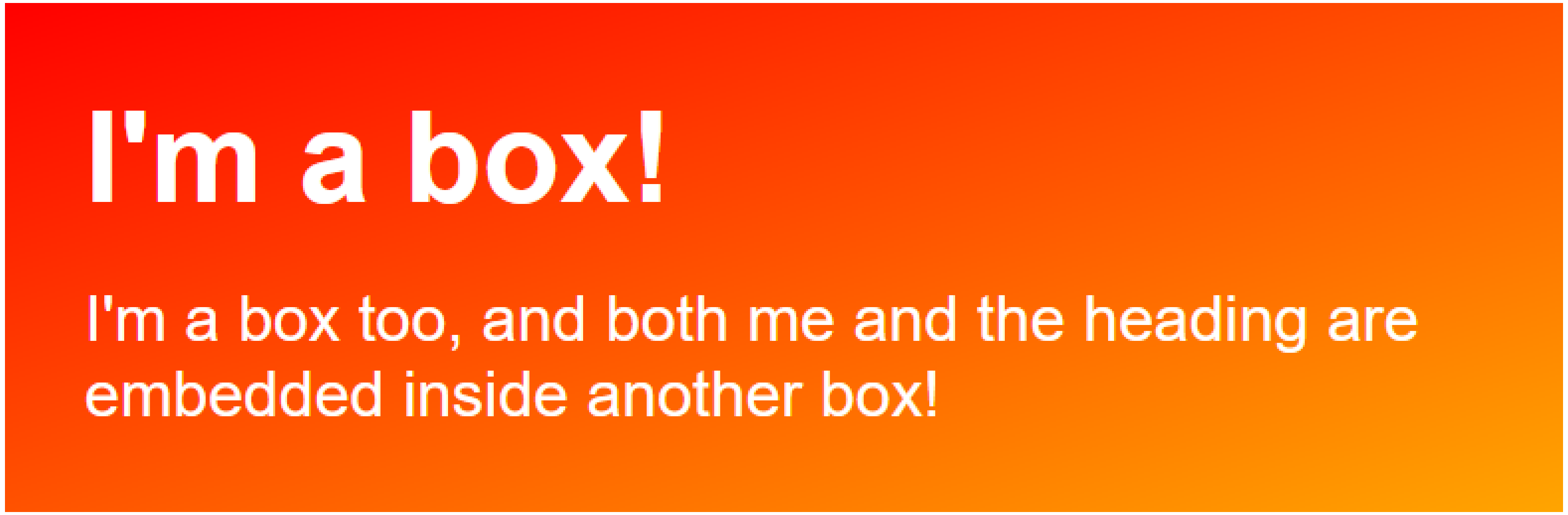

TOP
MARGIN
BORDER
PADDING
CONTENT
HEIGHT
WIDTH
LEFT
RIGHT
BOTTOM

# BOX MODEL



# STYLING – BOX MODEL

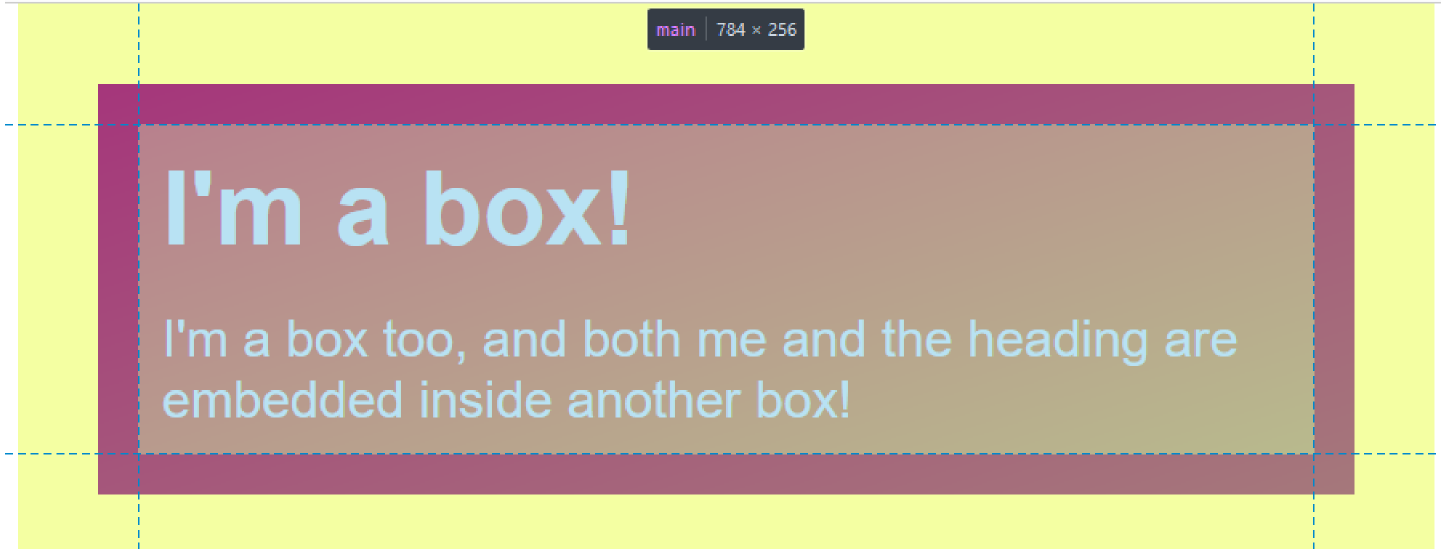
- Alles in CSS is een box
- Dit is niet hetzelfde als block vs inline!



# I'm a box!

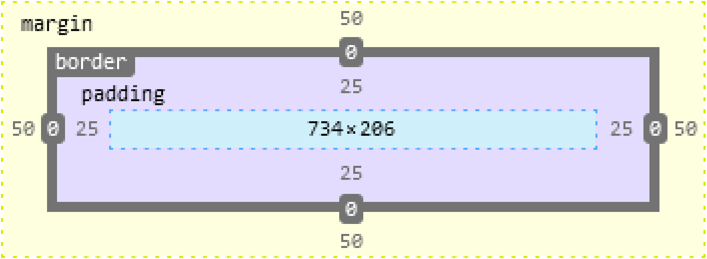
I'm a box too, and both me and the heading are embedded inside another box!

# STYLING – BOX MODEL



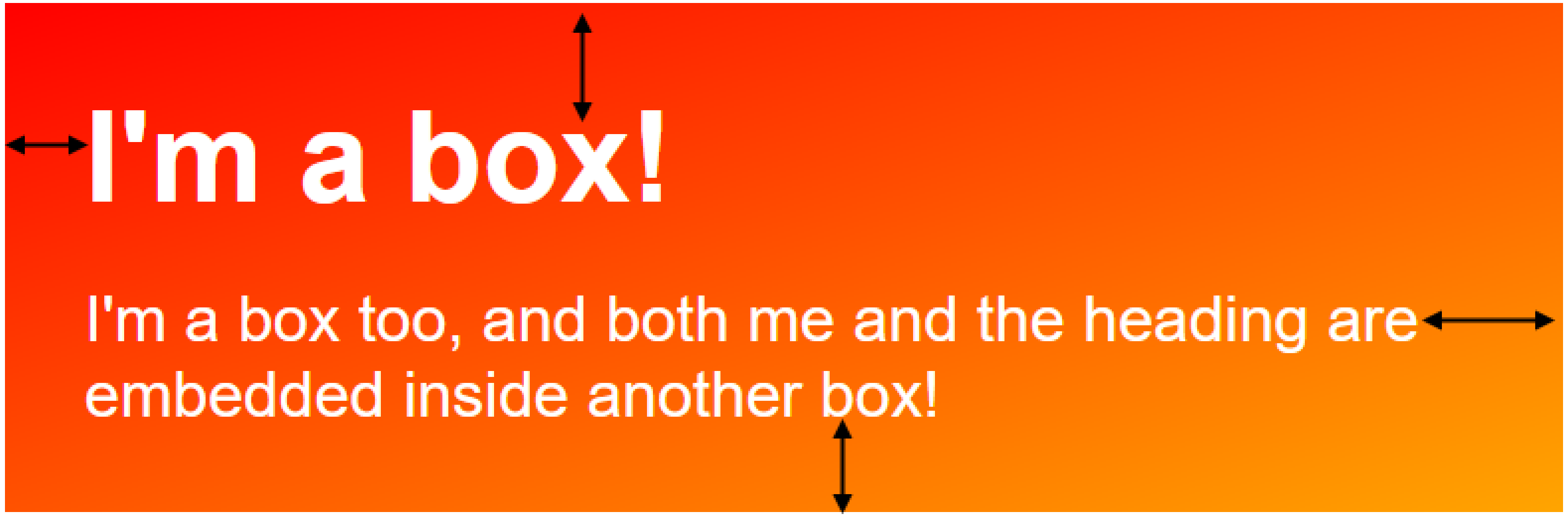
I'm a box!

I'm a box too, and both me and the heading are embedded inside another box!



# STYLING – PADDING

- Ruimte rond de inhoud

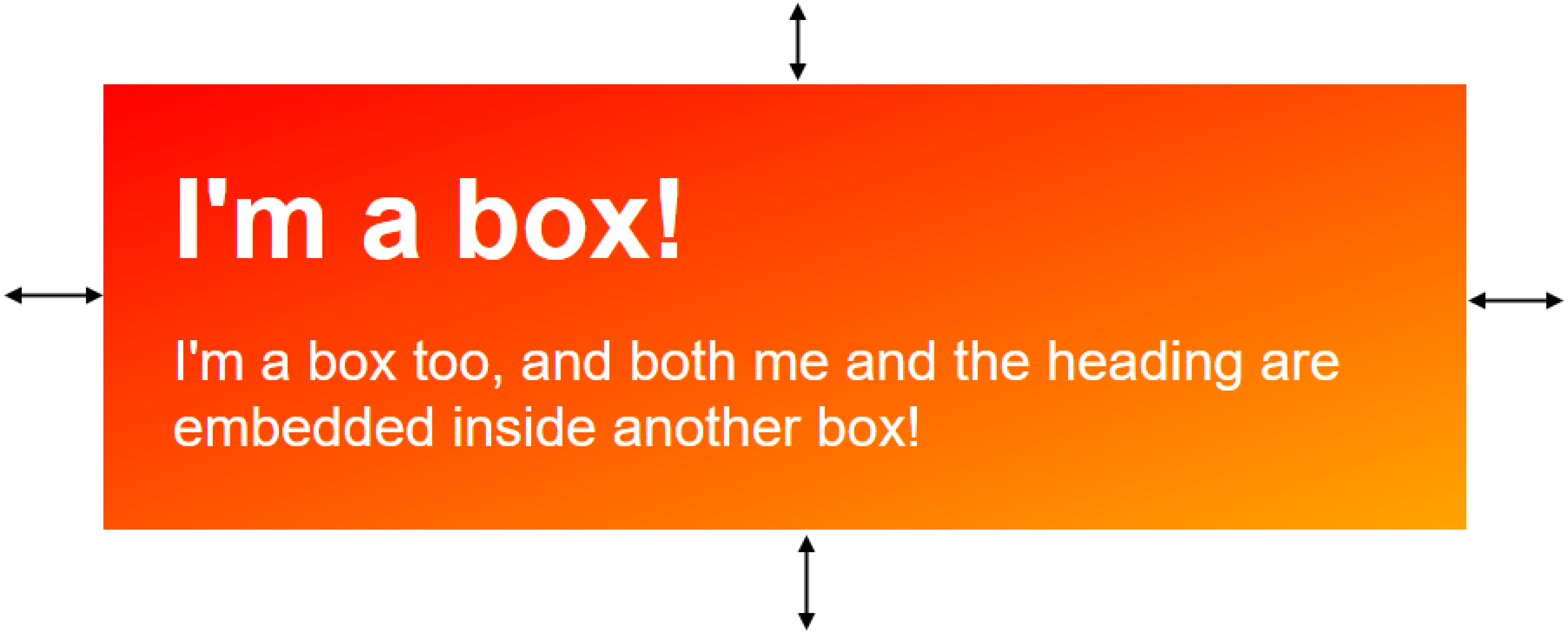


I'm a box!

I'm a box too, and both me and the heading are embedded inside another box!

# STYLING - MARGIN

- Ruimte rond het element

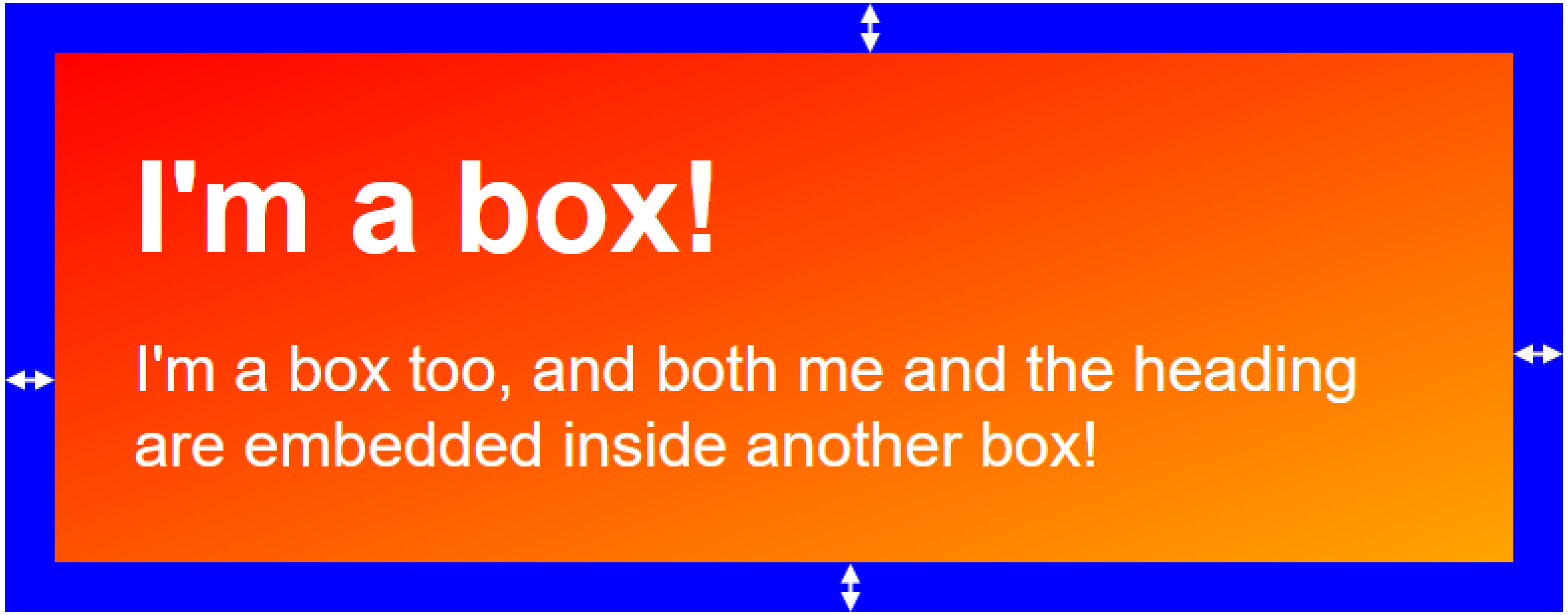


I'm a box!

I'm a box too, and both me and the heading are embedded inside another box!

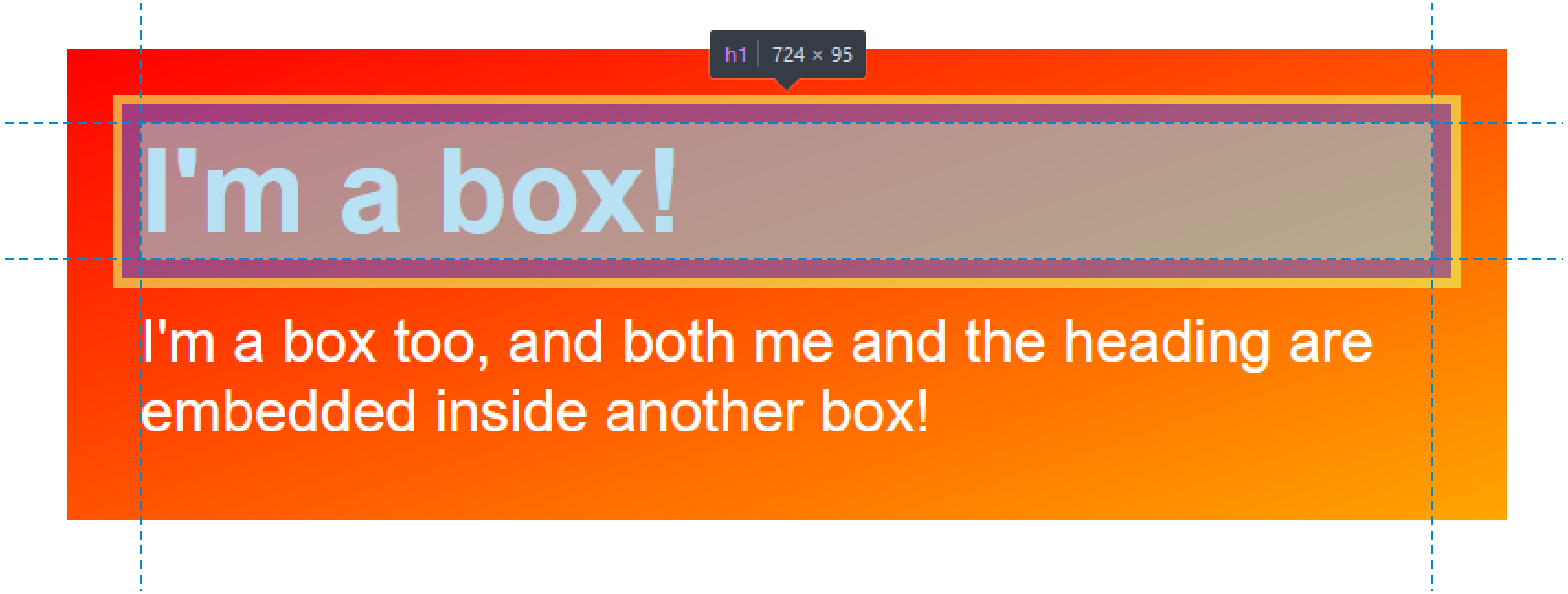
# STYLING – BORDER

- Ruimte tussen margin en padding



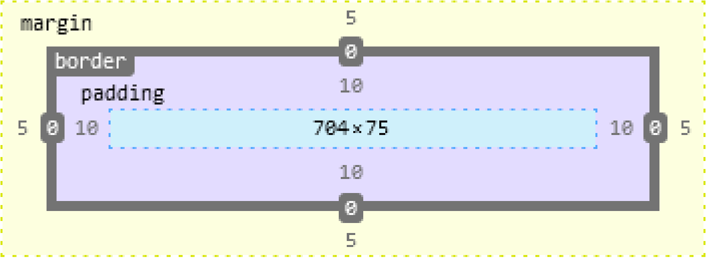
I'm a box!
I'm a box too, and both me and the heading are embedded inside another box!

# STYLING – BOX MODEL

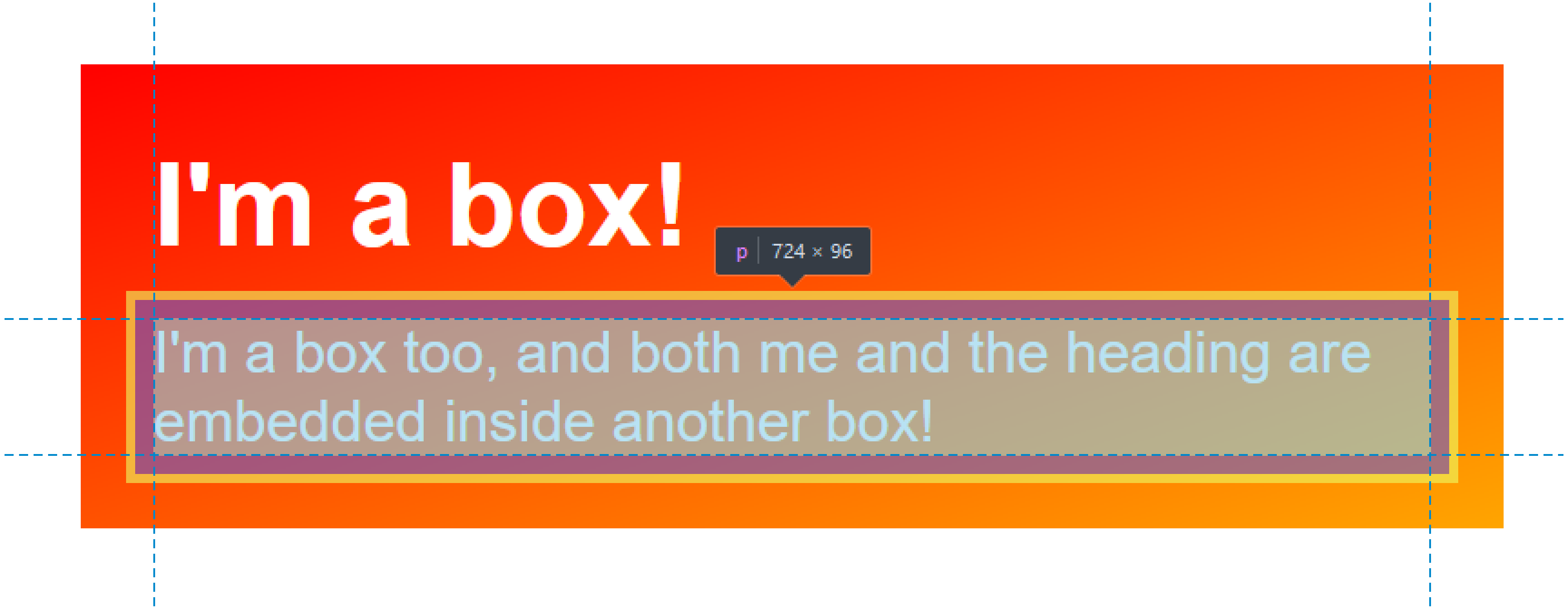


# I'm a box!

I'm a box too, and both me and the heading are embedded inside another box!

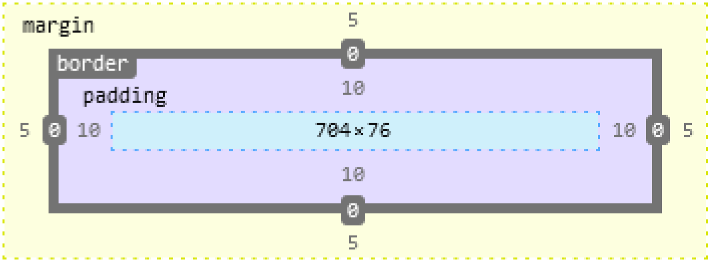


# STYLING – BOX MODEL

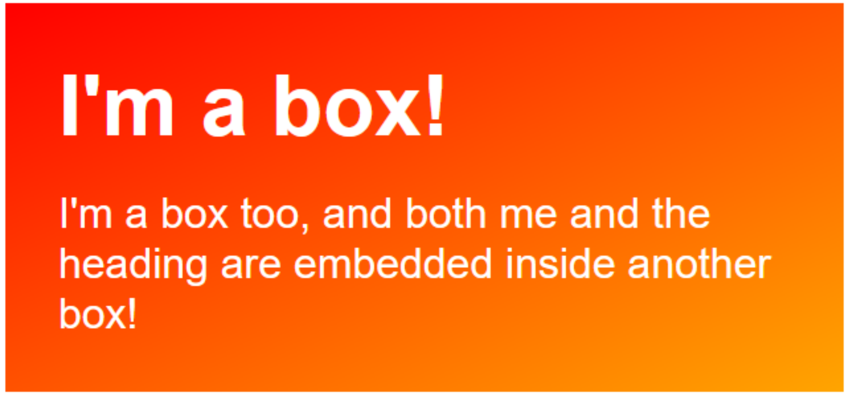


I'm a box!

I'm a box too, and both me and the heading are embedded inside another box!

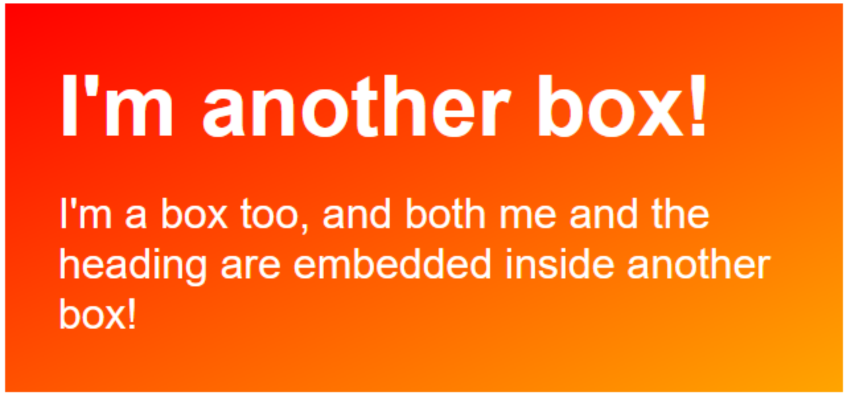


# STYLING – COLLAPSIBLE MARGIN



## I'm a box!

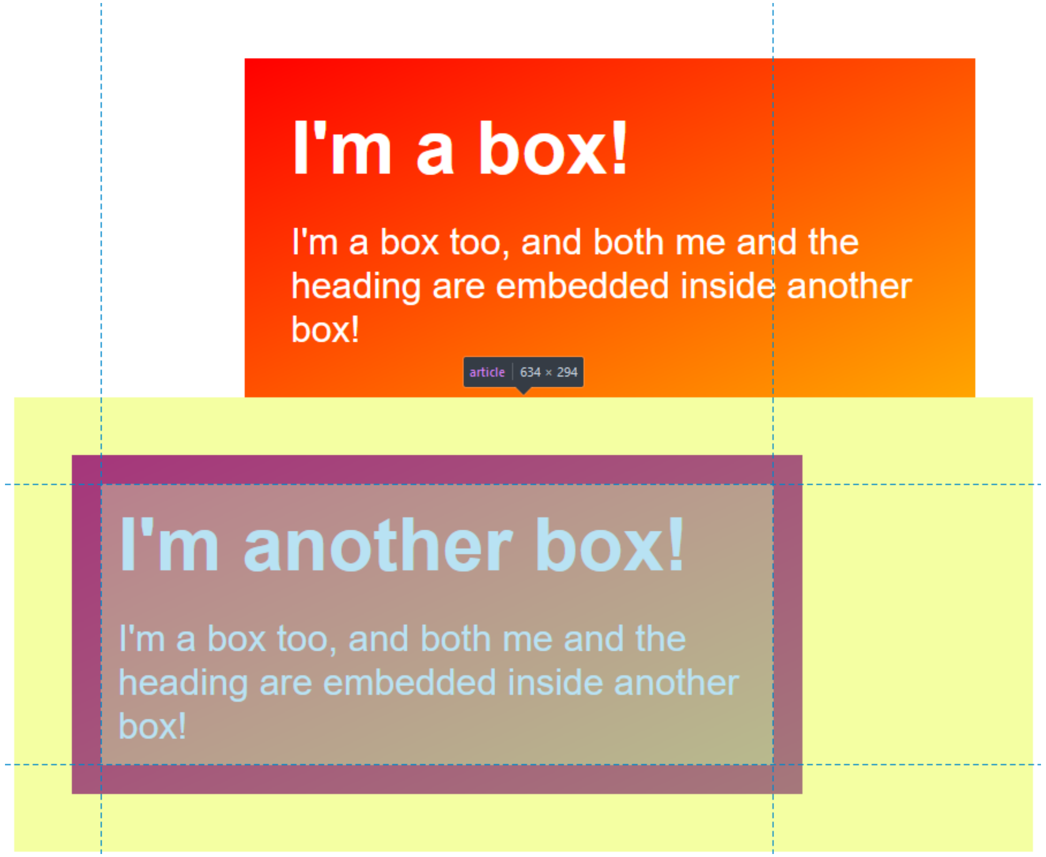
I'm a box too, and both me and the heading are embedded inside another box!



## I'm another box!

I'm a box too, and both me and the heading are embedded inside another box!

# STYLING – COLLAPSIBLE MARGIN

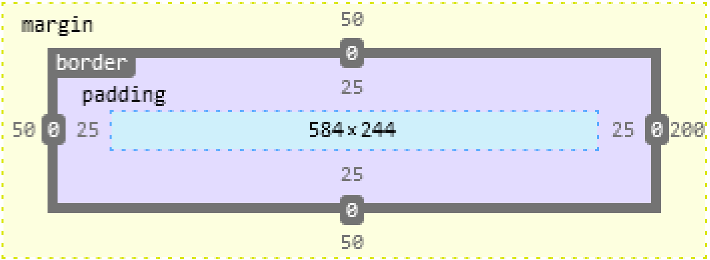


I'm a box!

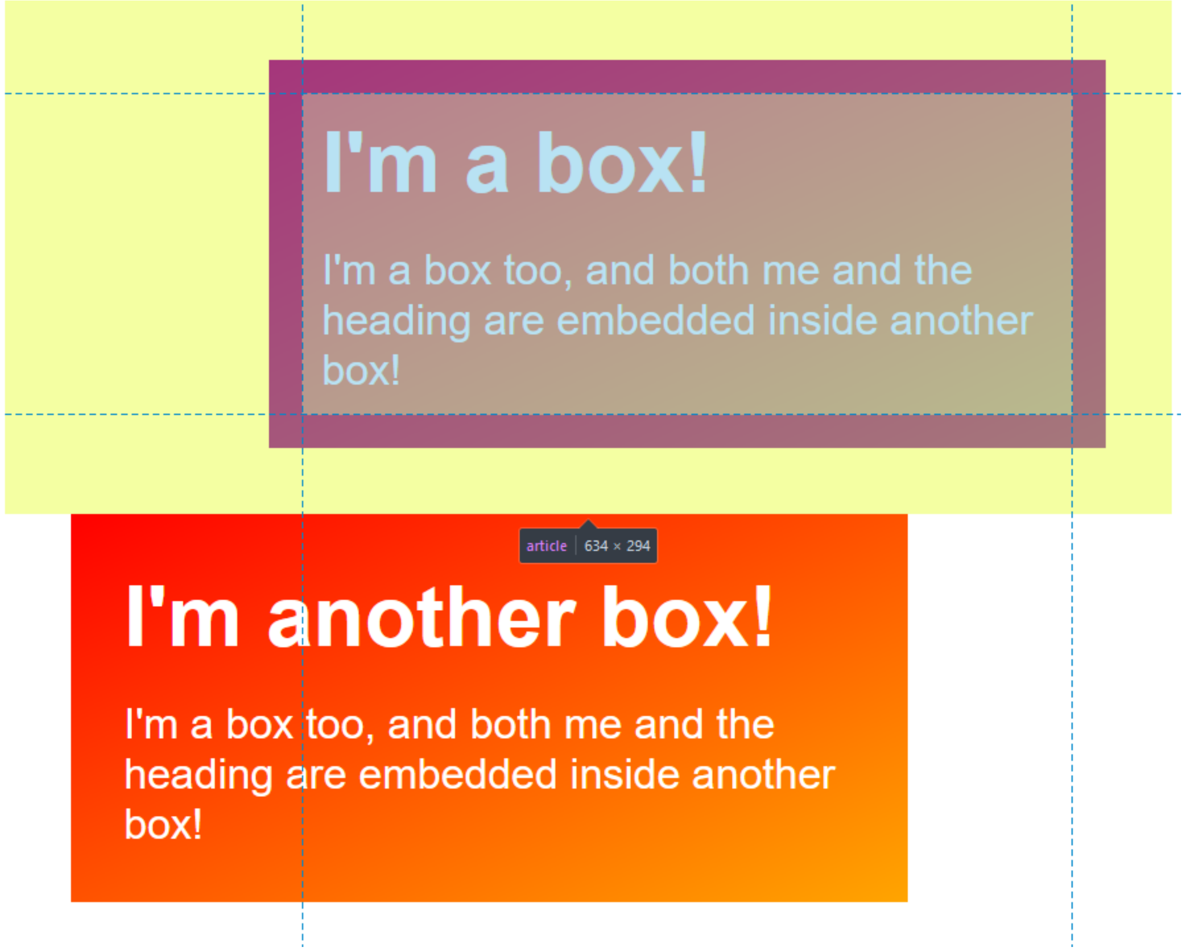
I'm a box too, and both me and the heading are embedded inside another box!

I'm another box!

I'm a box too, and both me and the heading are embedded inside another box!



# STYLING – COLLAPSIBLE MARGIN

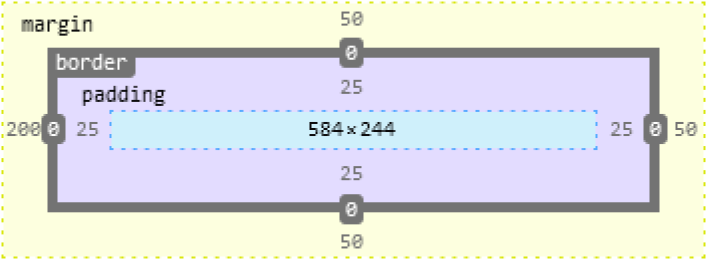


I'm a box!

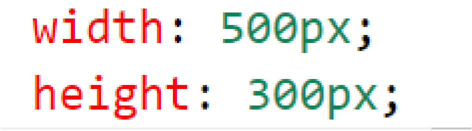
I'm a box too, and both me and the heading are embedded inside another box!

I'm another box!

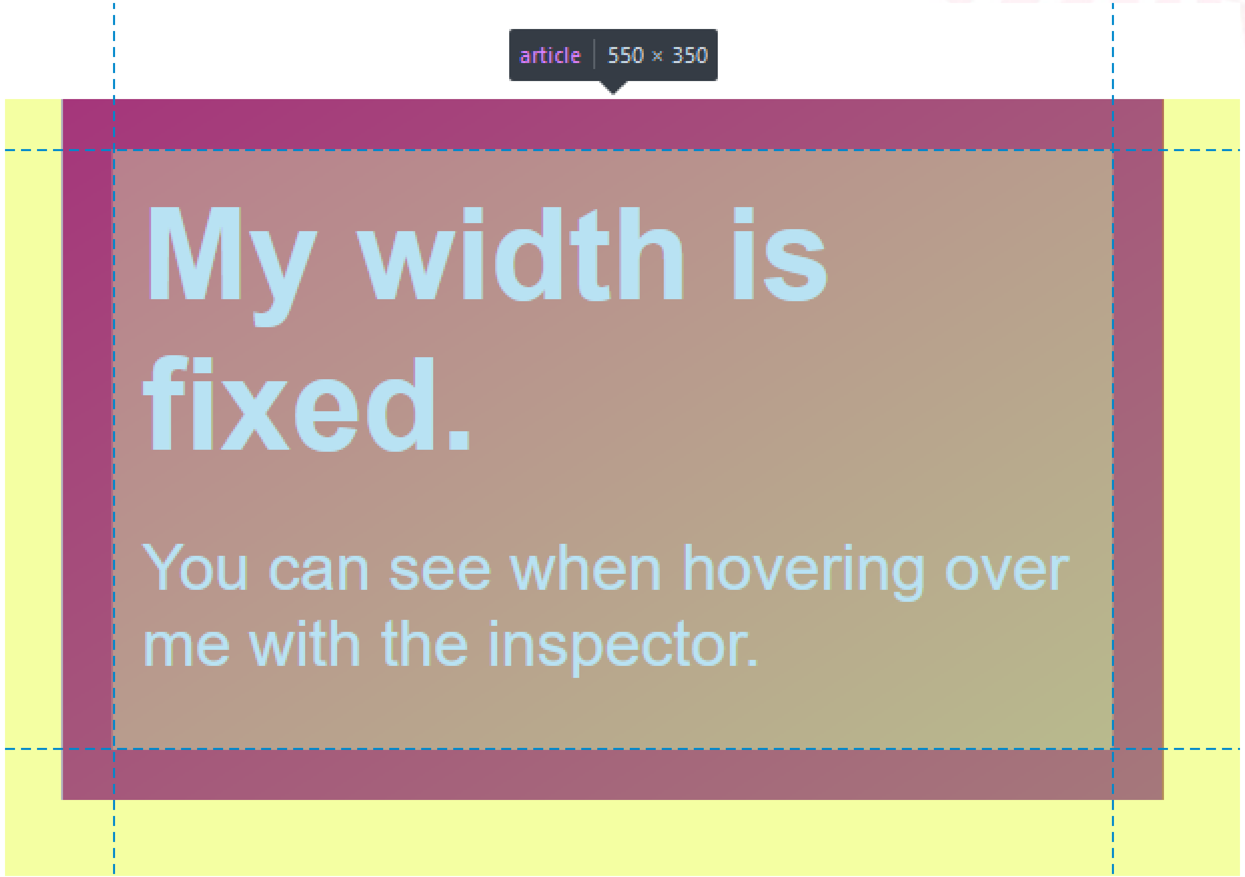
I'm a box too, and both me and the heading are embedded inside another box!



# STYLING – WIDTH & HEIGHT



```
width: 500px;
height: 300px;
```

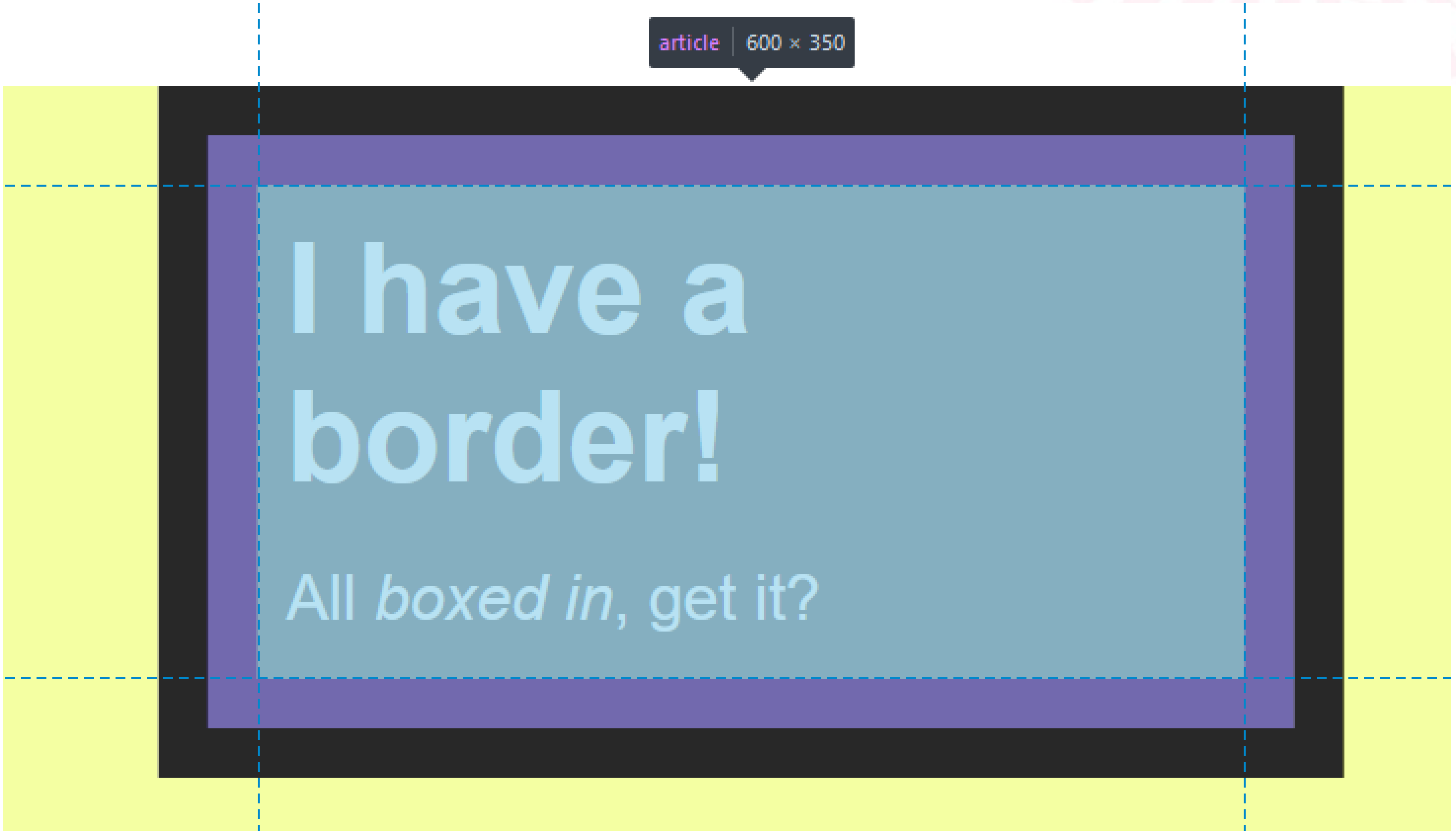


## My width is fixed.

You can see when hovering over me with the inspector.

# STYLING – WIDTH & HEIGHT

```
border: solid ■black 25px;
padding: 25px;
width: 500px;
height: 250px;
```

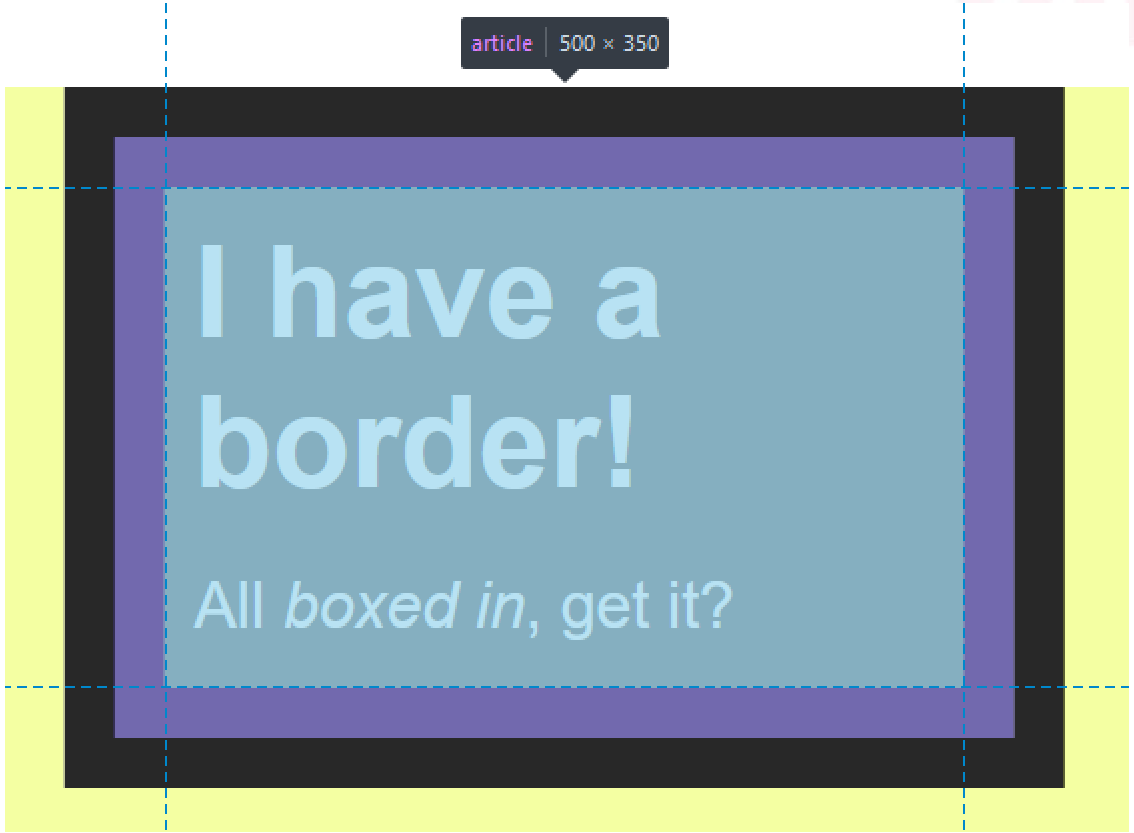


I have a border!

All *boxed in*, get it?

# STYLING – BOX-SIZING

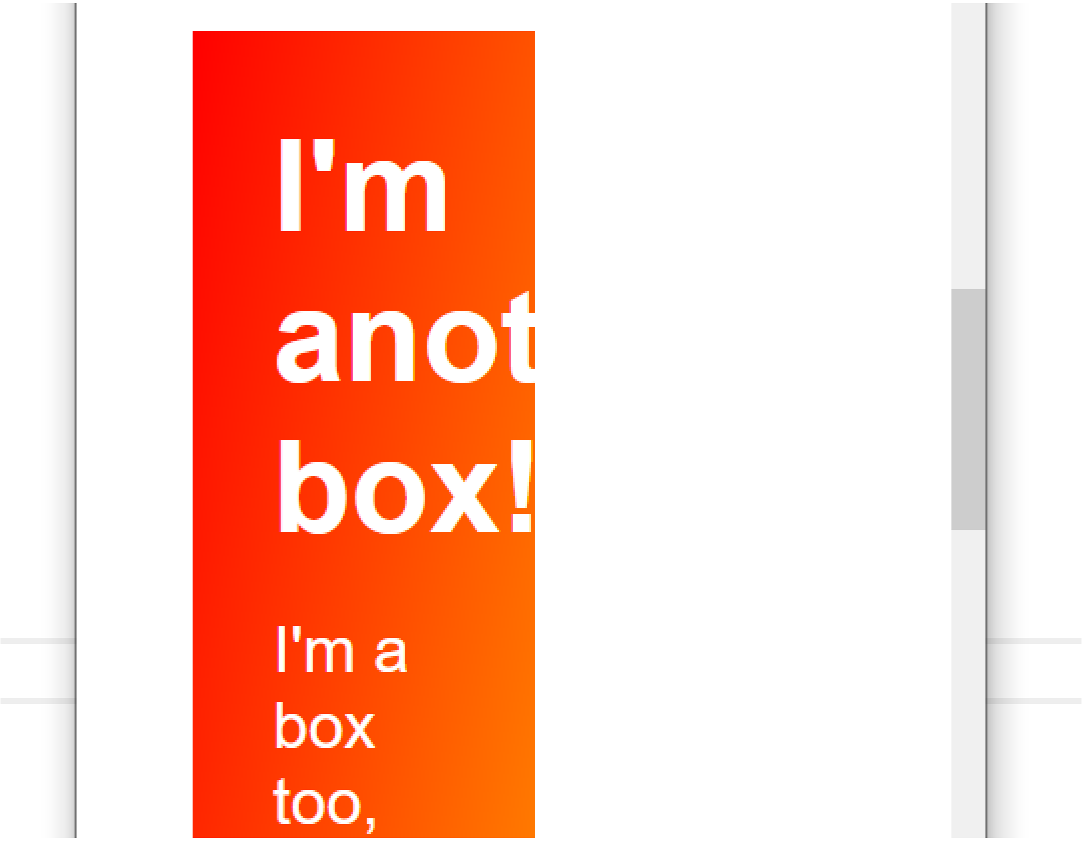
```
border: solid ■black 25px;
padding: 25px;
width: 500px;
height: 350px;
box-sizing: border-box;
```



I have a border!

All *boxed in*, get it?

# STYLING – MIN WIDTH

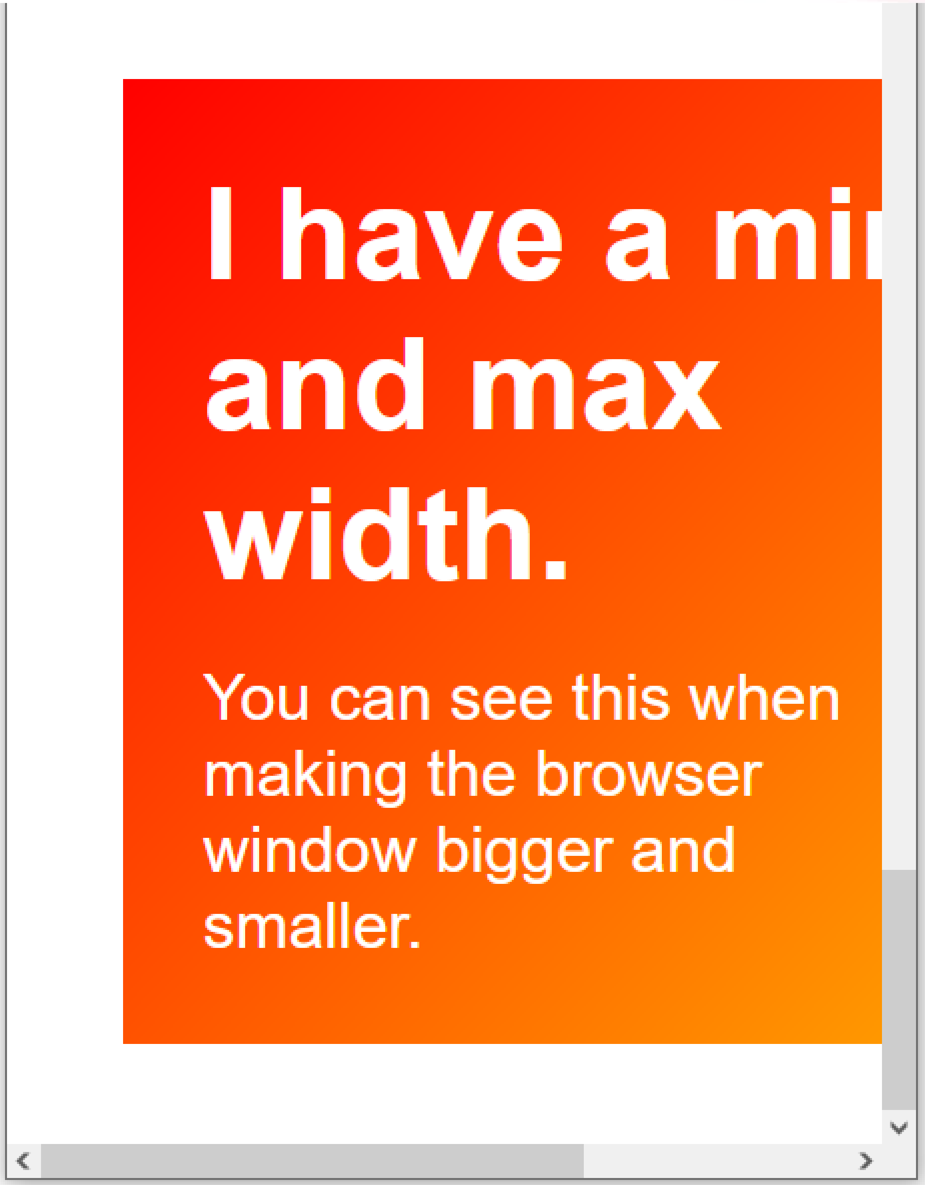


# I'm anot box!

I'm a box too,

# STYLING – MIN WIDTH

```
min-width: 400px;
```



I have a mi

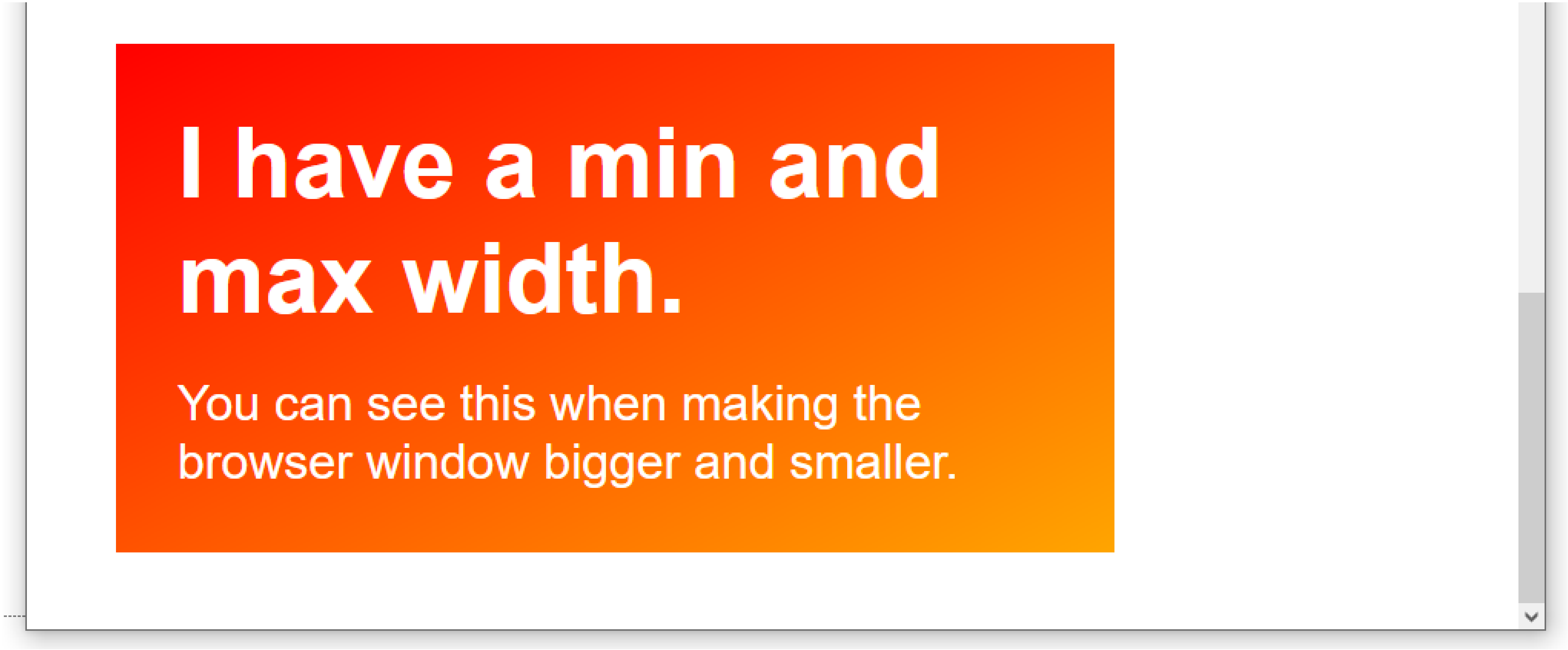
and max

width.

You can see this when making the browser window bigger and smaller.

# STYLING – MAX WIDTH

```
max-width: 600px;
```



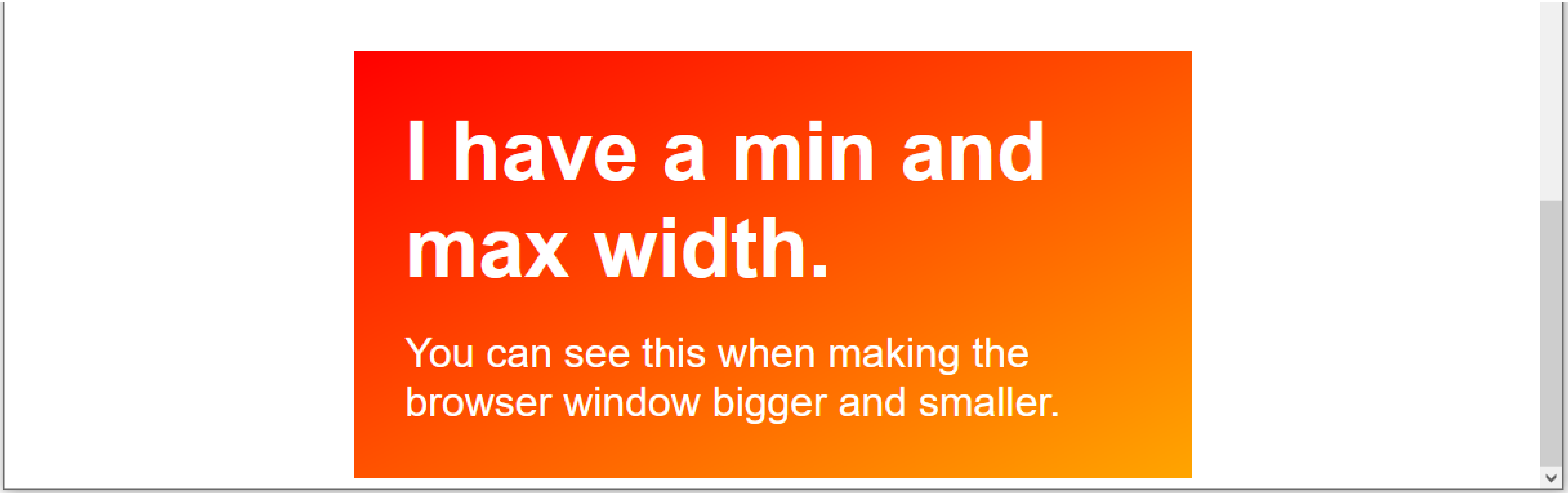
I have a min and max width.

You can see this when making the browser window bigger and smaller.

# STYLING – MARGIN AUTO

- Elementen met width, max-width, mooi centreren

```
max-width: 600px;
min-width: 400px;
margin: 0 auto;
```



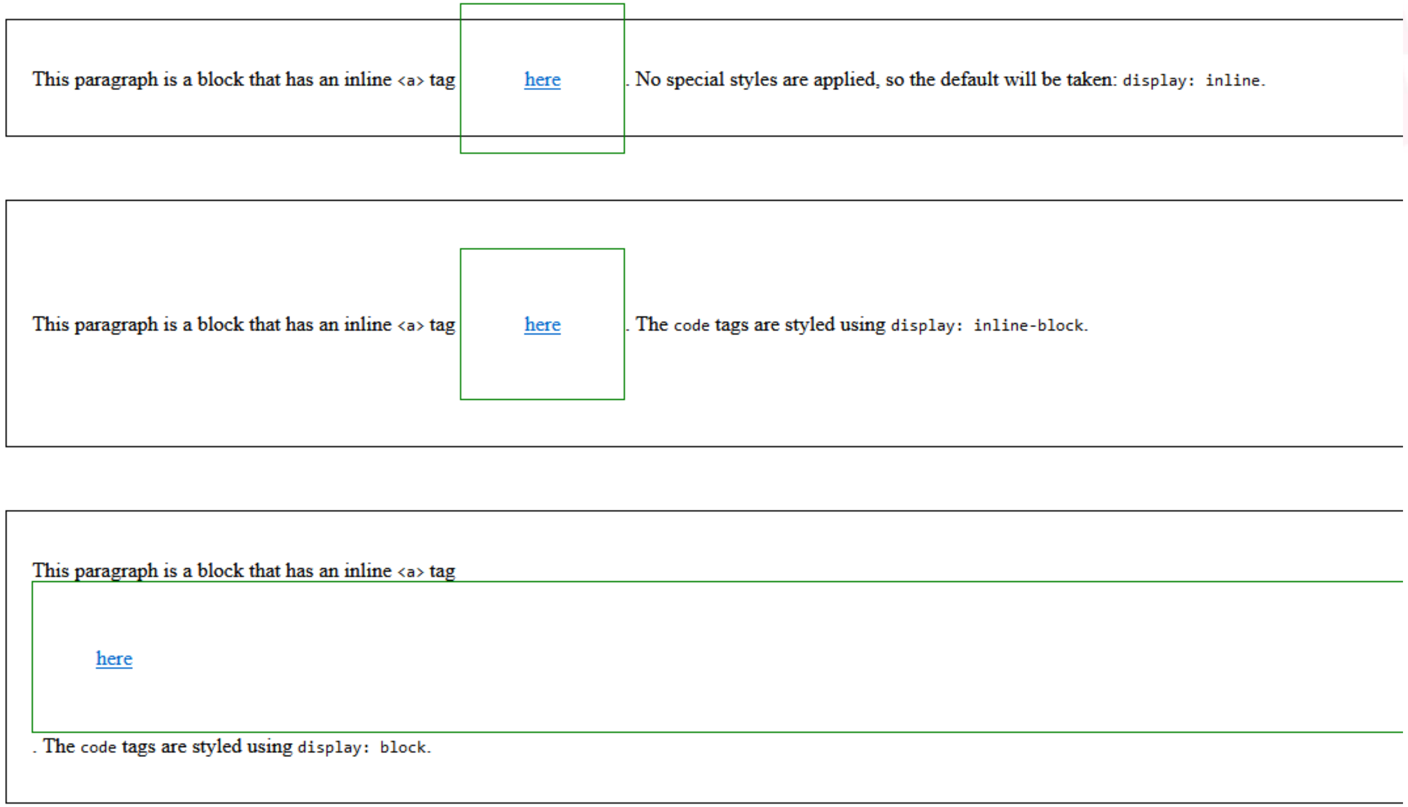
I have a min and max width.

You can see this when making the browser window bigger and smaller.

# STYLING – BOX MODEL - OEFENING

# STYLING – DISPLAY

- block
- inline
- inline-block



This paragraph is a block that has an inline `<a>` tag here . No special styles are applied, so the default will be taken: `display: inline`.

This paragraph is a block that has an inline `<a>` tag here . The `code` tags are styled using `display: inline-block`.

This paragraph is a block that has an inline `<a>` tag

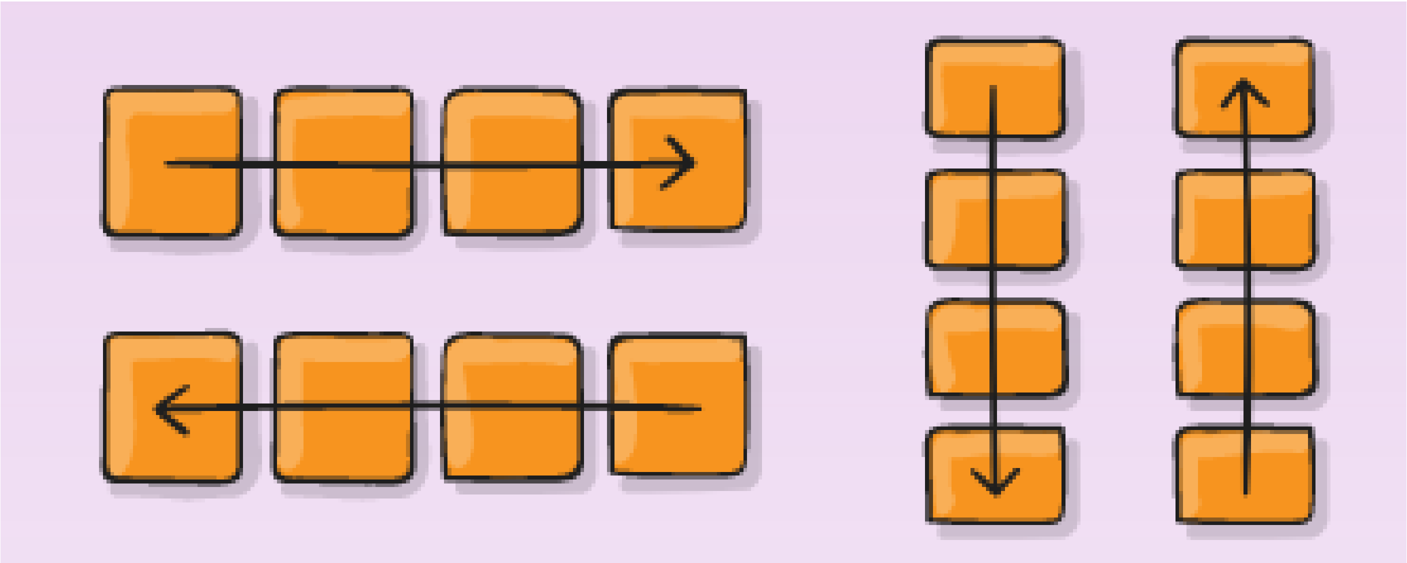
here

. The `code` tags are styled using `display: block`.

https://developer.mozilla.org/en-US/docs/Web/CSS/display

# STYLING - FLEXBOX

- Positioneren van elementen in 1 richting

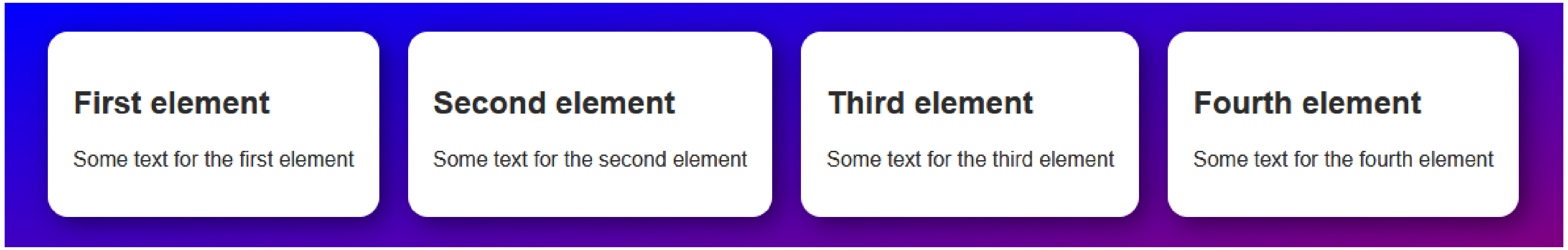


https://css-tricks.com/snippets/css/a-guide-to-flexbox/

# STYLING – FLEXBOX

- display: flex
- flex-direction: row | column | row-reverse | column-reverse
- flex-wrap: nowrap | wrap | wrap-reverse

# STYLING – FLEXBOX



**First element**

Some text for the first element

**Second element**

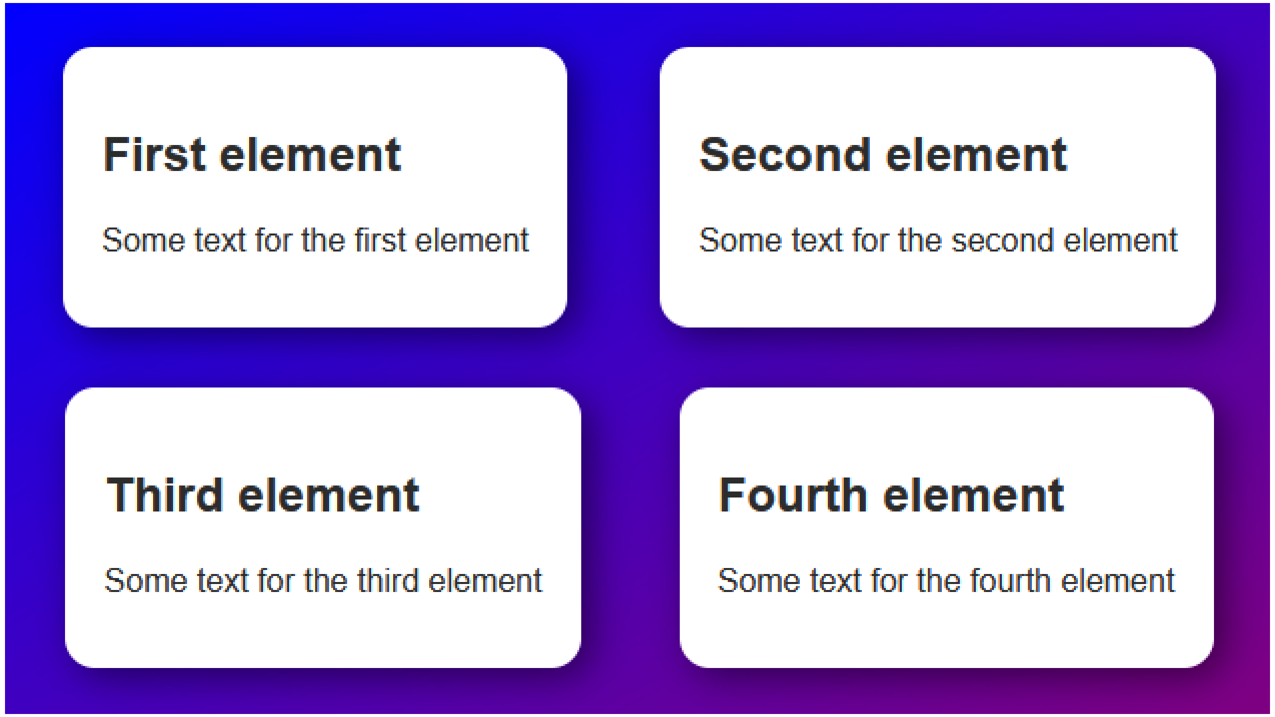
Some text for the second element

**Third element**

Some text for the third element

**Fourth element**

Some text for the fourth element



**First element**

Some text for the first element

**Second element**

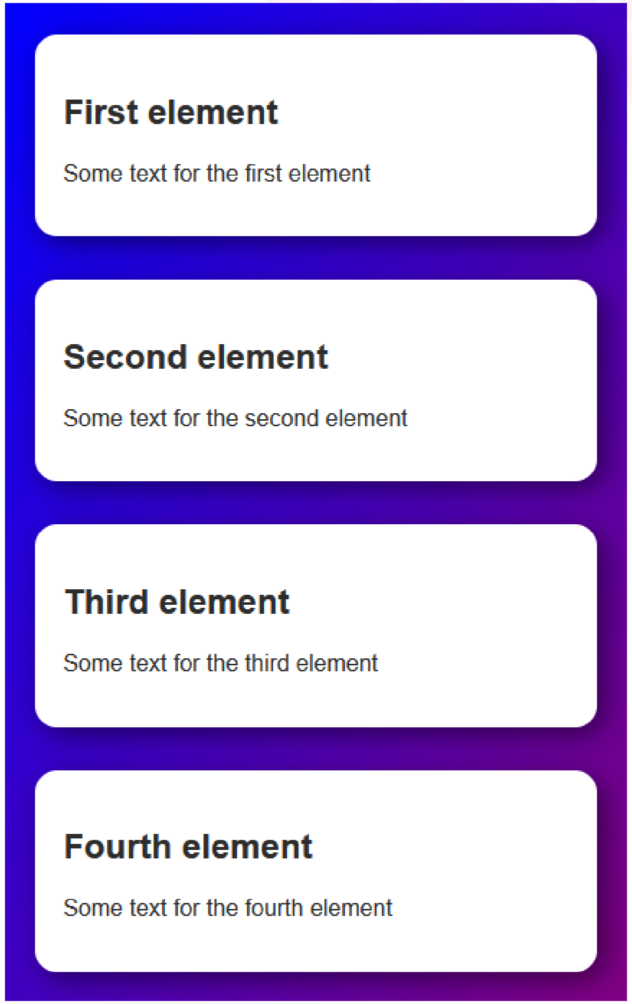
Some text for the second element

**Third element**

Some text for the third element

**Fourth element**

Some text for the fourth element



**First element**

Some text for the first element

**Second element**

Some text for the second element

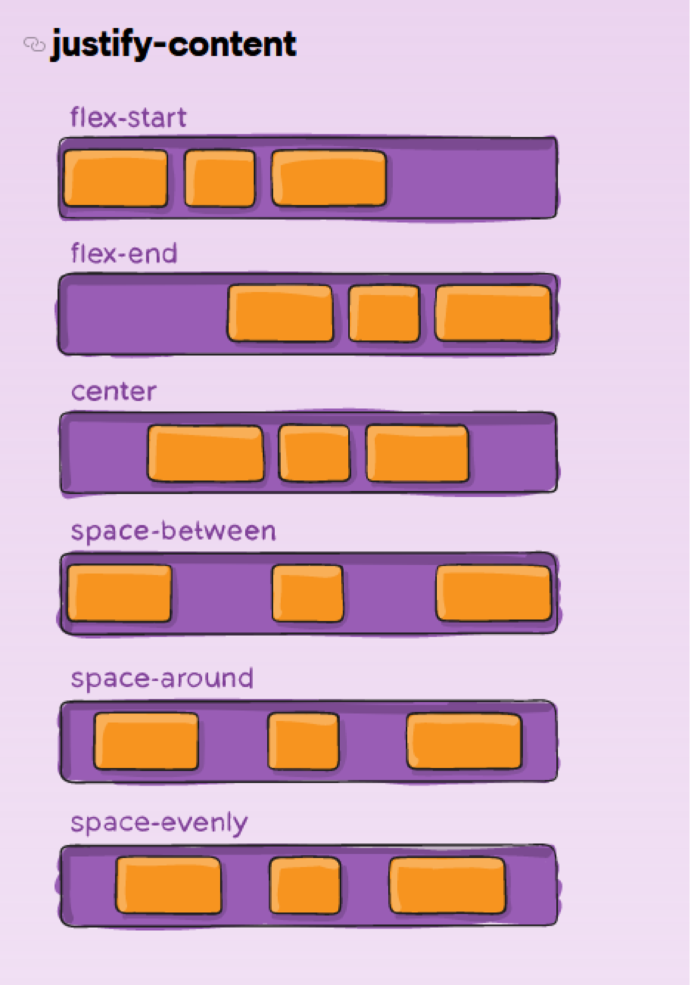
**Third element**

Some text for the third element

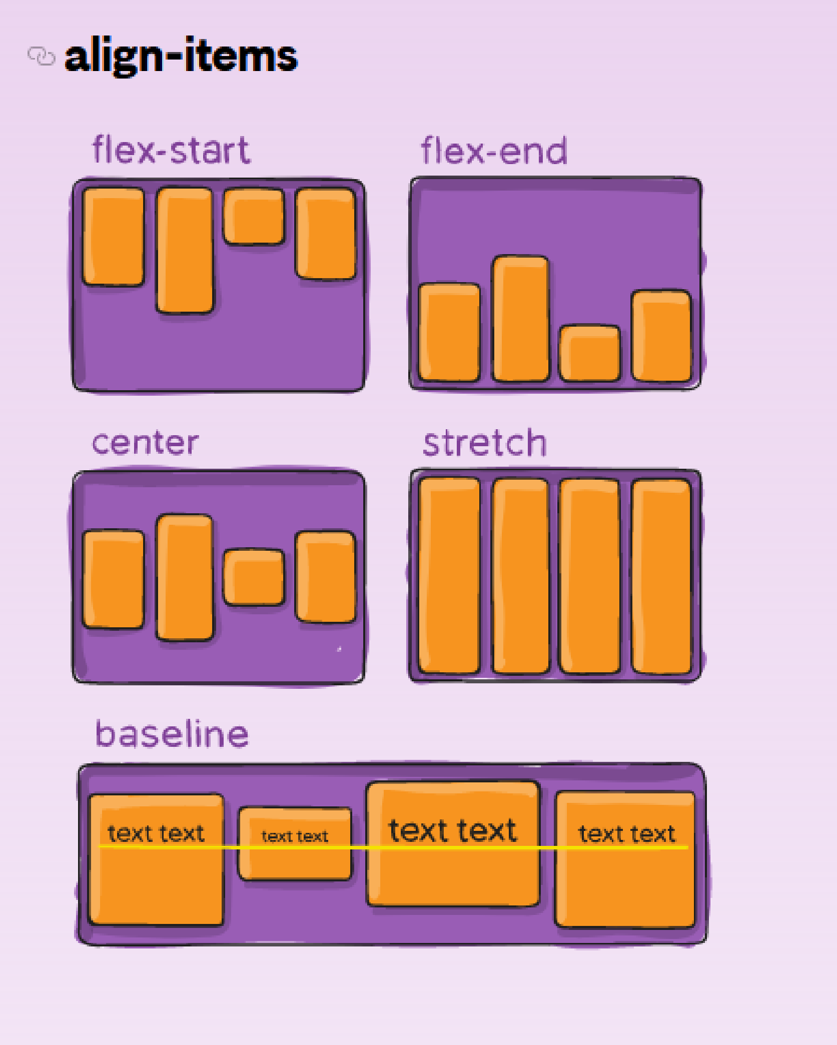
**Fourth element**

Some text for the fourth element

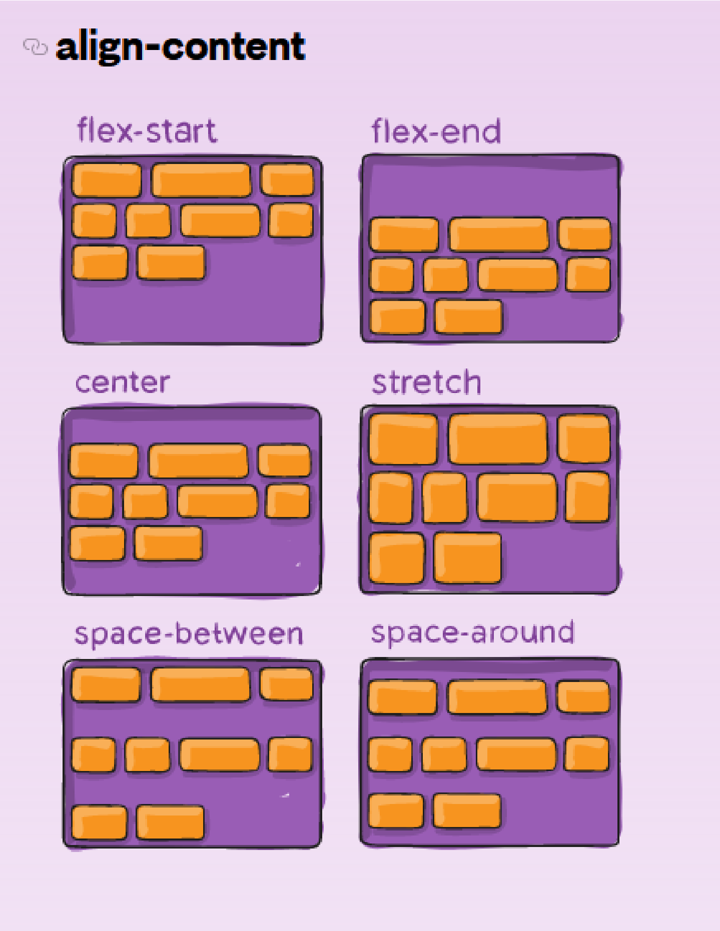
# STYLING – FLEXBOX – PARENT



# STYLING – FLEXBOX – PARENT

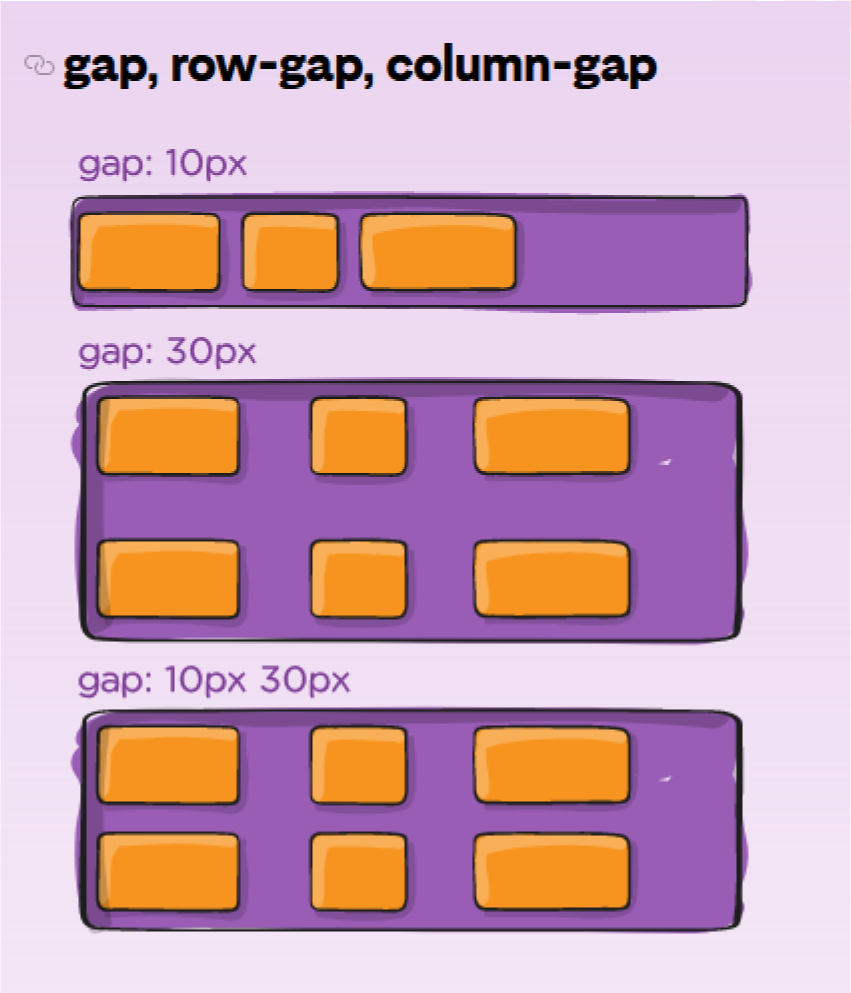


# STYLING – FLEXBOX – PARENT

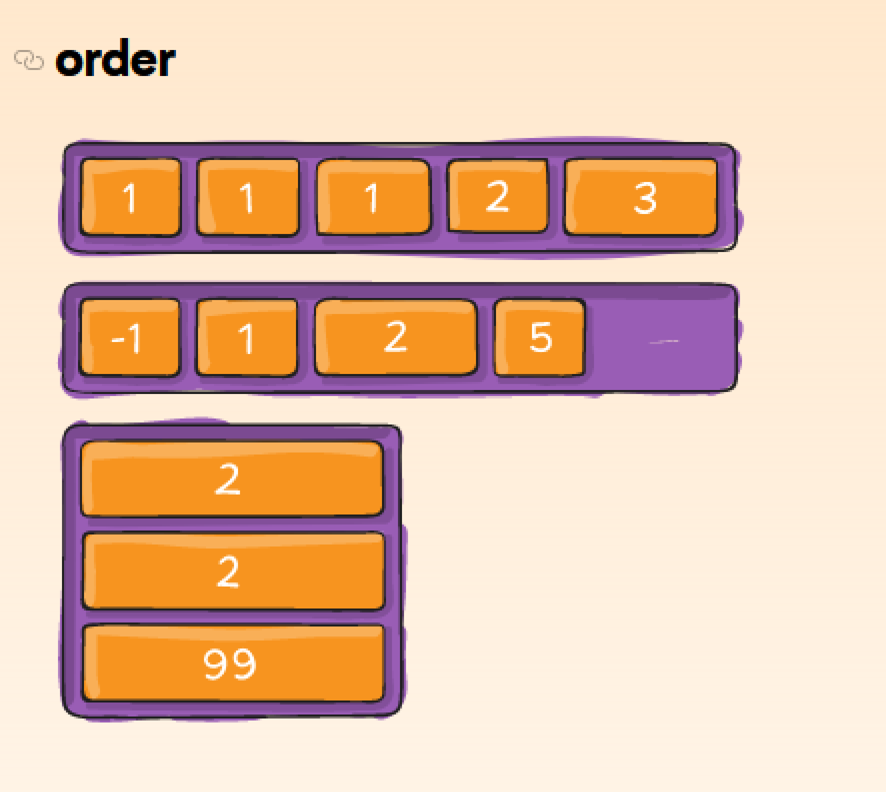


# STYLING – FLEXBOX – PARENT

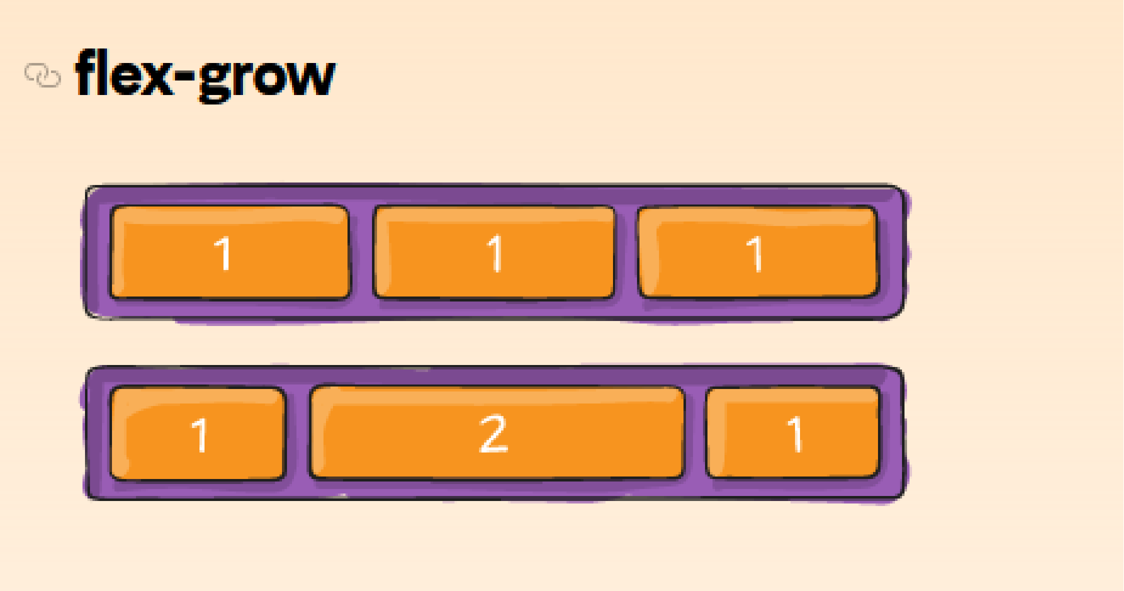
- enkel tussen items, niet tussen items en rand!



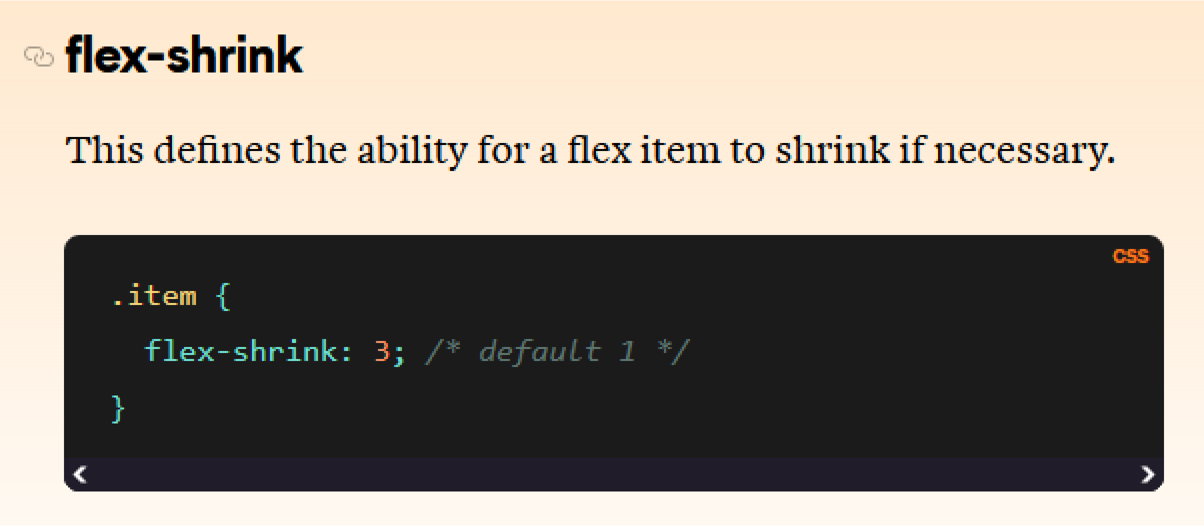
# STYLING – FLEXBOX – CHILDREN



# STYLING – FLEXBOX – CHILDREN



# STYLING – FLEXBOX – CHILDREN

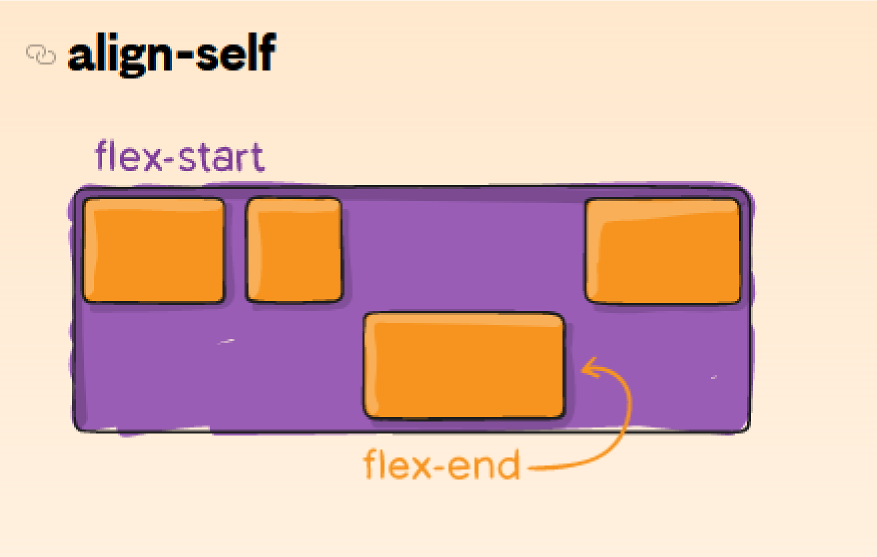


## flex-shrink

This defines the ability for a flex item to shrink if necessary.

```css
.item {
  flex-shrink: 3; /* default 1 */
}
```

# STYLING – FLEXBOX - CHILDREN


align-self
flex-start
flex-end