

Frontend leerlijn

HTML & CSS

Les 3

Planning

14:00 - 15:15

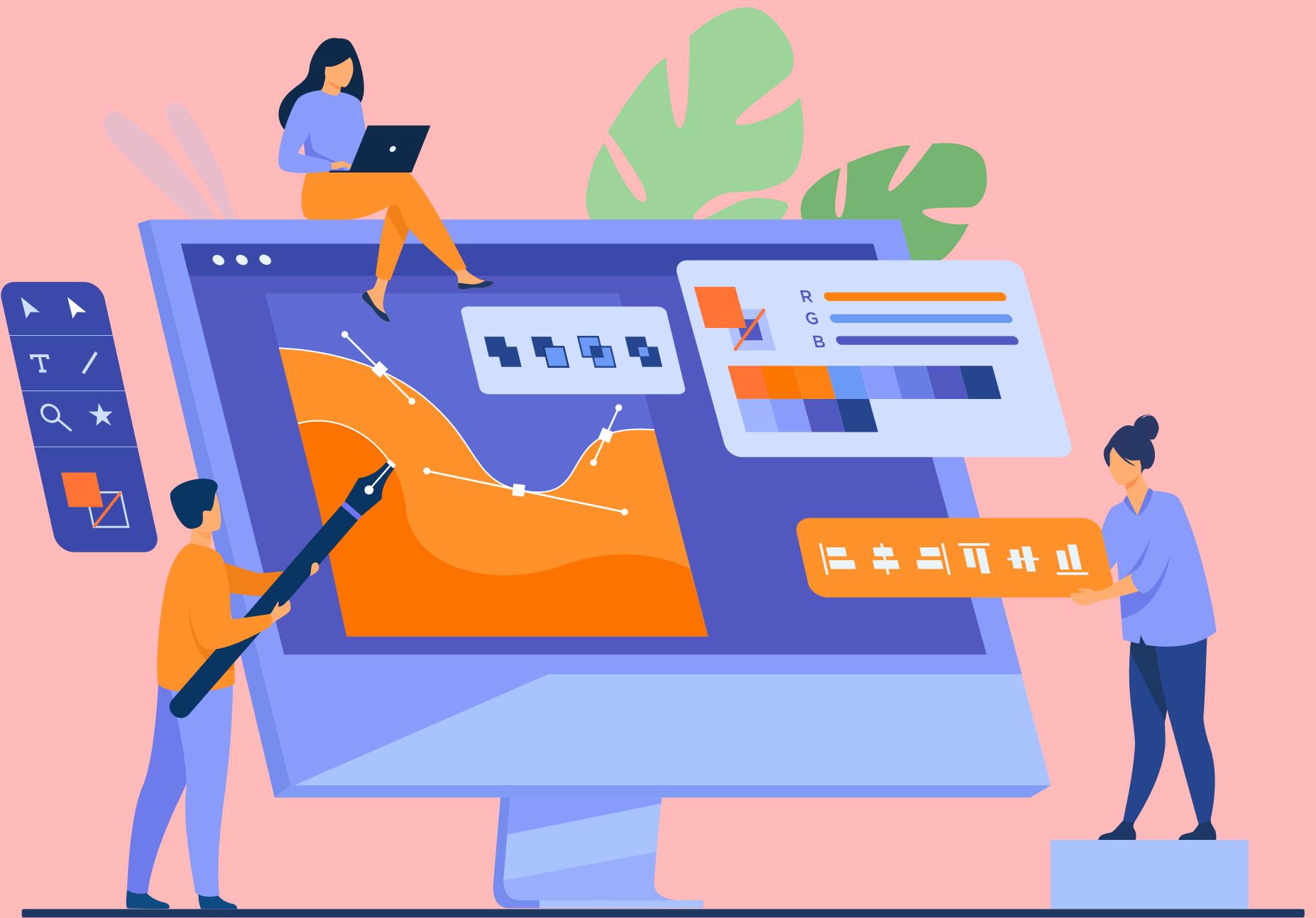
- Uitleg opdracht
- Van design naar webpagina: hoe pak je dat aan?
- Layout strategie
- Positionering en schuine hoeken

15:30 - 17:00

- *Lesopdracht*: conceptopdracht

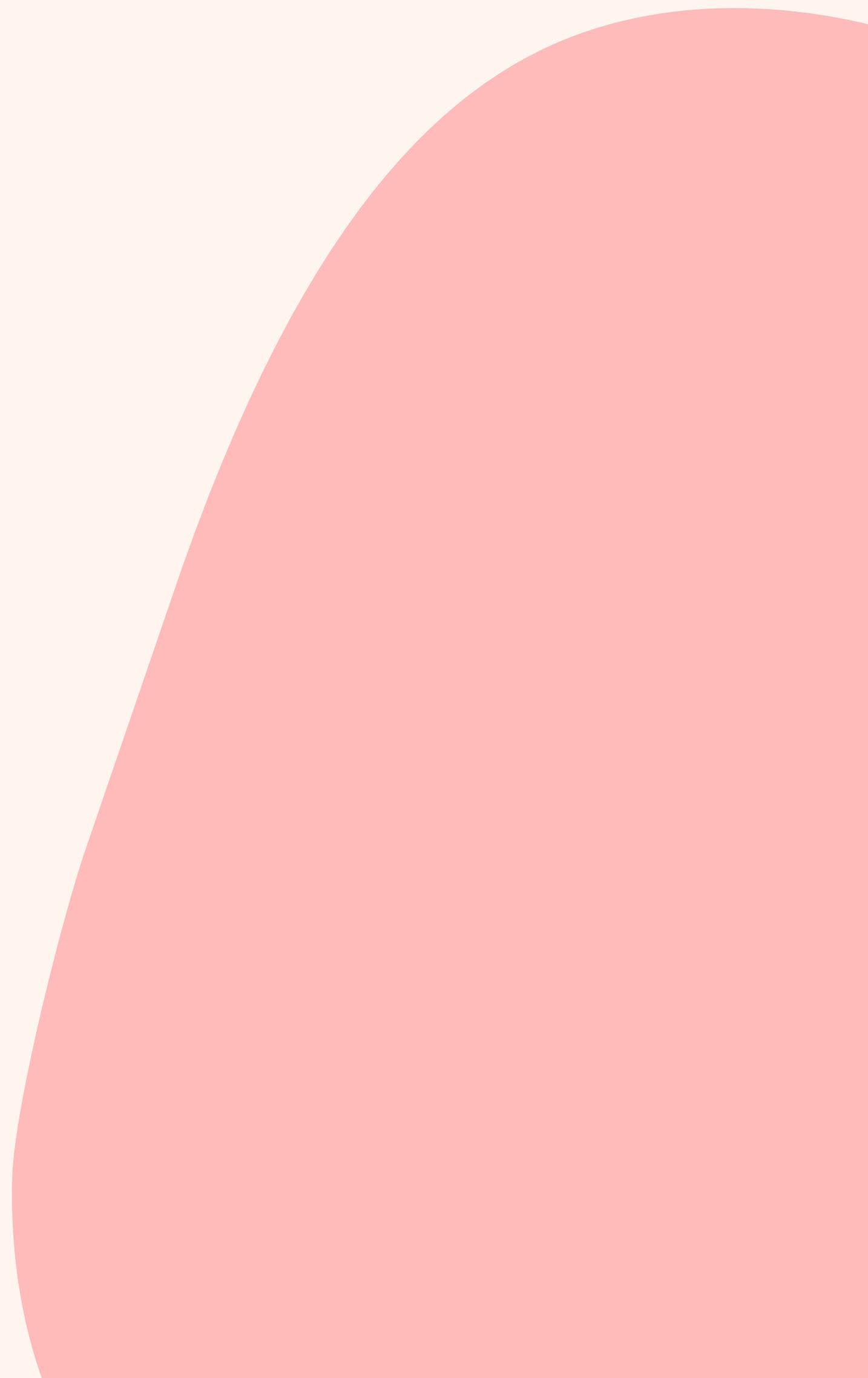
Aanpak

Van design naar webpagina



Realistische opdracht

Je krijgt een design. En dan?



Realistische opdracht

Je krijgt een design. En dan?

- In de praktijk krijg je in 90% van de gevallen een ontwerp van de designer

Realistische opdracht

Je krijgt een design. En dan?

- In de praktijk krijg je in 90% van de gevallen een ontwerp van de designer
- Het is aan jou om dat ontwerp zo gedetailleerd mogelijk om te zetten naar een responsieve website

Realistische opdracht

Je krijgt een design. En dan?

- In de praktijk krijg je in 90% van de gevallen een ontwerp van de designer
- Het is aan jou om dat ontwerp zo gedetailleerd mogelijk om te zetten naar een responsieve website
- **Dat vergt inzicht.** Je kunt niet zomaar lukraak beginnen, dus je analyseert eerst het ontwerp en bedenkt een plan!

Your website is an investment in the success of your business.

We are here to help you succeed.

LET'S WORK TOGETHER



Rekupe is a digital agency in Los Angeles.

We use strategic UX, SEO optimization and beautiful visual designs to help your business grow.

We see every project as a fresh opportunity, pushing ourselves to create new digital experiences.

Your website is an investment in the success of your business.

We are here to help you succeed.

LET'S WORK TOGETHER

Demo!

Rekupe is a digital agency in Los Angeles.

We use strategic UX, SEO optimization and beautiful visual designs to help your business grow.

We see every project as a fresh opportunity, pushing ourselves to create new digital experiences.

20 min

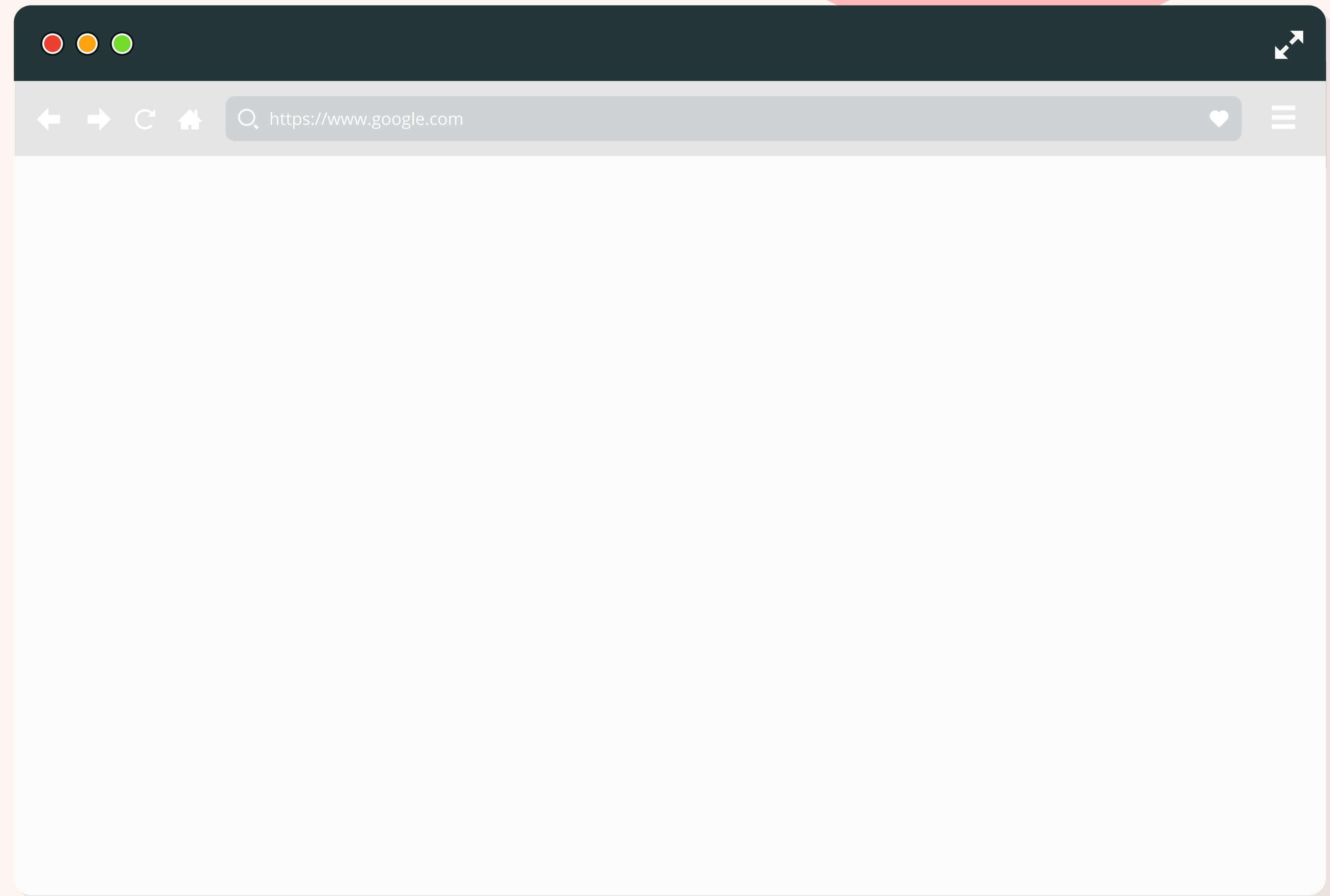
Opdracht - Concept opdracht plan

Bekijk het schermontwerp goed en beantwoord de volgende vragen samen:

- Welke groepen elementen herken je? Teken ze op het design!
- Welke semantische elementen zou je gebruiken? Denk aan:
` <div> <section> <article> <header> <footer> <main> <nav>`
- Hoe deel je de verdeling van heading-titels in? Welke worden een
`<h1> <h2> <h3>`, etc.

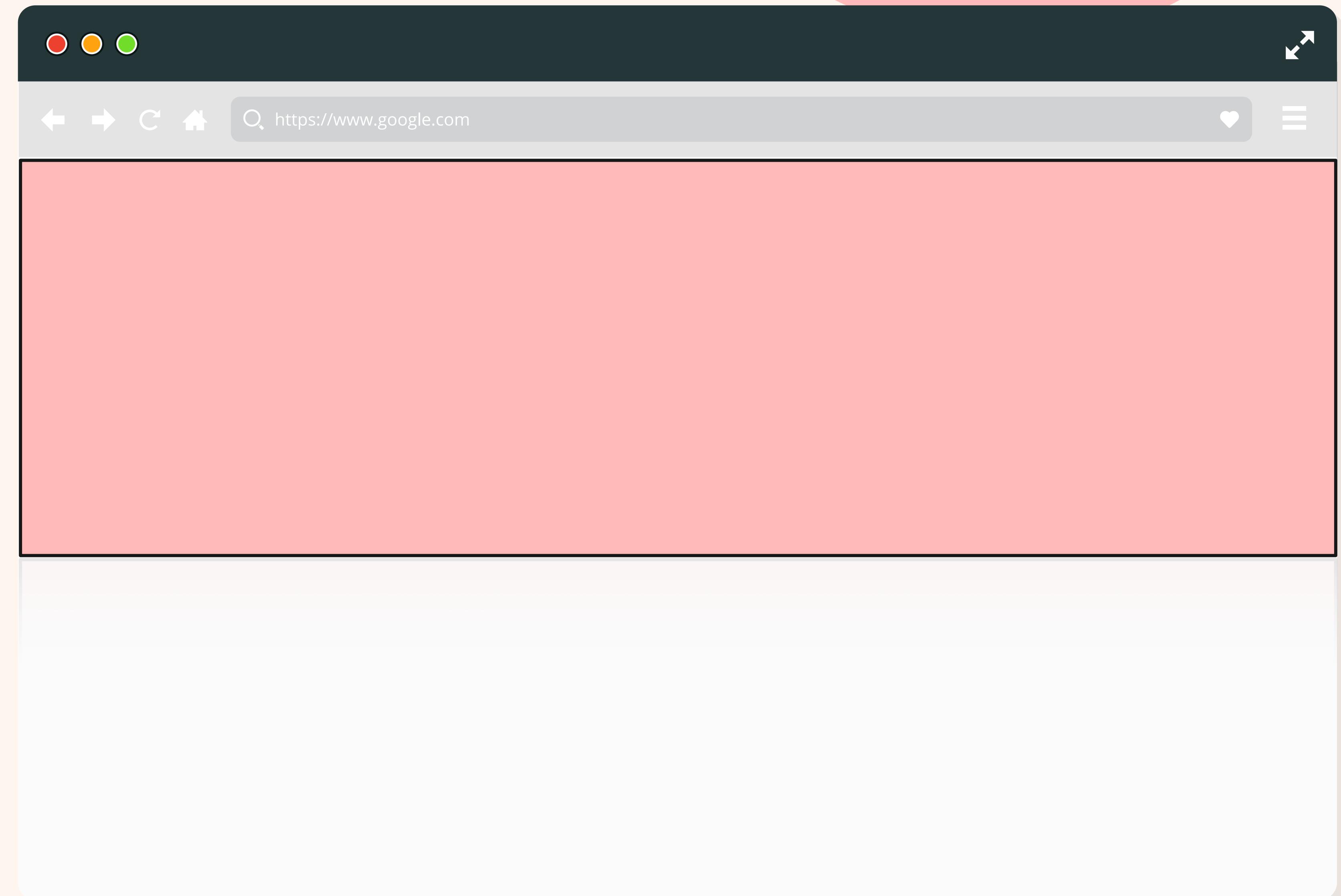
Opbouw

Inner-outer



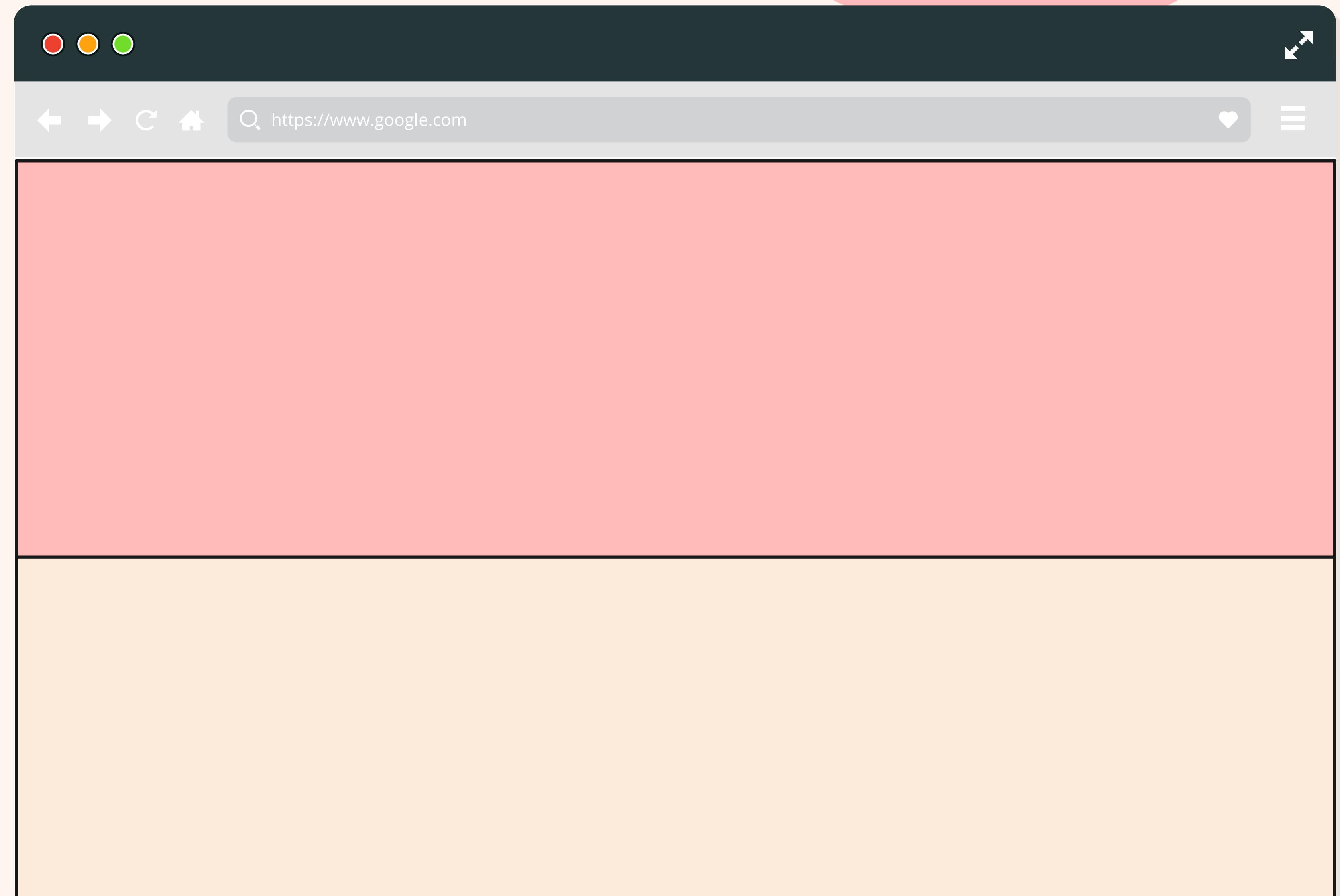
Opbouw

Inner-outer



Opbouw

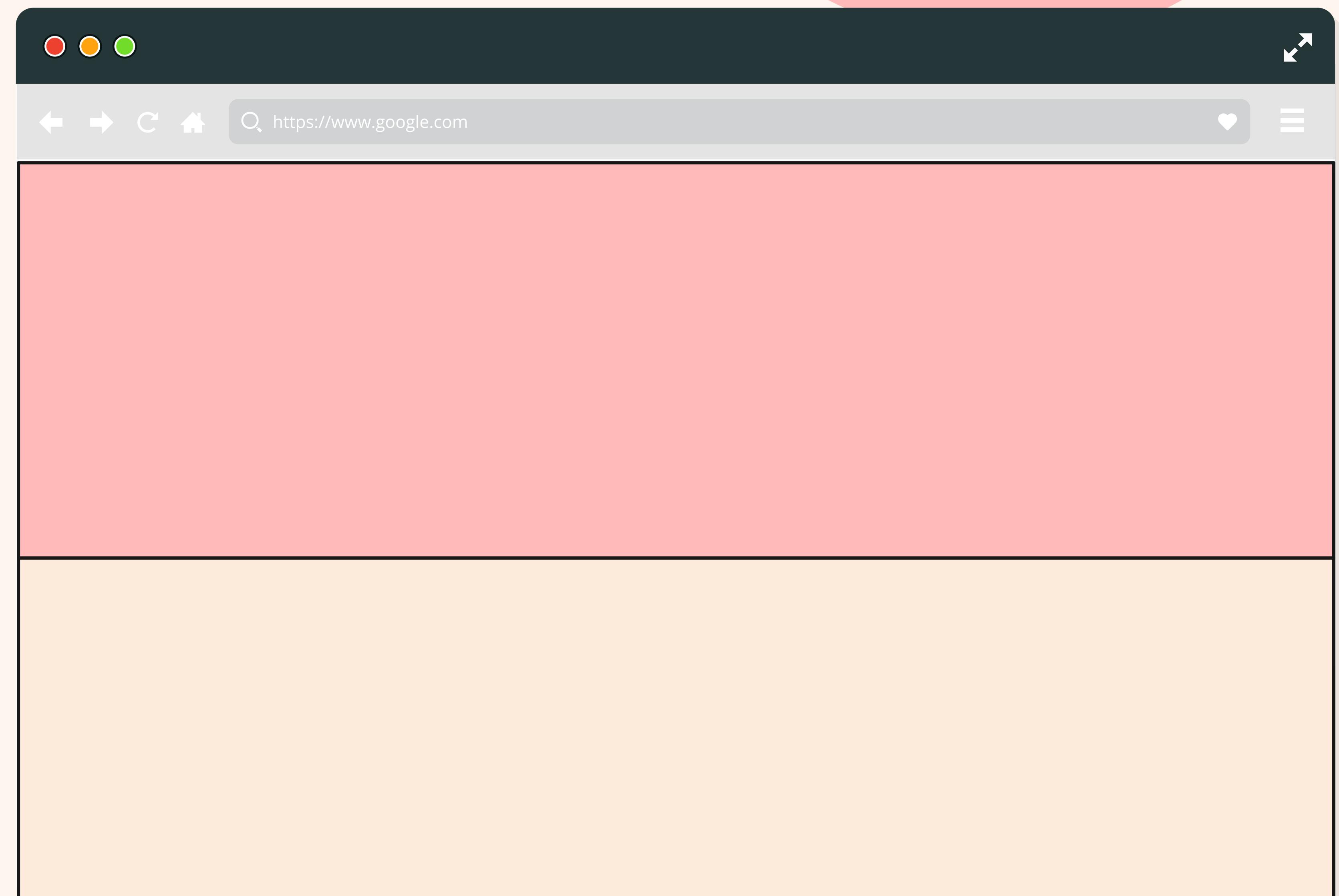
Inner-outer



Opbouw

Inner-outer

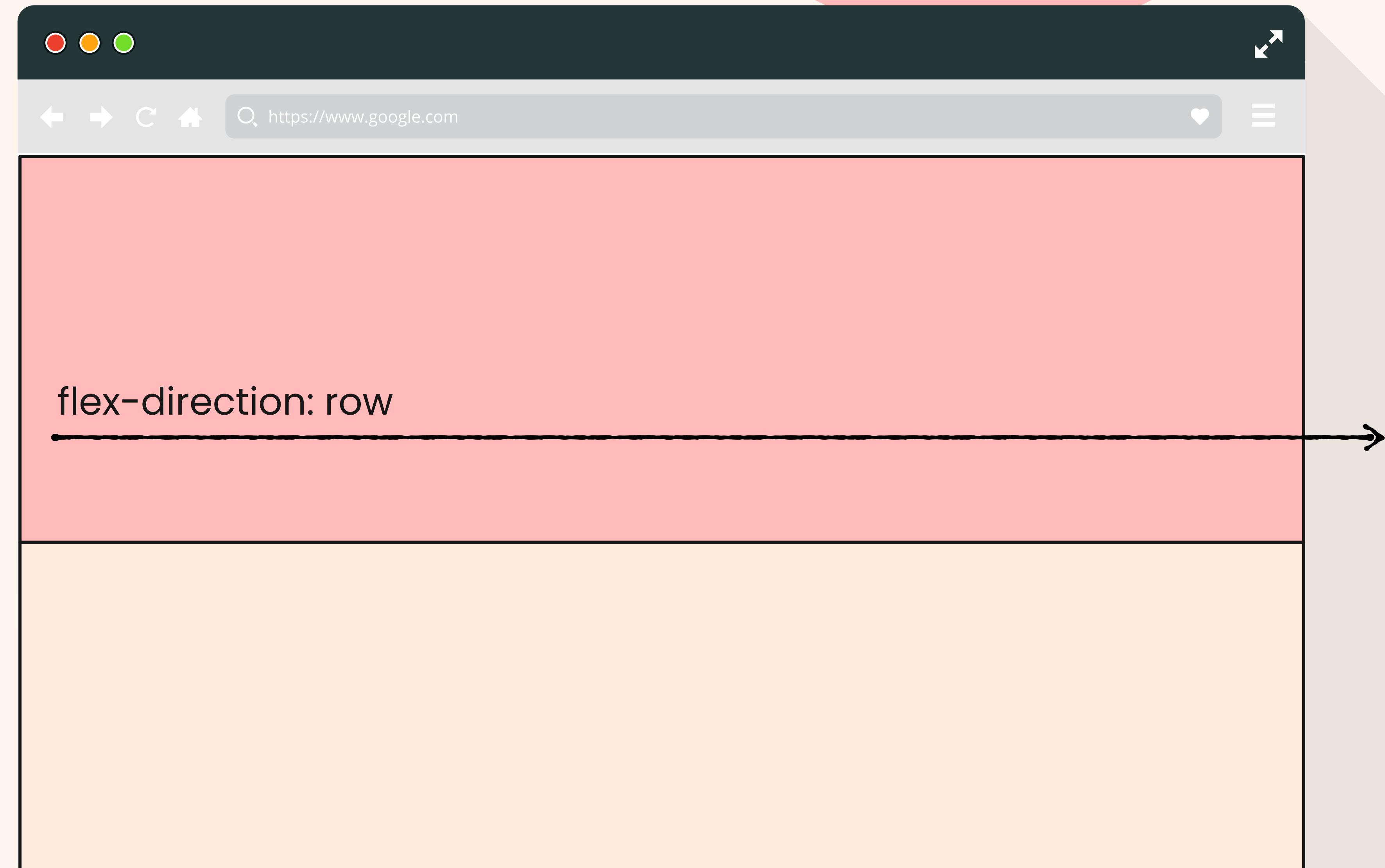
Ieder vlak of area wil je zien als individuele outer-container



Opbouw

Inner-outer

Ieder vlak of area wil je zien als individuele outer-container

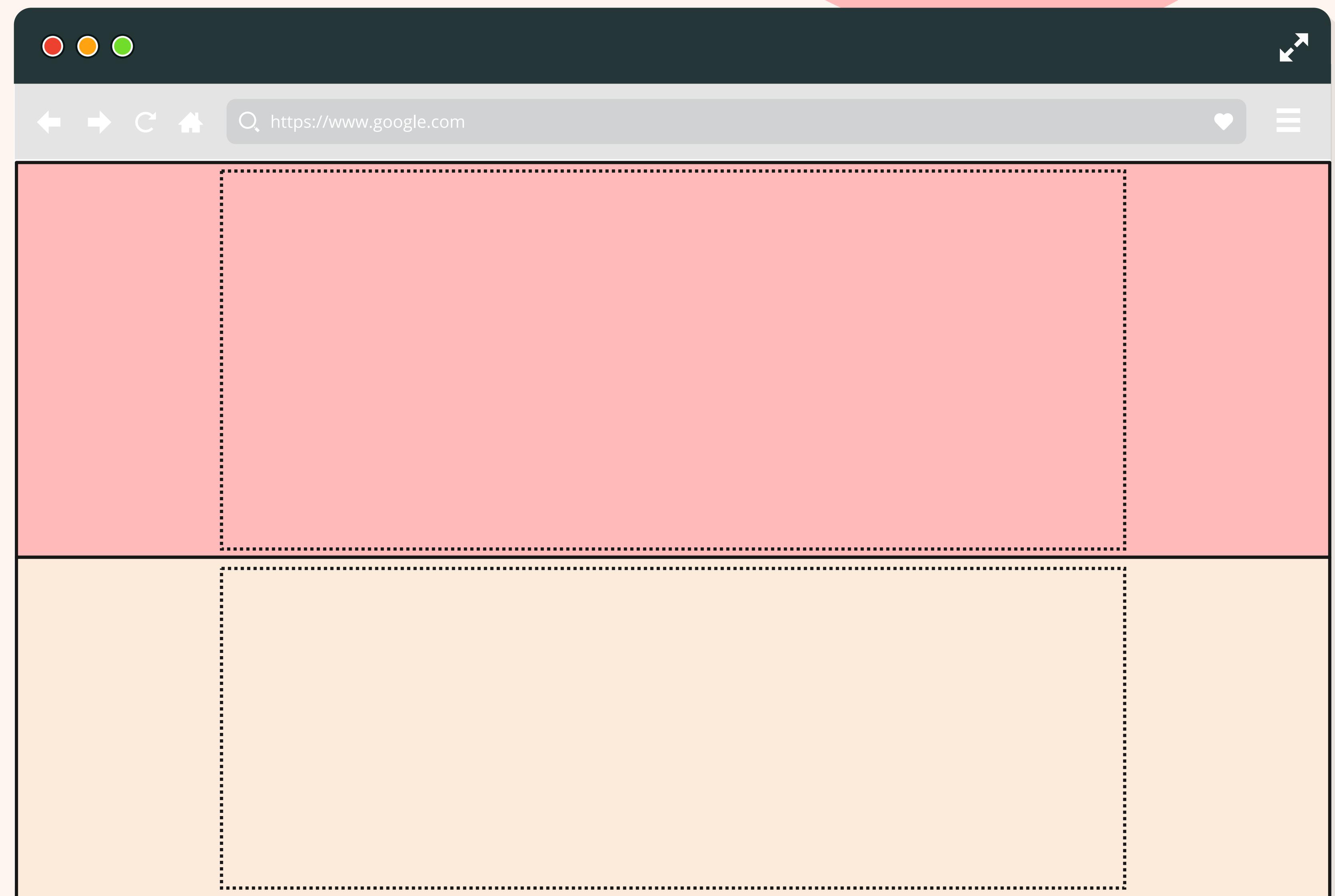


Opbouw

Inner-outer

Ieder vlak of area wil je zien als individuele outer-container

Daarin plaats je telkens een nieuwe groep, de inner-container



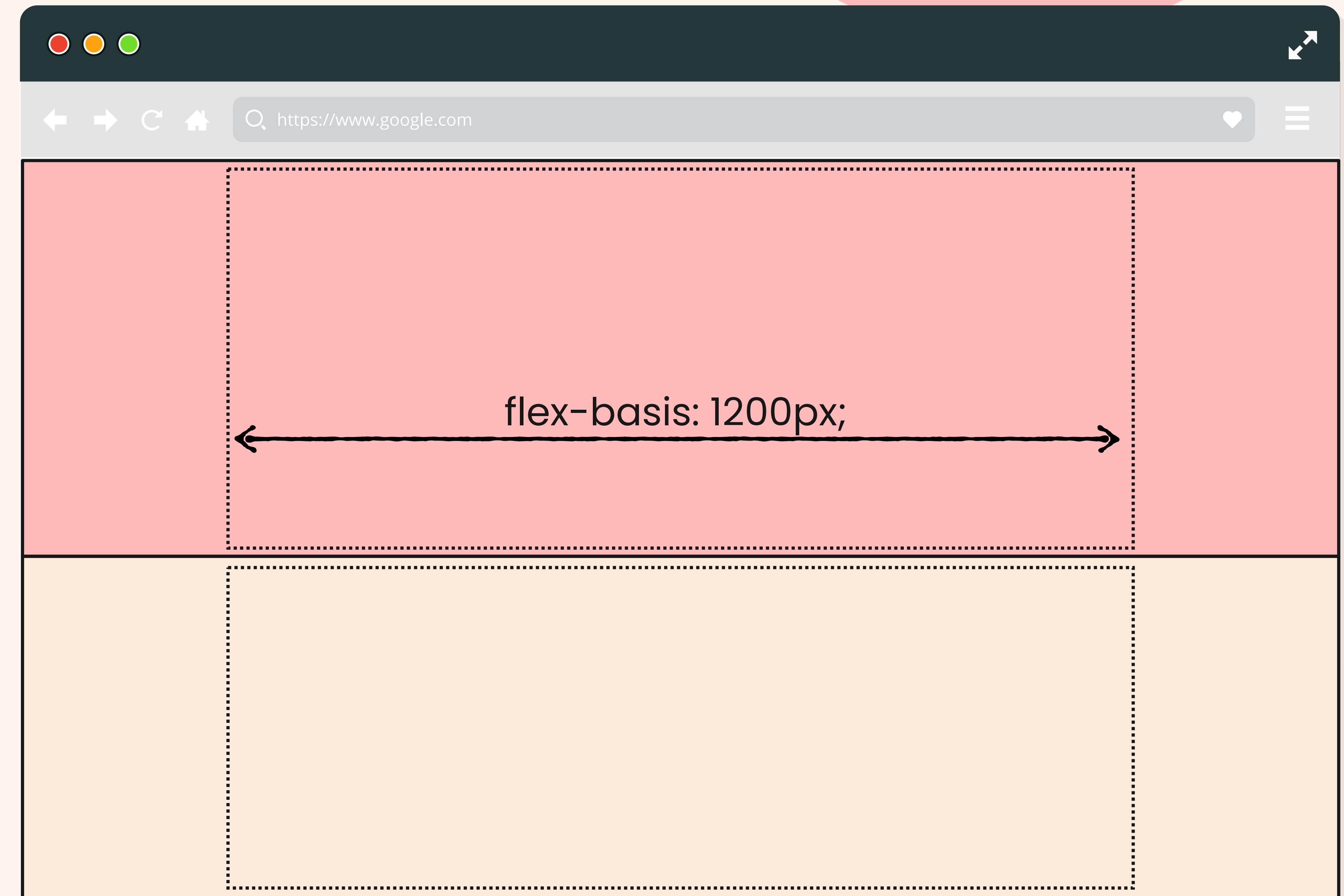
Opbouw

Inner-outer

Ieder vlak of area wil je zien als individuele outer-container

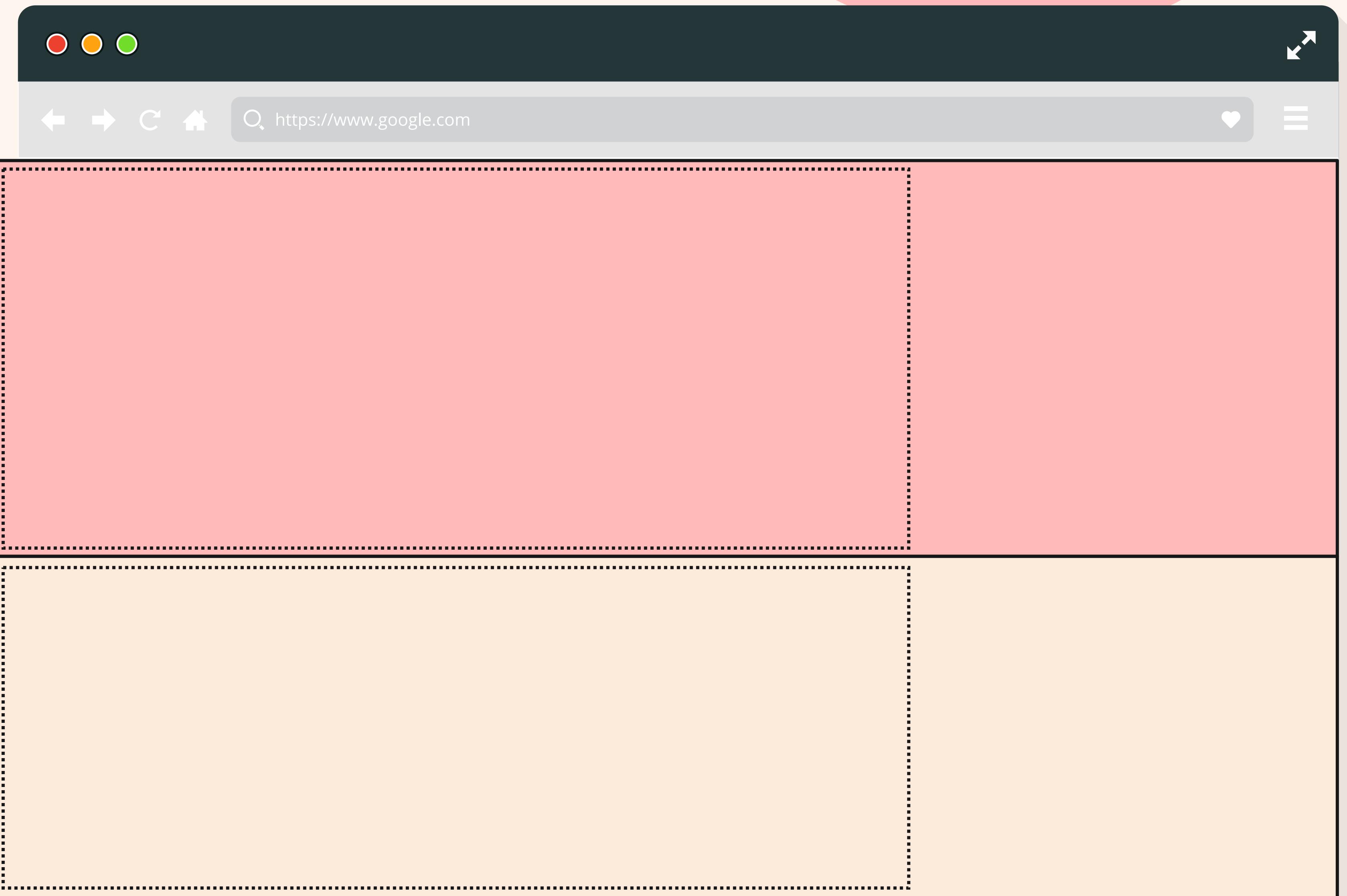
Daarin plaats je telkens een nieuwe groep, de inner-container

Deze container krijgt een maximale breedte met **flex-basis**, zodat de content altijd dezelfde afstand tot de rand heeft!



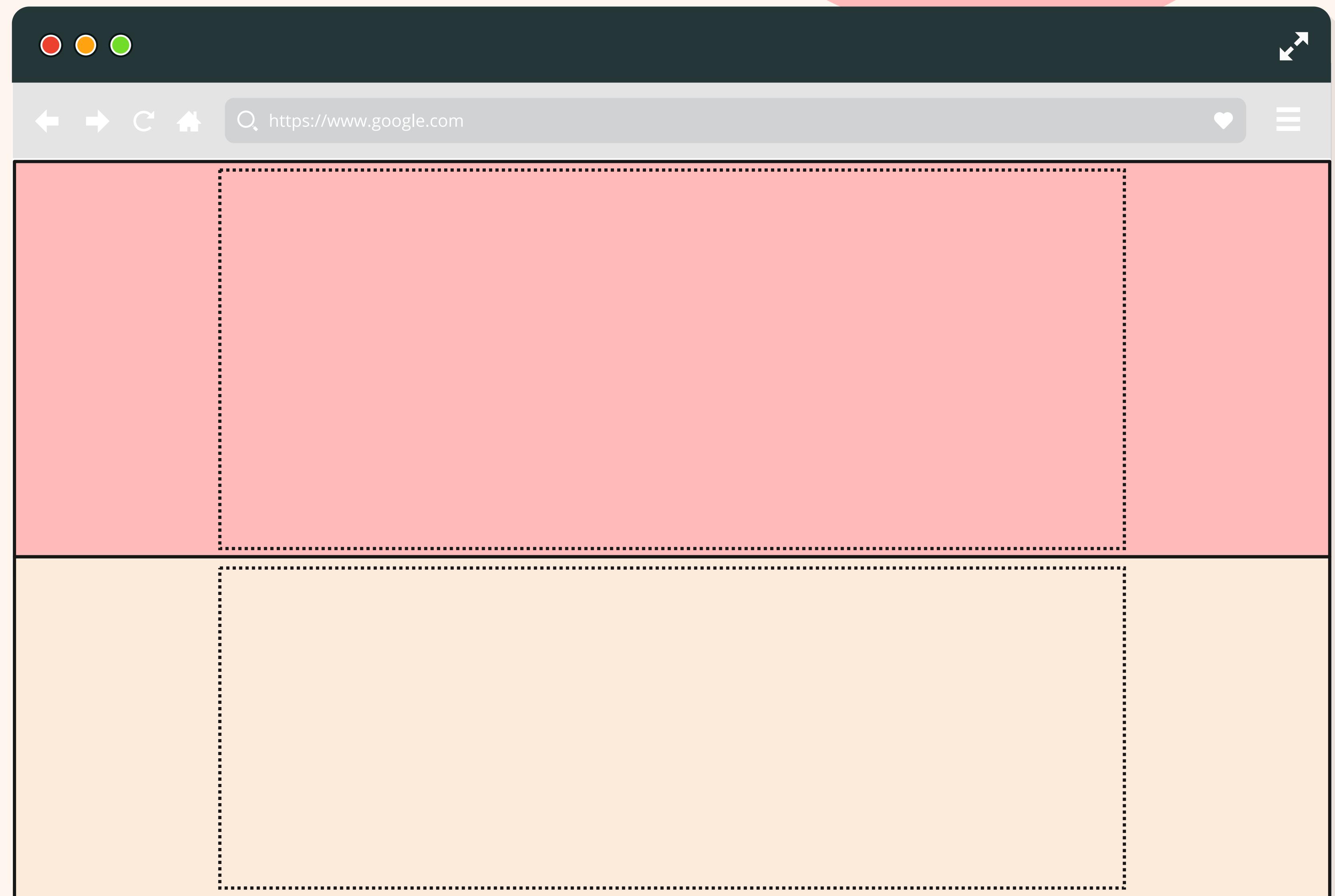
Opbouw

Inner-outer



Opbouw

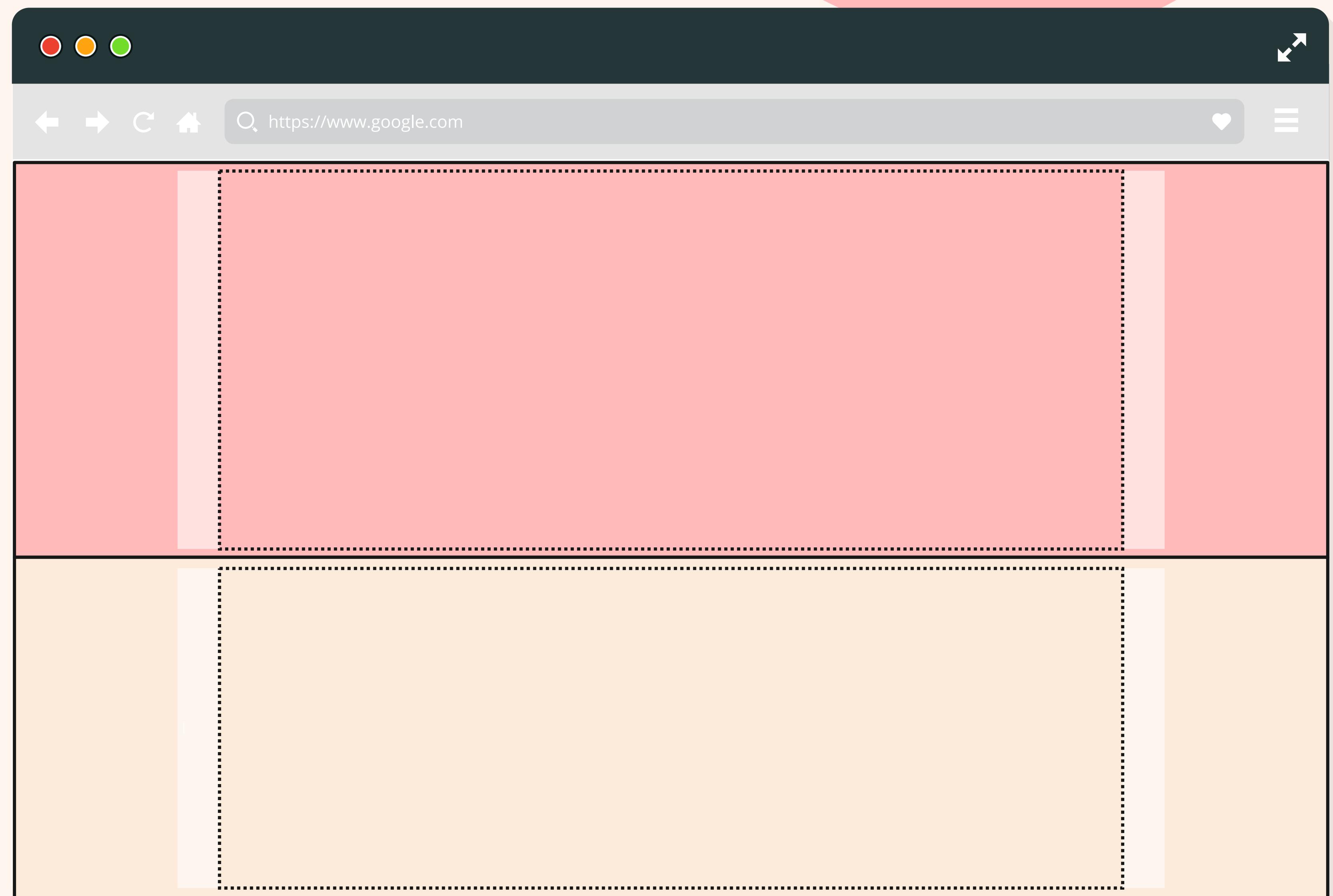
Inner-outer



Opbouw

Inner-outer

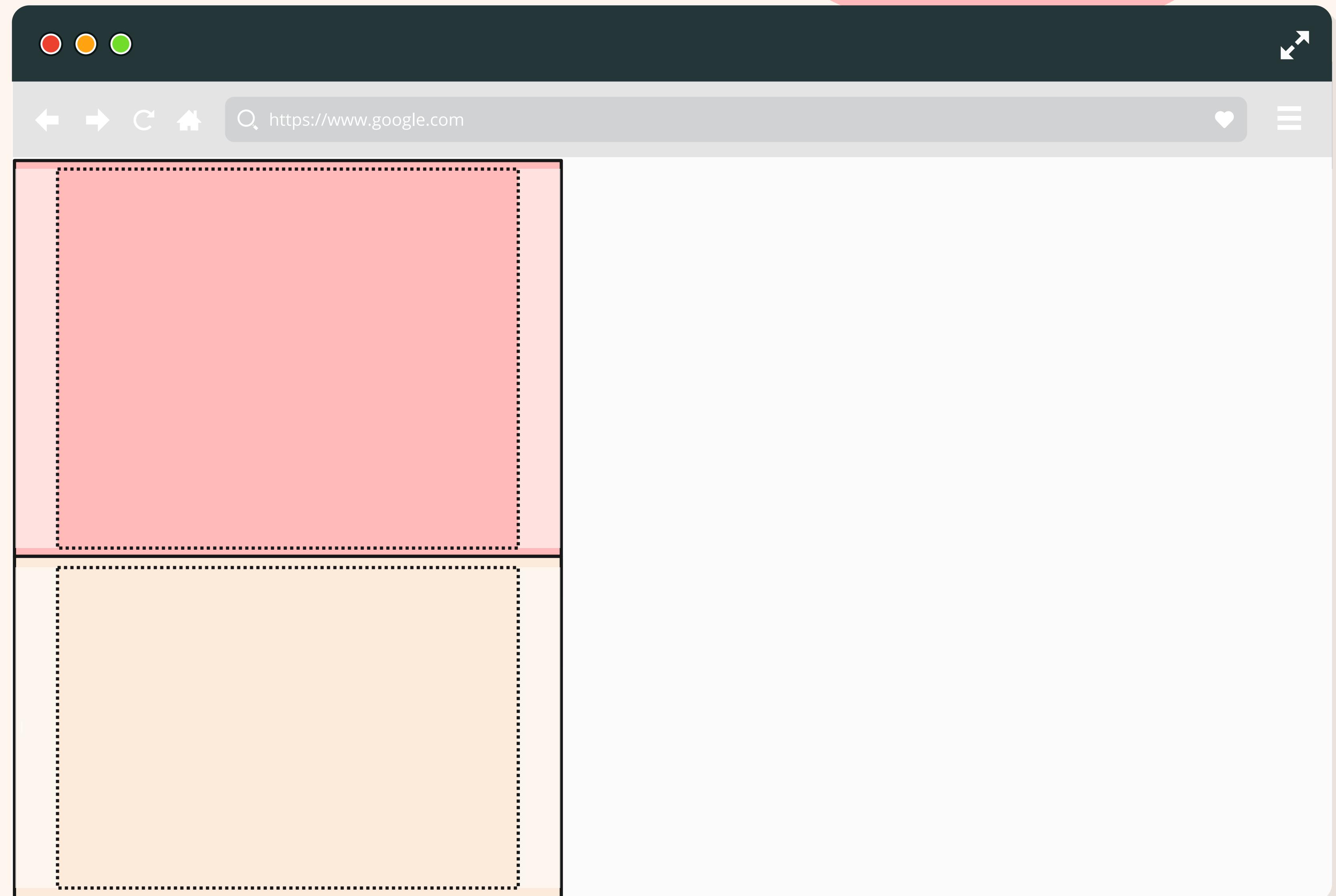
Zorg ook altijd dat je een beetje margin toekoegt aan de zijkanten. Dit zie je op grote schermen niet, maar je wil de content niet helemaal tegen de rand drukken op mobiel



Opbouw

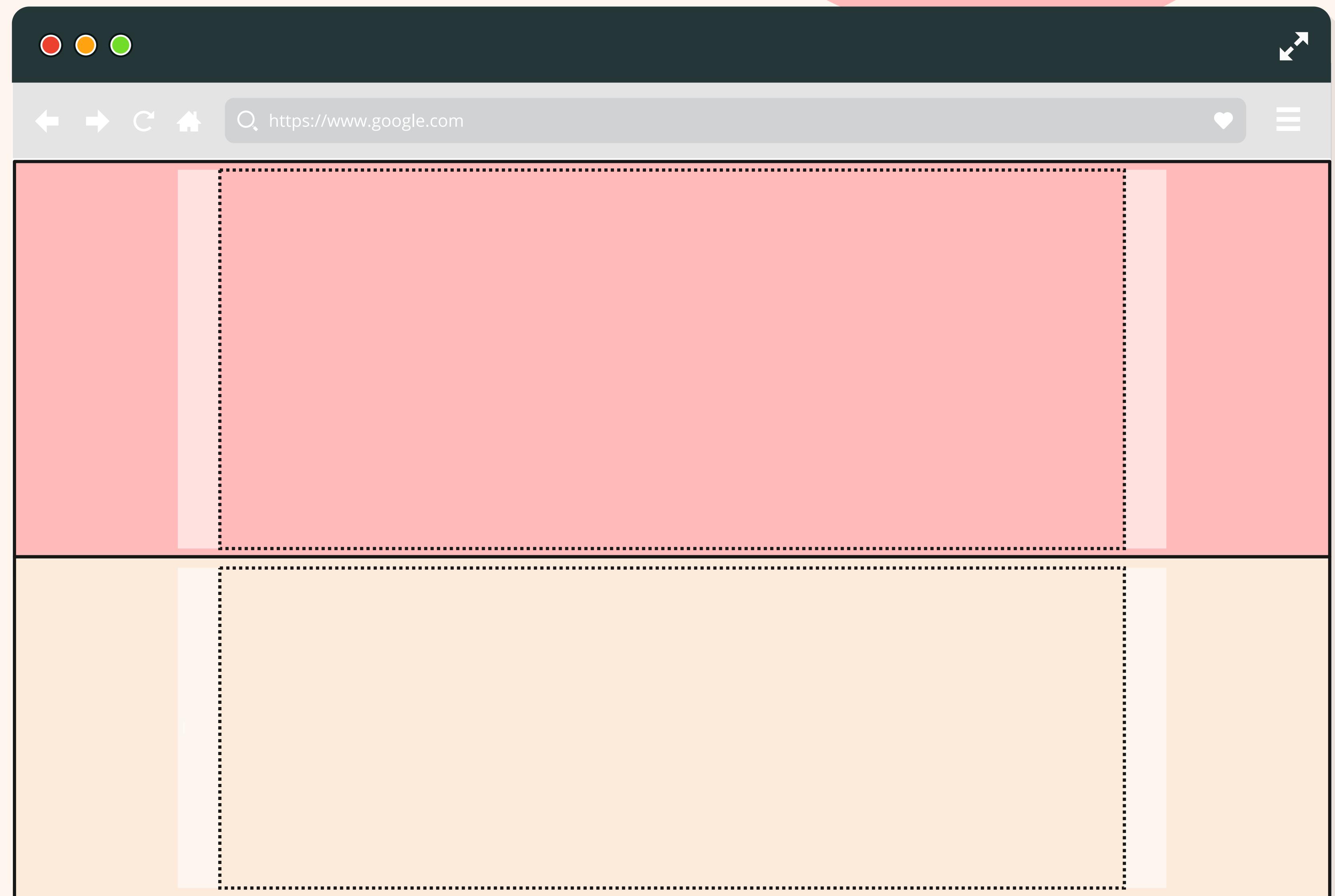
Inner-outer

Zorg ook altijd dat je een beetje margin toekoegt aan de zijkanten. Dit zie je op grote schermen niet, maar je wil de content niet helemaal tegen de rand drukken op mobiel



Opbouw

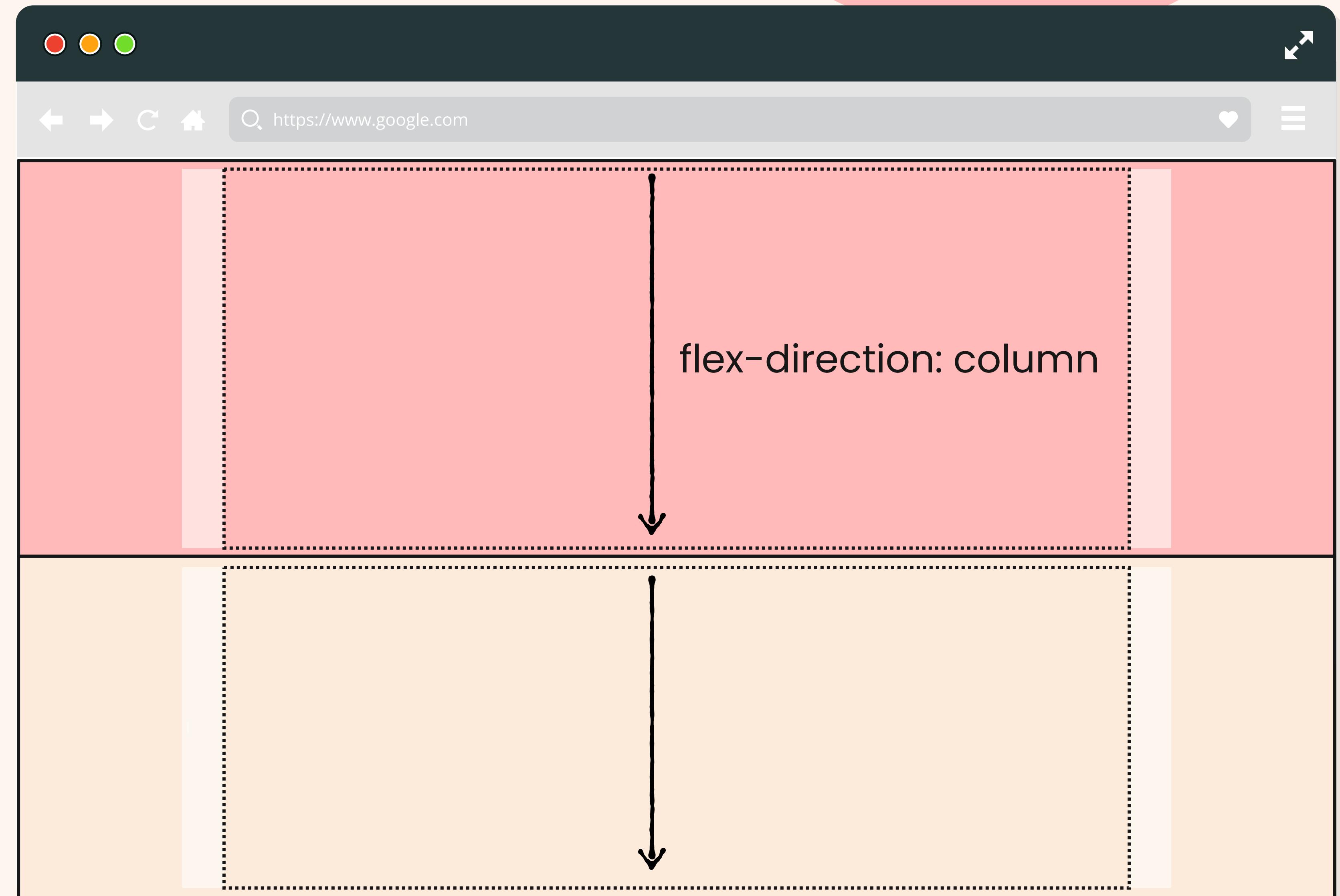
Inner-outer



Opbouw

Inner-outer

Het is logisch om de **inner-container** altijd een column-richting te geven, omdat de content-groepjes zich onder elkaar zullen moeten bevinden.

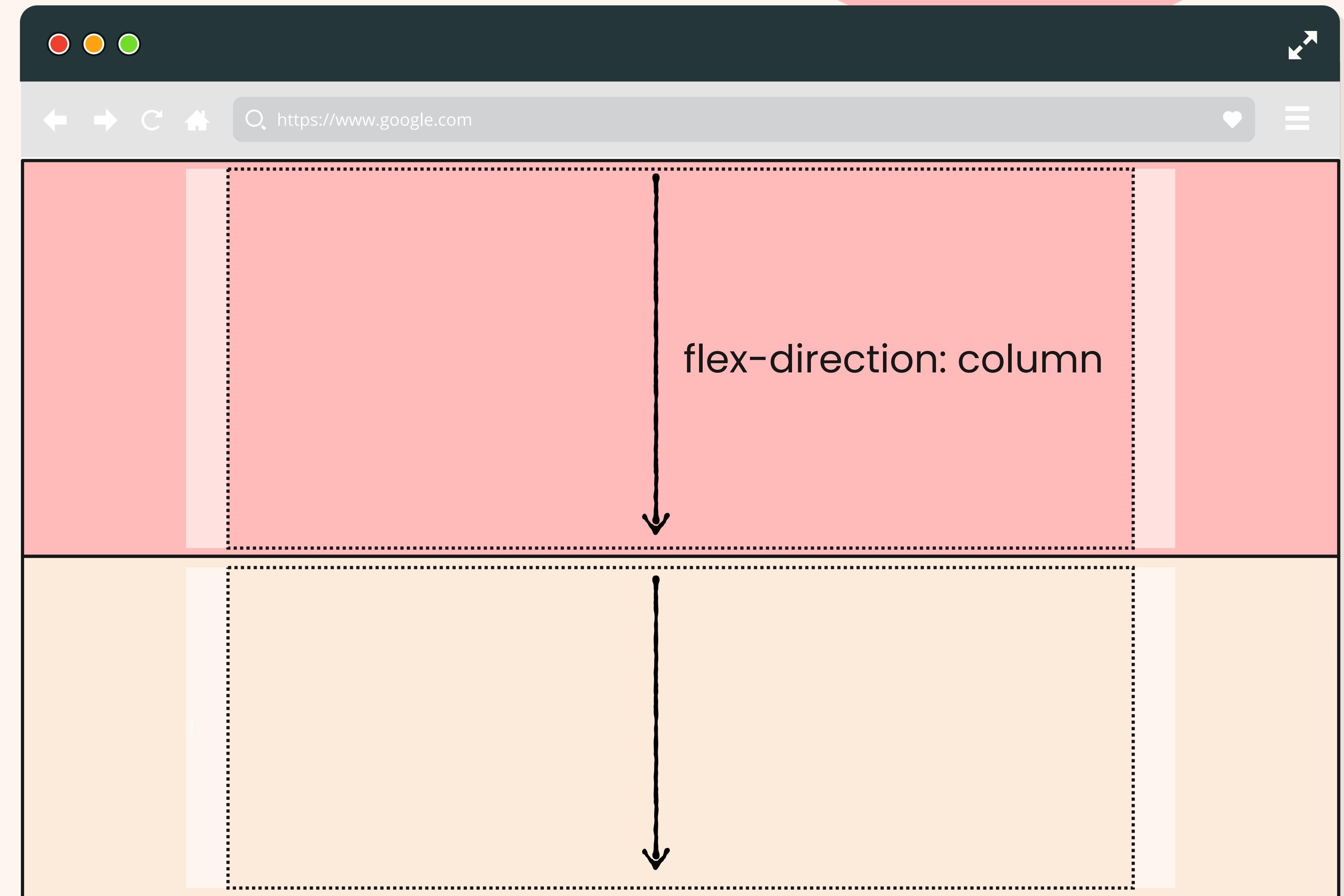


Opbouw

Inner-outer

Het is logisch om de **inner-container** altijd een column-richting te geven, omdat de content-groepjes zich onder elkaar zullen moeten bevinden.

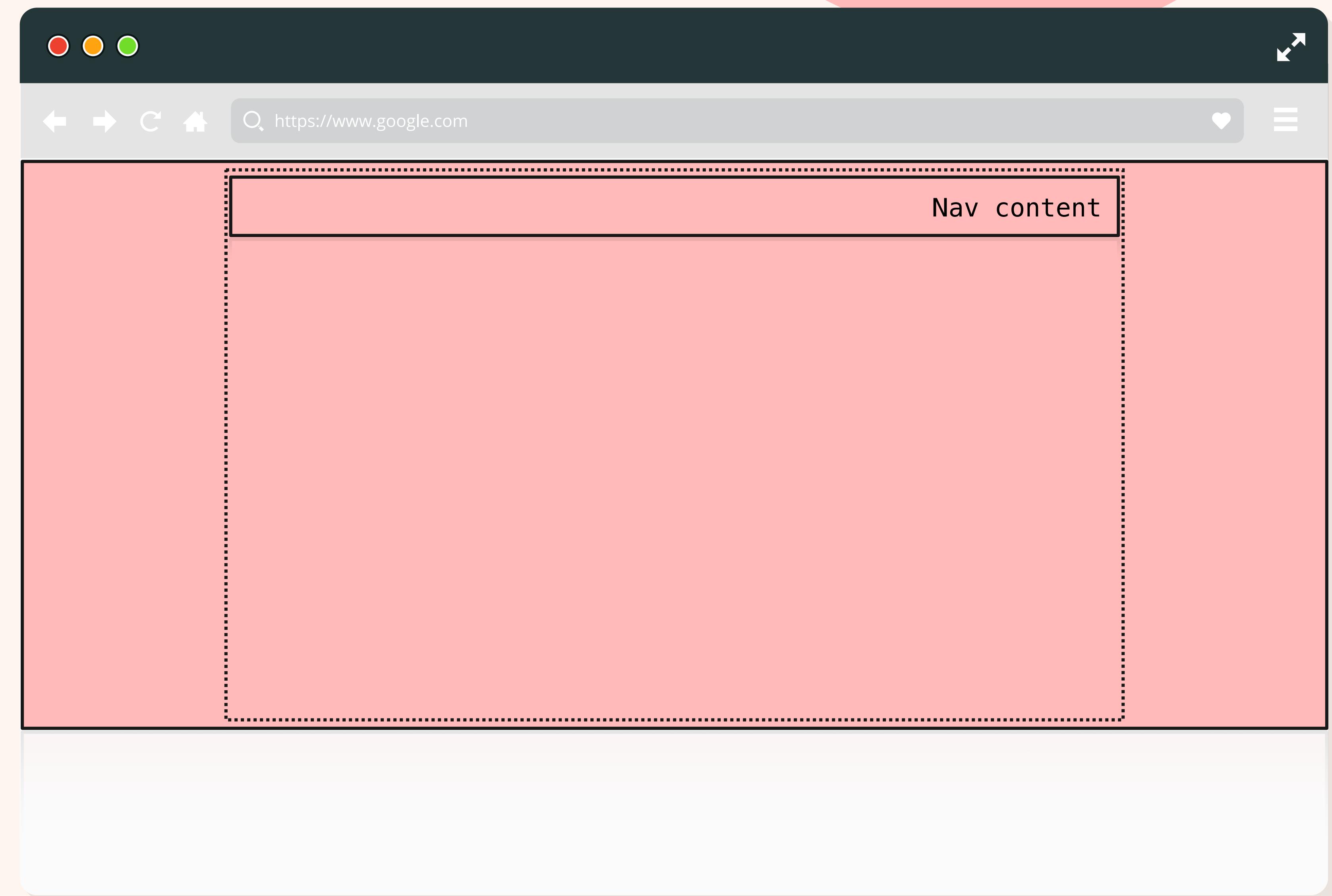
Elementen kunnen dus zowel een flex-item als een flex-container tegelijk zijn!



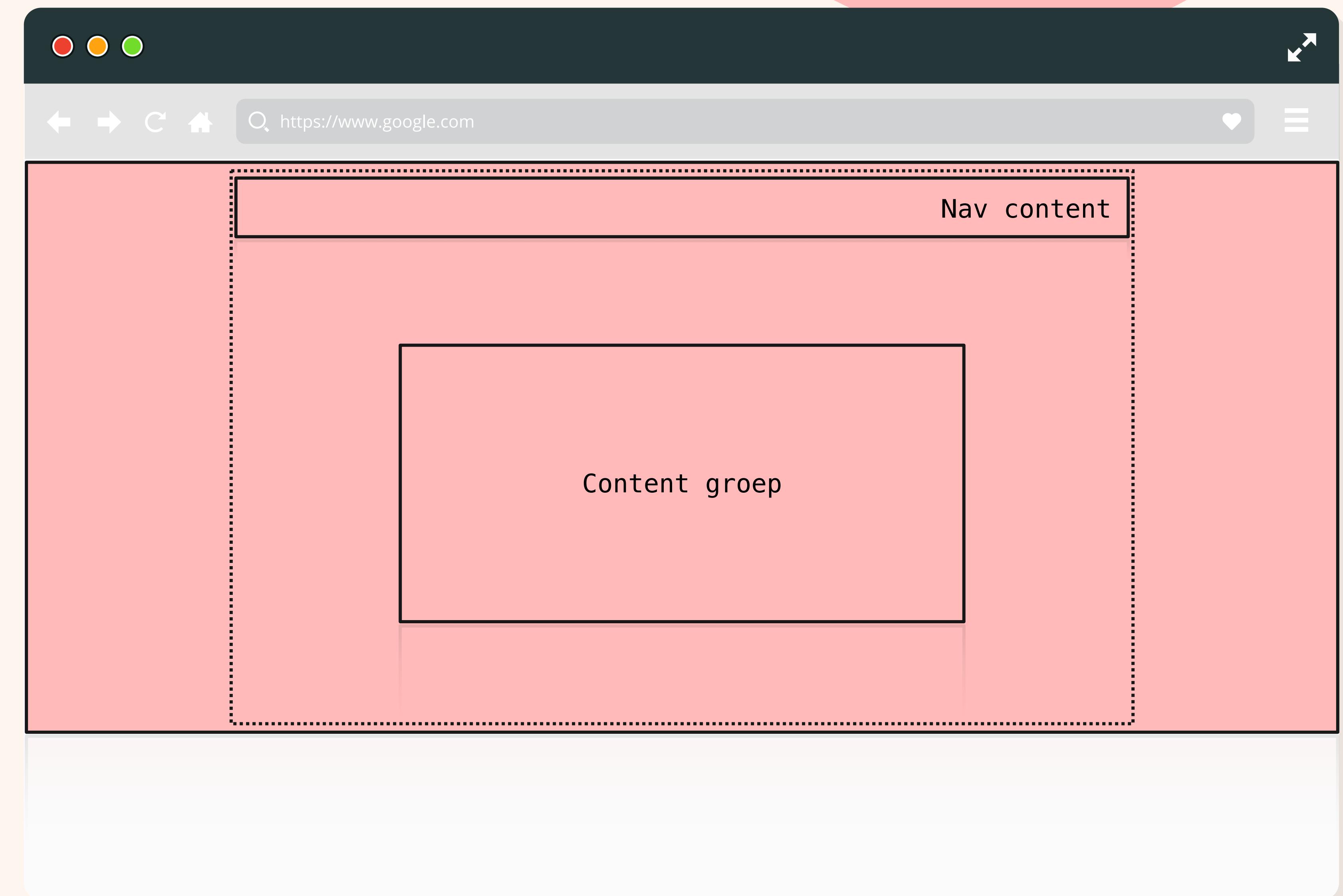
Header



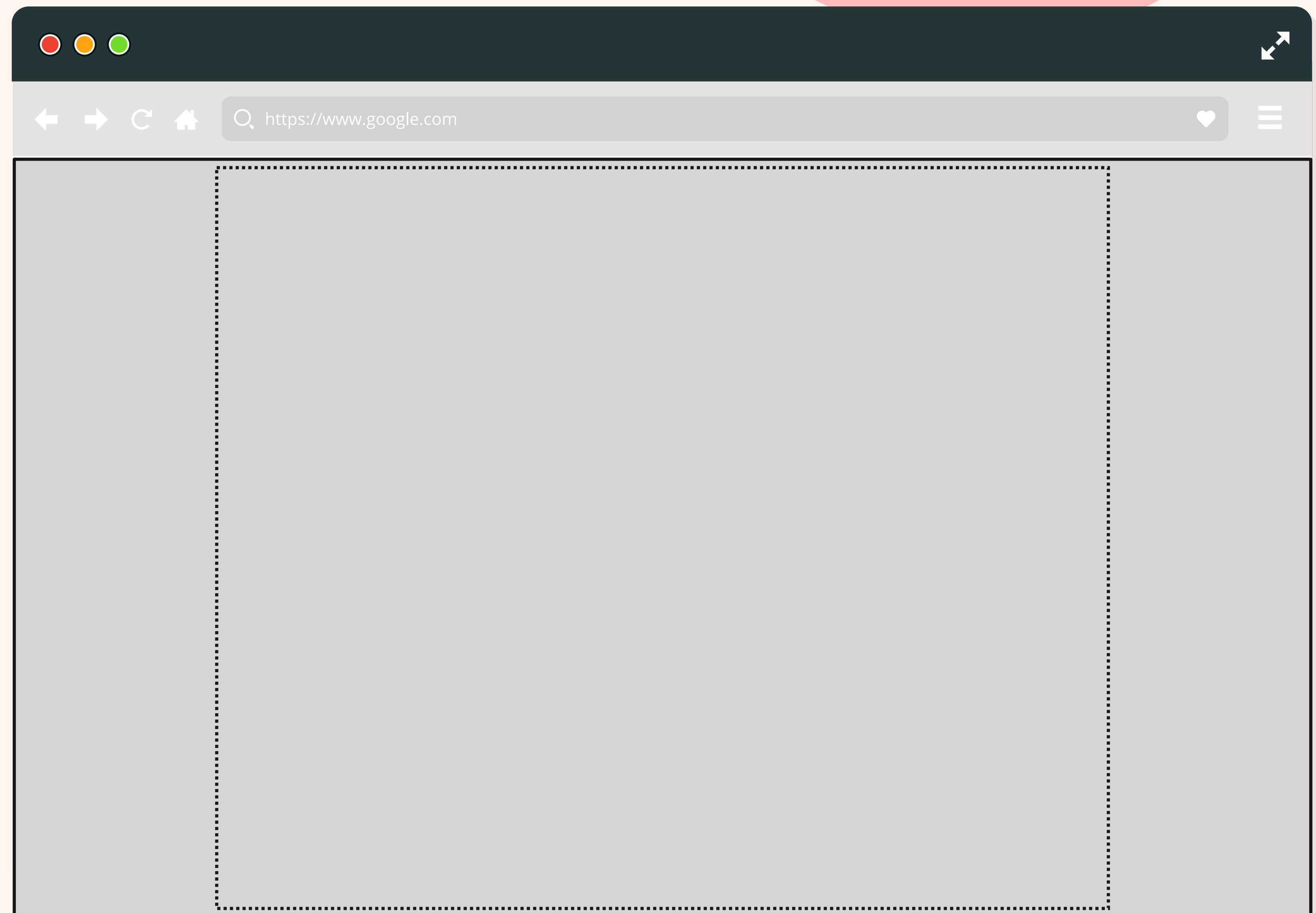
Header



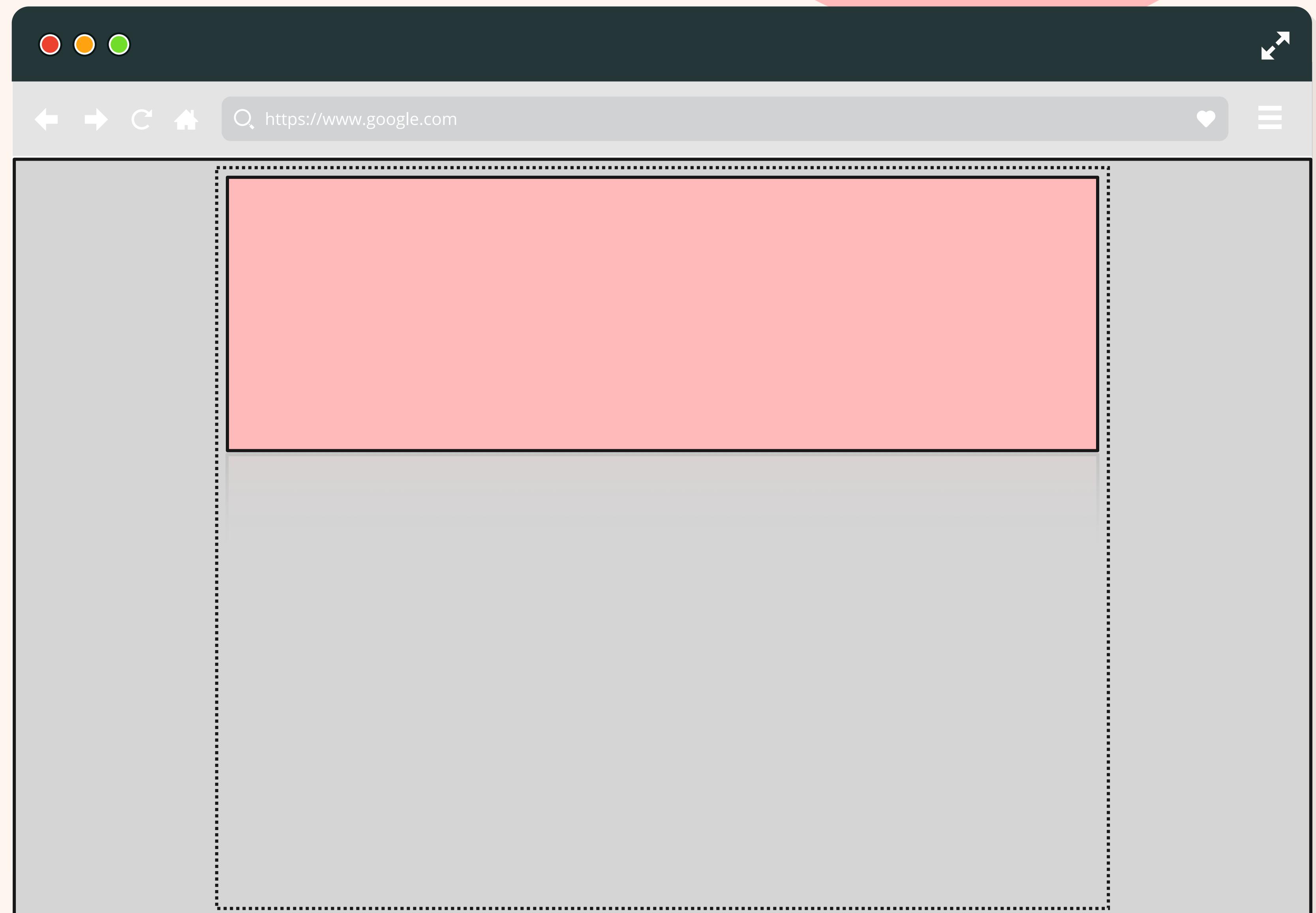
Header



Work



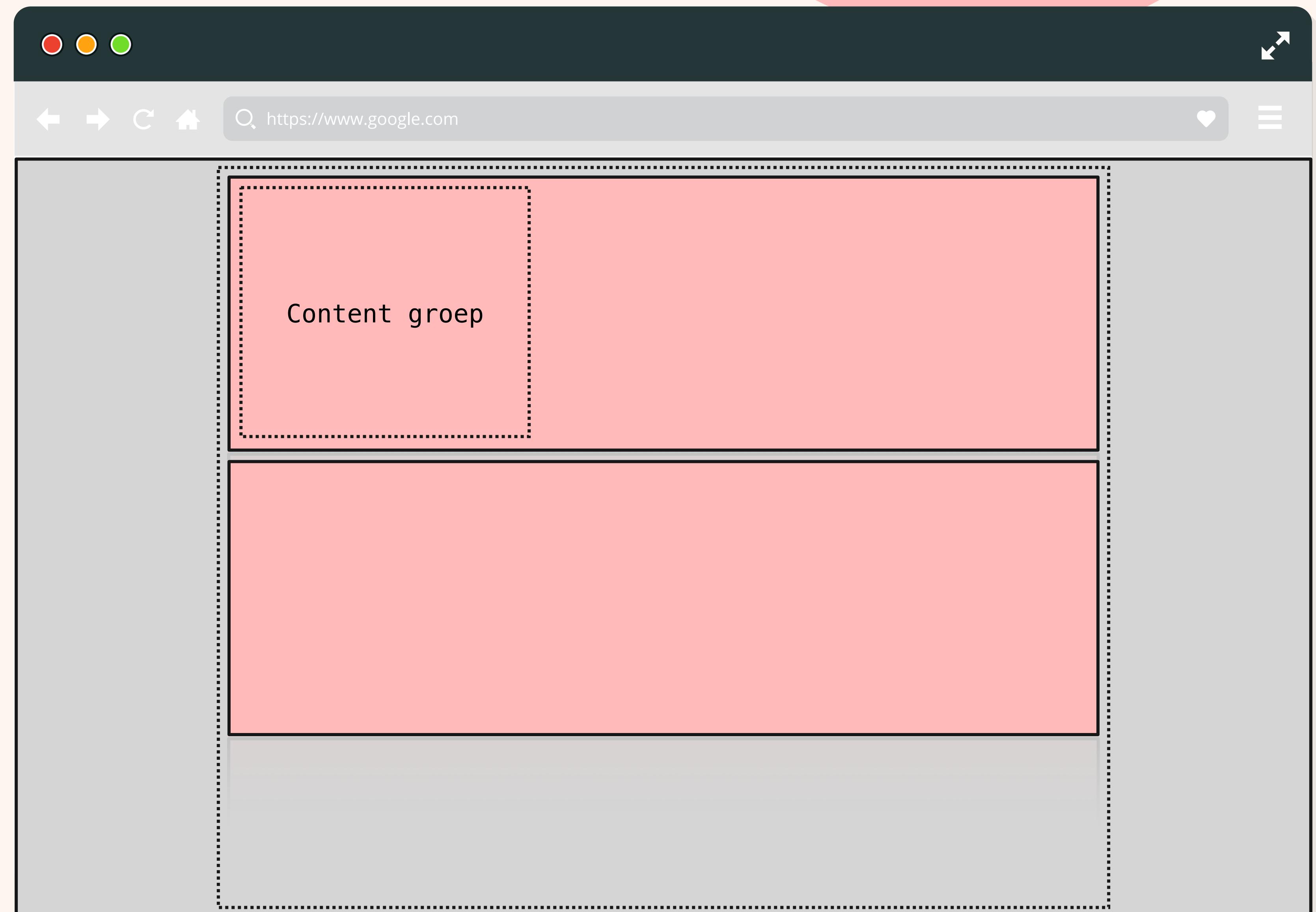
Work



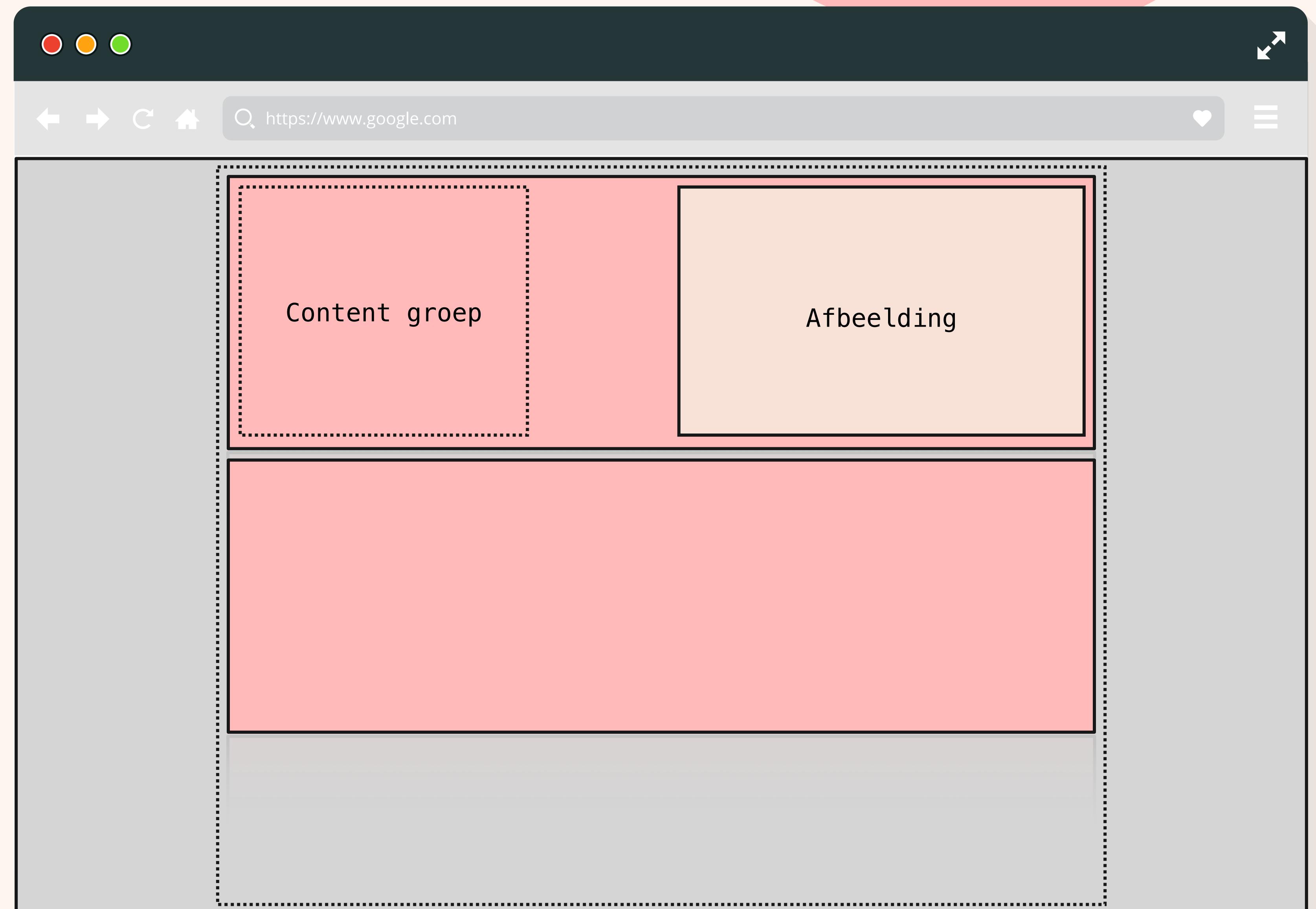
Work



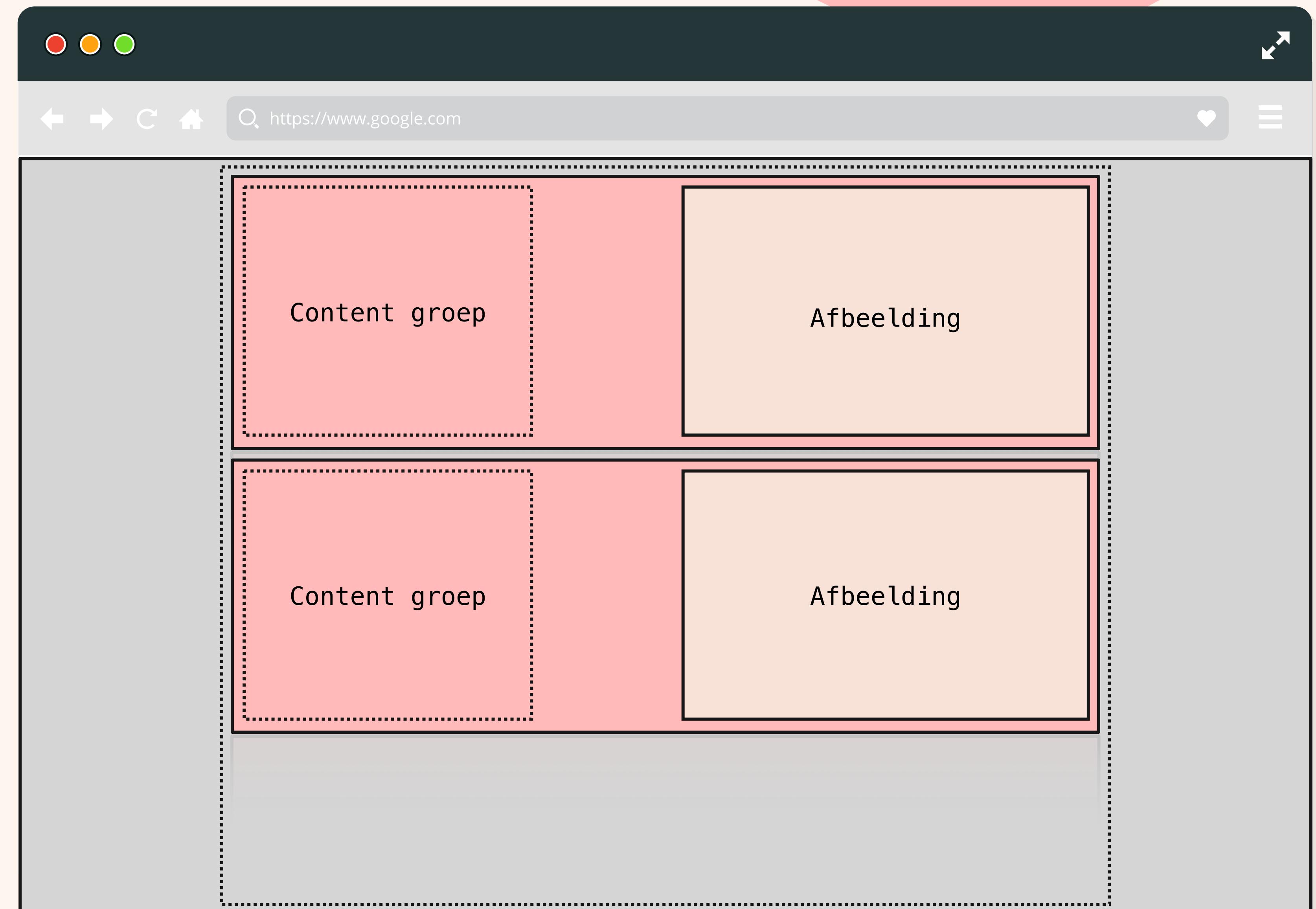
Work



Work



Work



Positionering

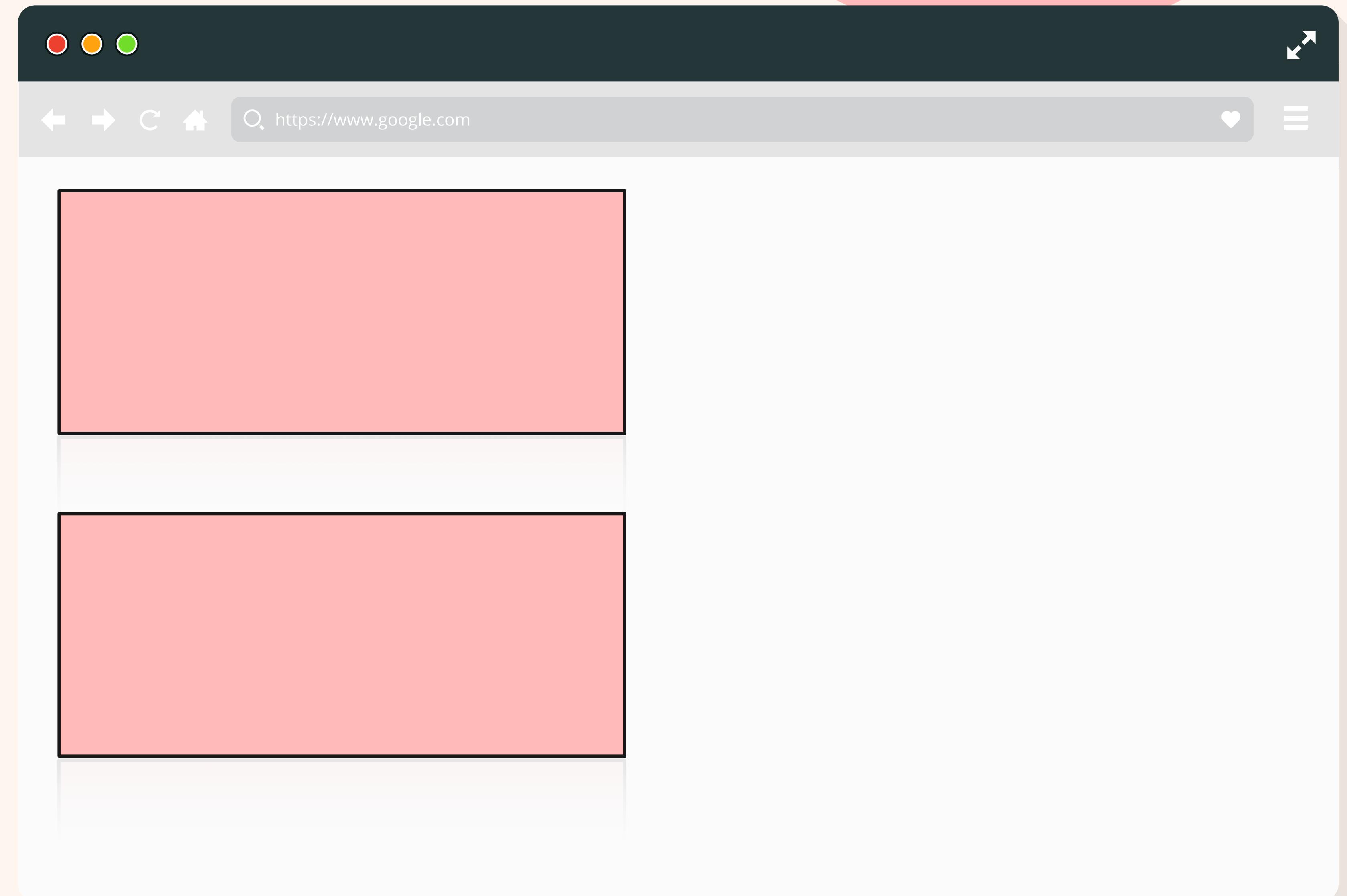
Elementen uit hun normale flow van de pagina halen

- Wanneer je elementen absoluut gaat positioneren, is het belangrijk om te weten *ten opzichte van welk parent element* je dat doet
- Google Dev Tools are your best friend!

Position

Relative

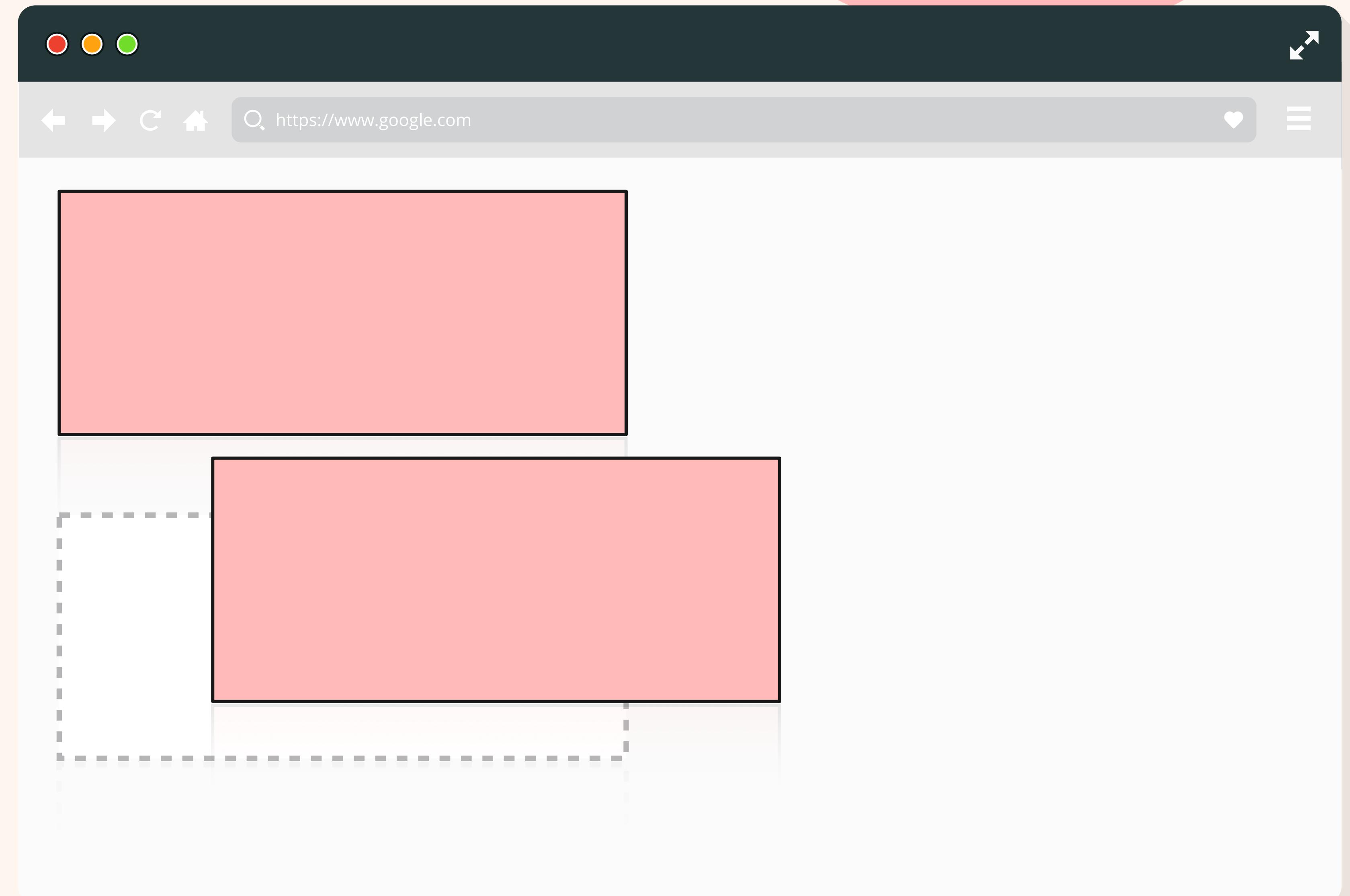
Elementen eromheen gedragen zich alsof het element er nog is



Position

Relative

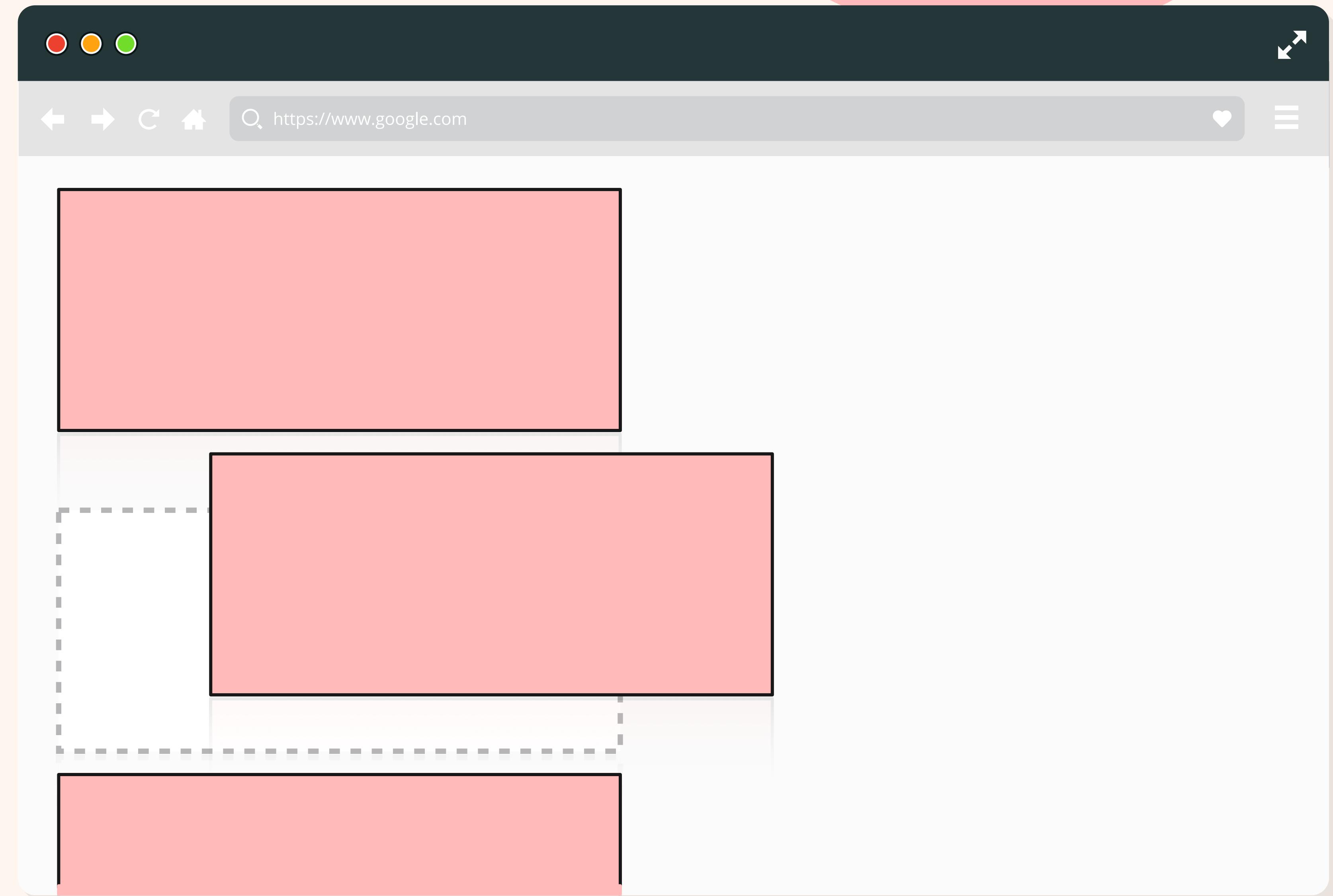
Elementen eromheen gedragen zich alsof het element er nog is



Position

Relative

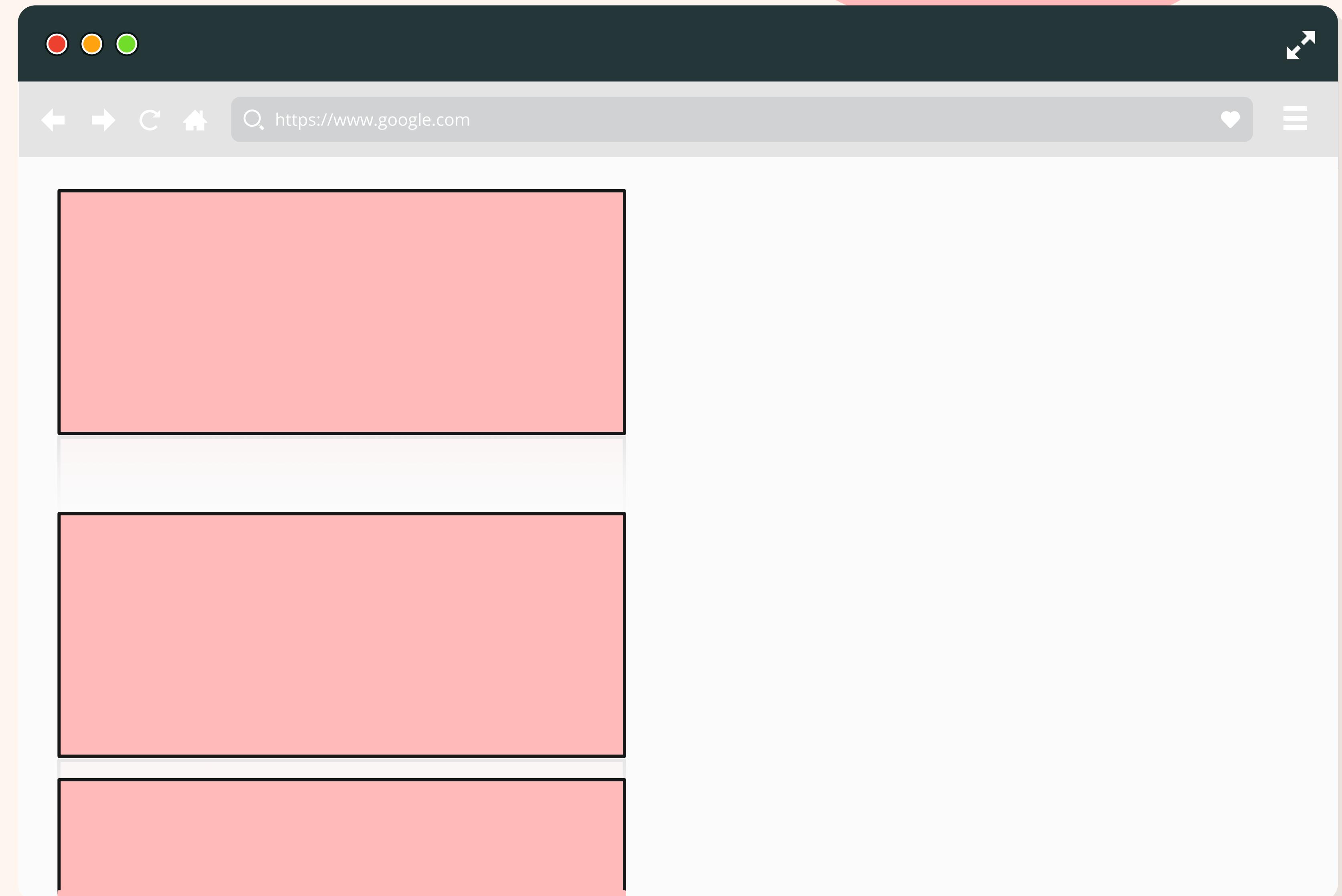
Elementen eromheen gedragen zich alsof het element er nog is



Position

Absolute

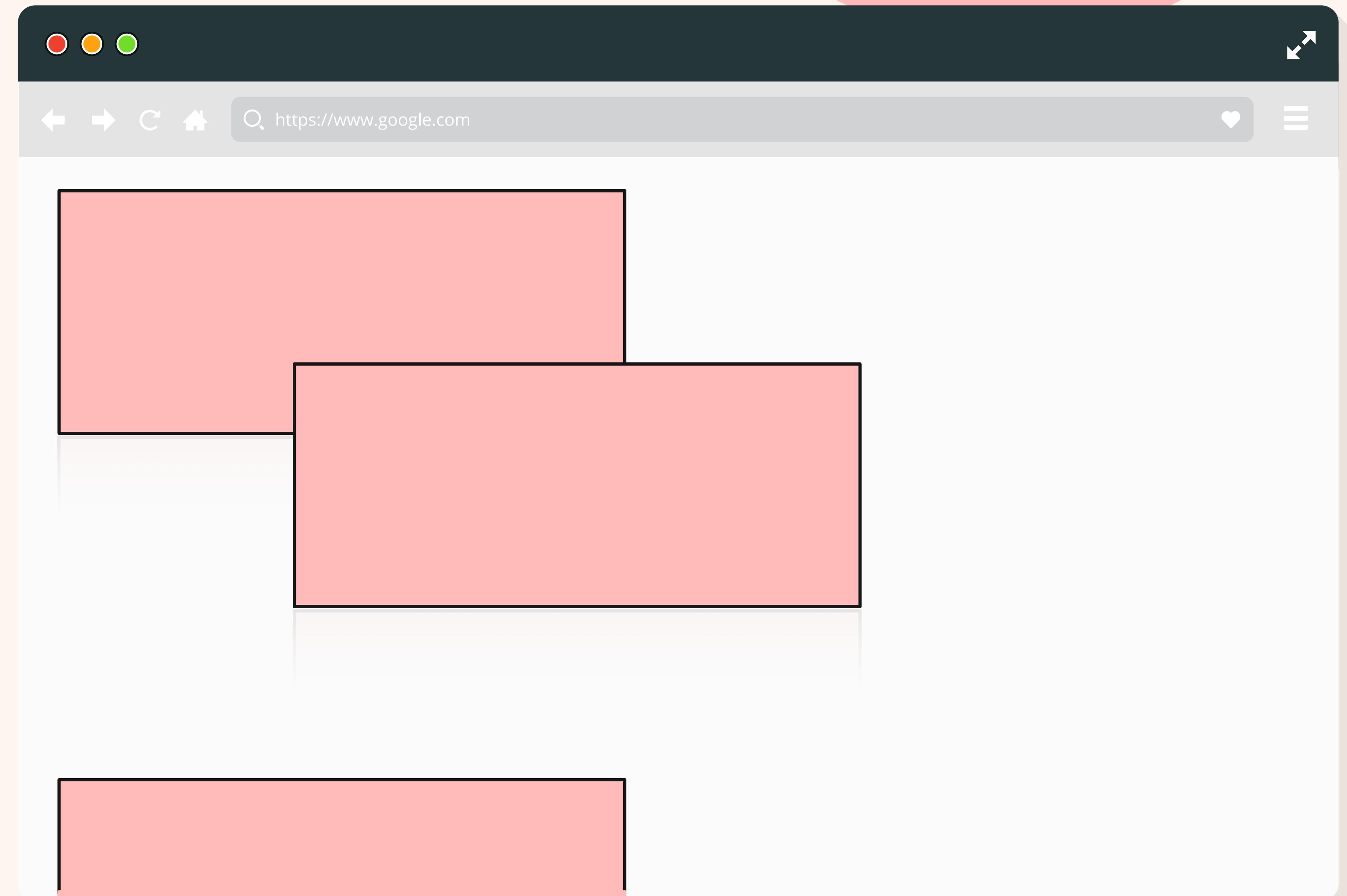
Elementen eromheen gedragen zich alsof het element volledig uit de "flow" van de pagina is verdwenen



Position

Absolute

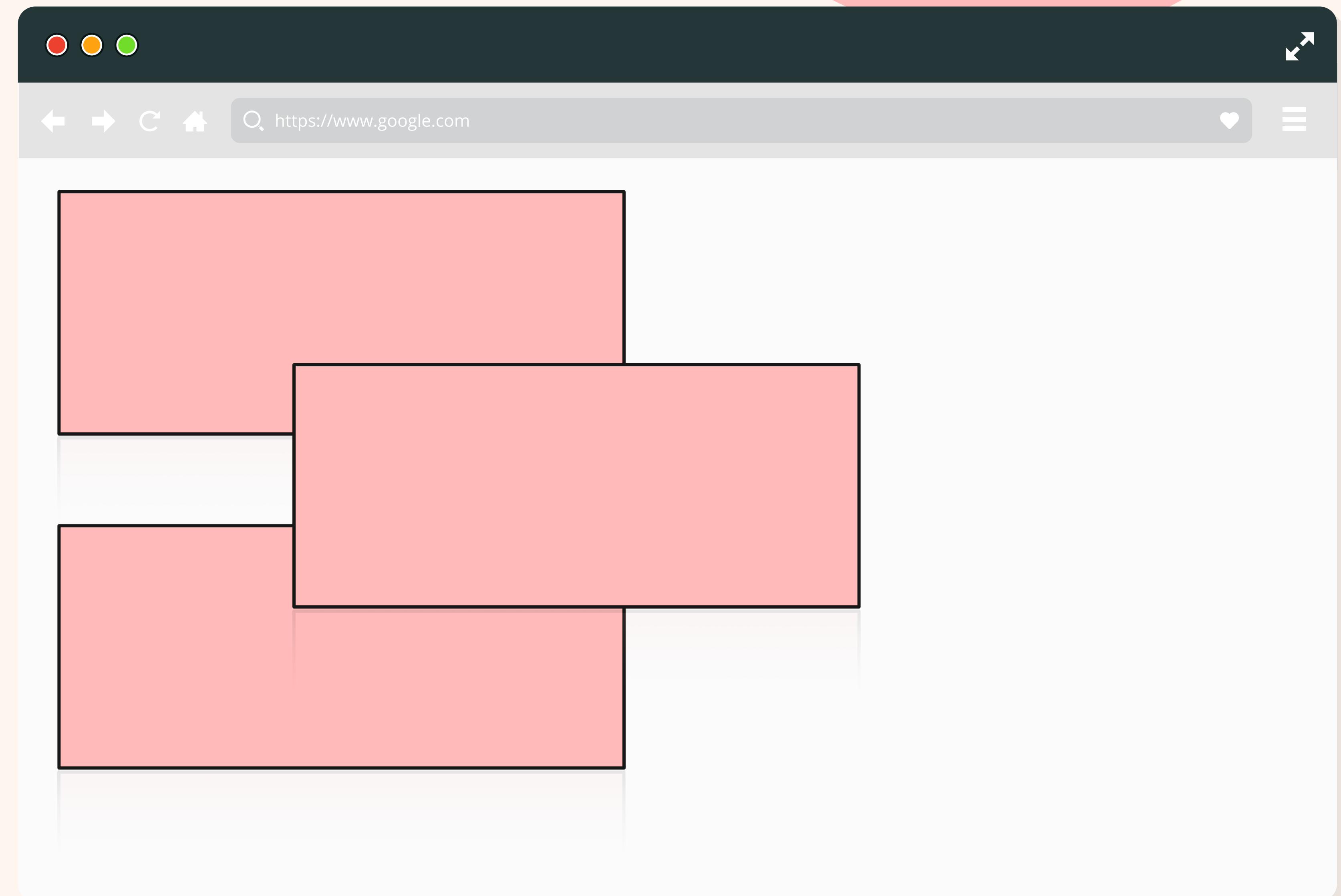
Elementen eromheen gedragen zich alsof het element volledig uit de "flow" van de pagina is verdwenen



Position

Absolute

Elementen eromheen gedragen zich alsof het element volledig uit de "flow" van de pagina is verdwenen

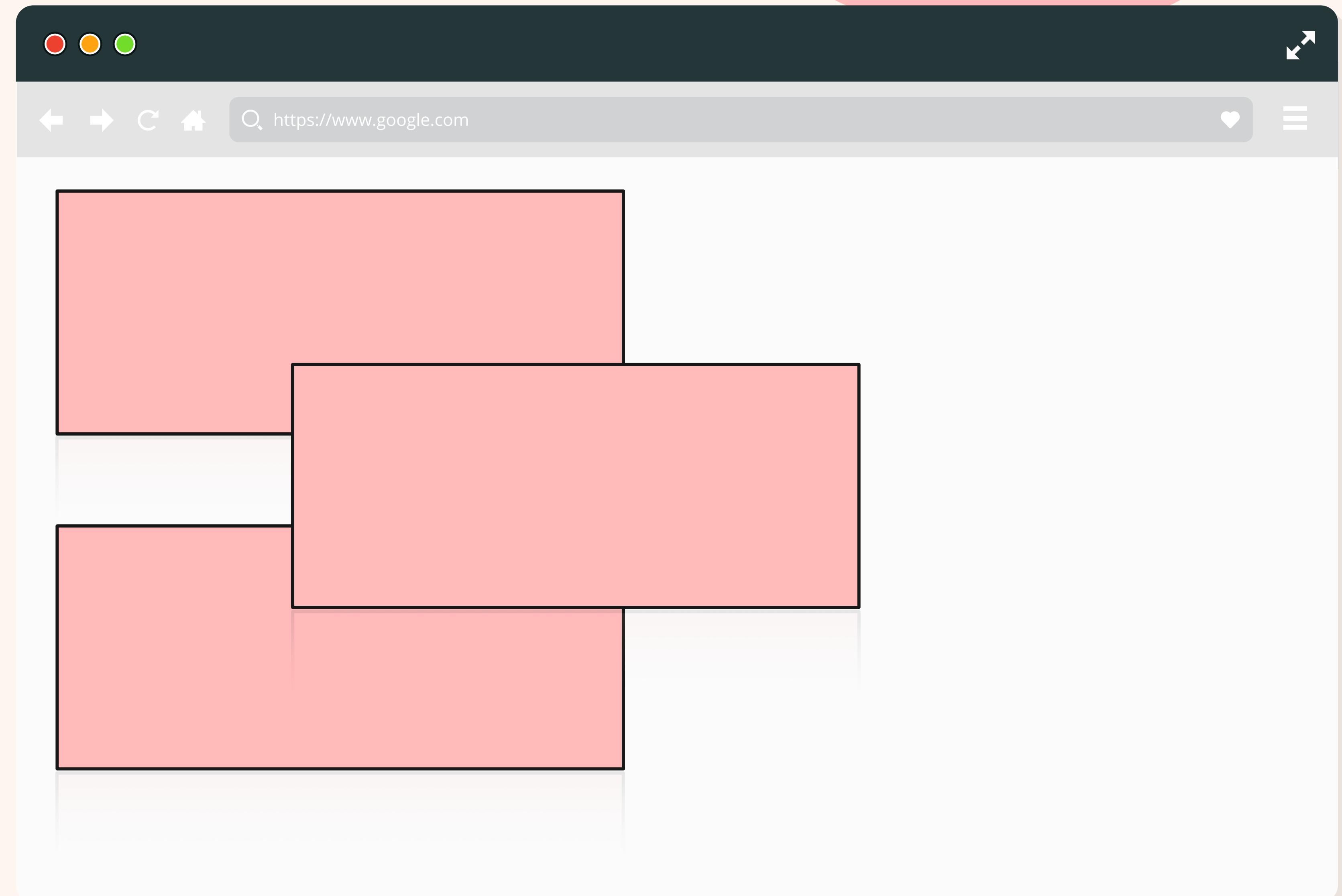


Position

Absolute

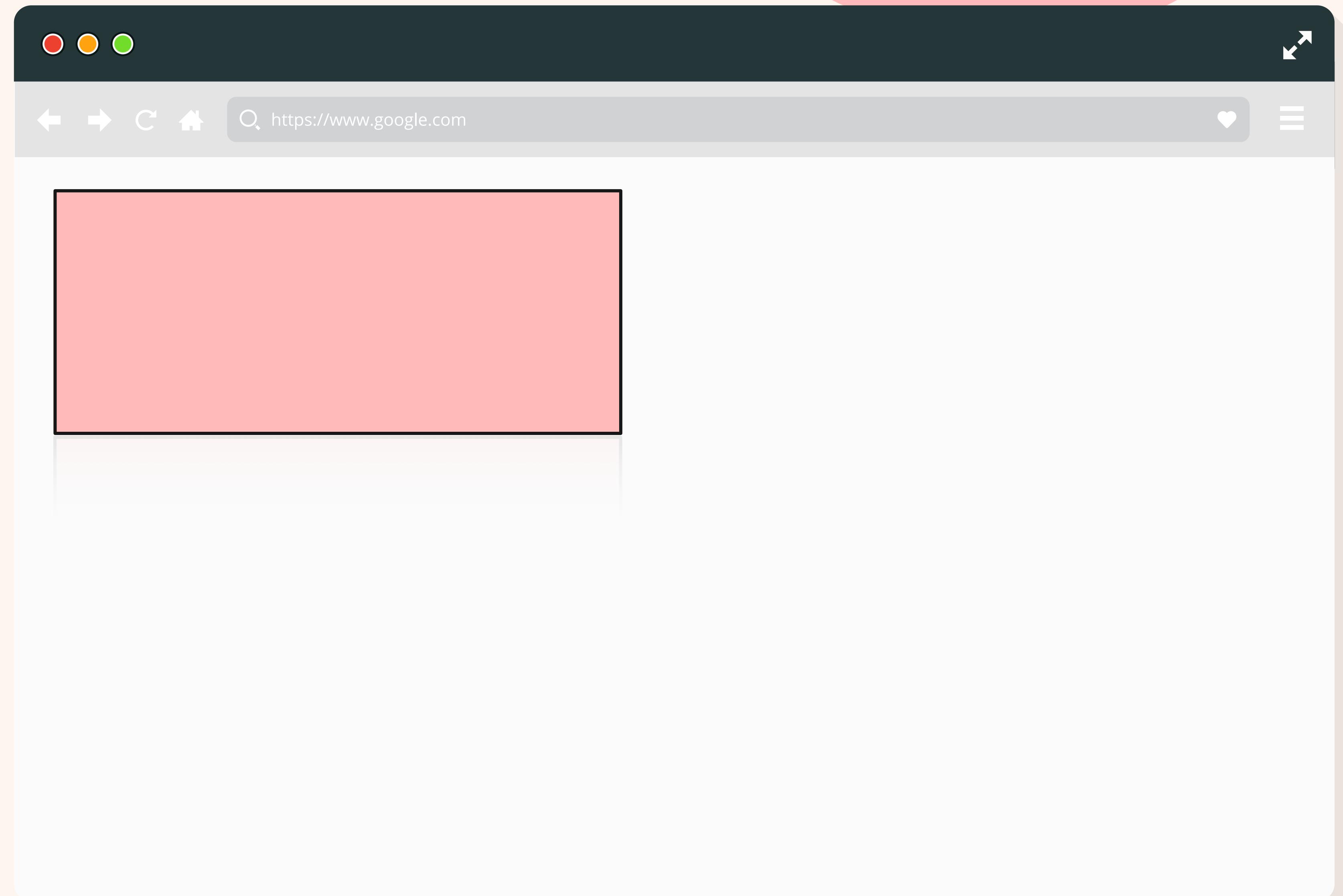
Elementen eromheen gedragen zich alsof het element volledig uit de "flow" van de pagina is verdwenen

Maar position absolute doet soms een beetje vreemd...



Position

Absolute



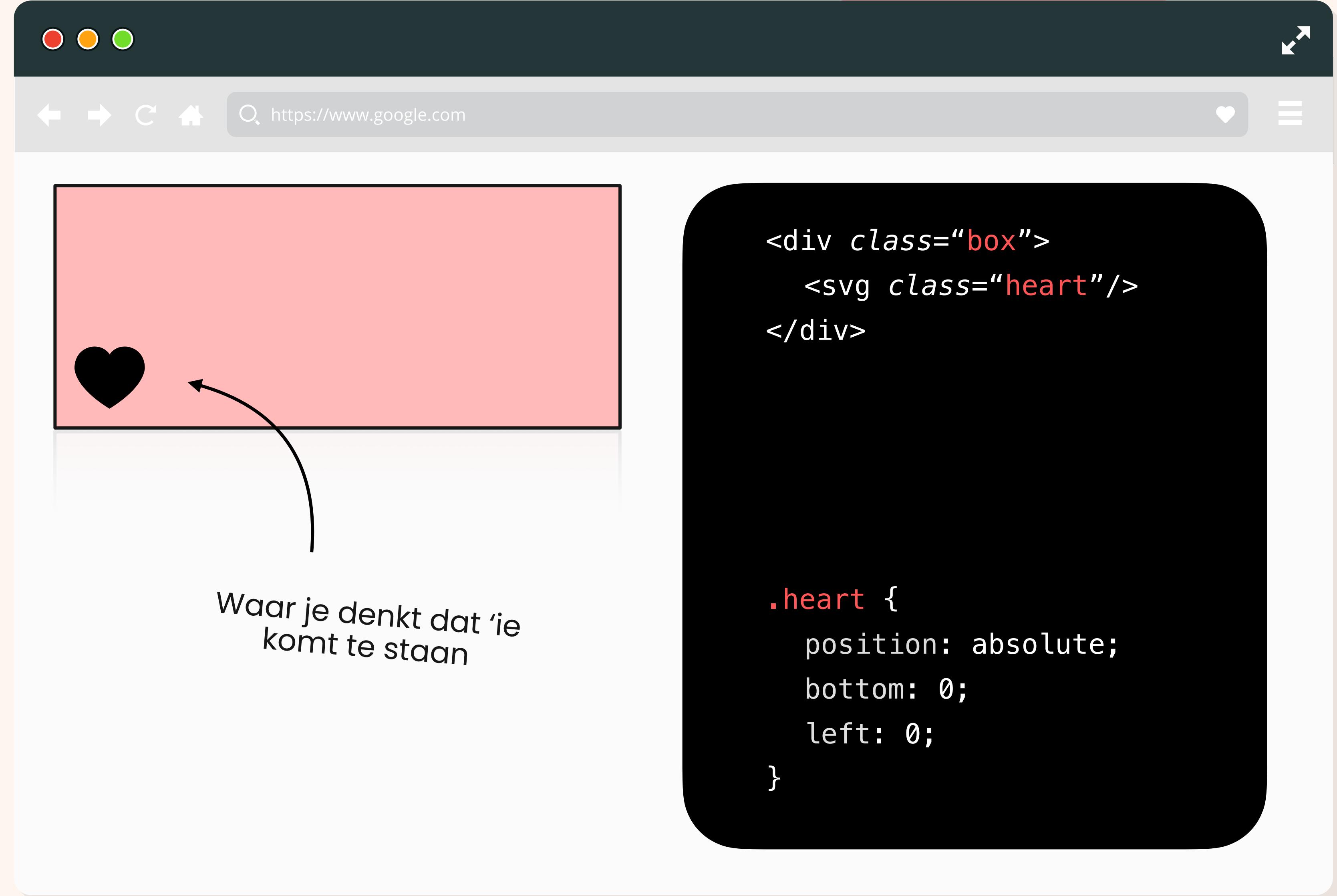
Position

Absolute



Position

Absolute



Waar je denkt dat 'ie komt te staan

```
<div class="box">
  <svg class="heart"/>
</div>

.heart {
  position: absolute;
  bottom: 0;
  left: 0;
}
```

Position

Absolute

Waar 'ie echt komt te staan

```
<div class="box">
  <svg class="heart"/>
</div>
```

```
.heart {
  position: absolute;
  bottom: 0;
  left: 0;
}
```

Position

Absolute

Waarom?

De `position: absolute;` positioneert zichzelf ten opzichte van het eerste parent element met `position: relative;`



Position

Absolute

Waarom?

De `position: absolute;` positioneert zichzelf ten opzichte van het eerste parent element met `position: relative;`

Heb je dat nergens gespecificeerd? Dan positioneert hij zich ten opzichte van de body!

Waar 'ie echt komt te staan

```
<div class="box">
  <svg class="heart"/>
</div>

.heart {
  position: absolute;
  bottom: 0;
  left: 0;
}
```

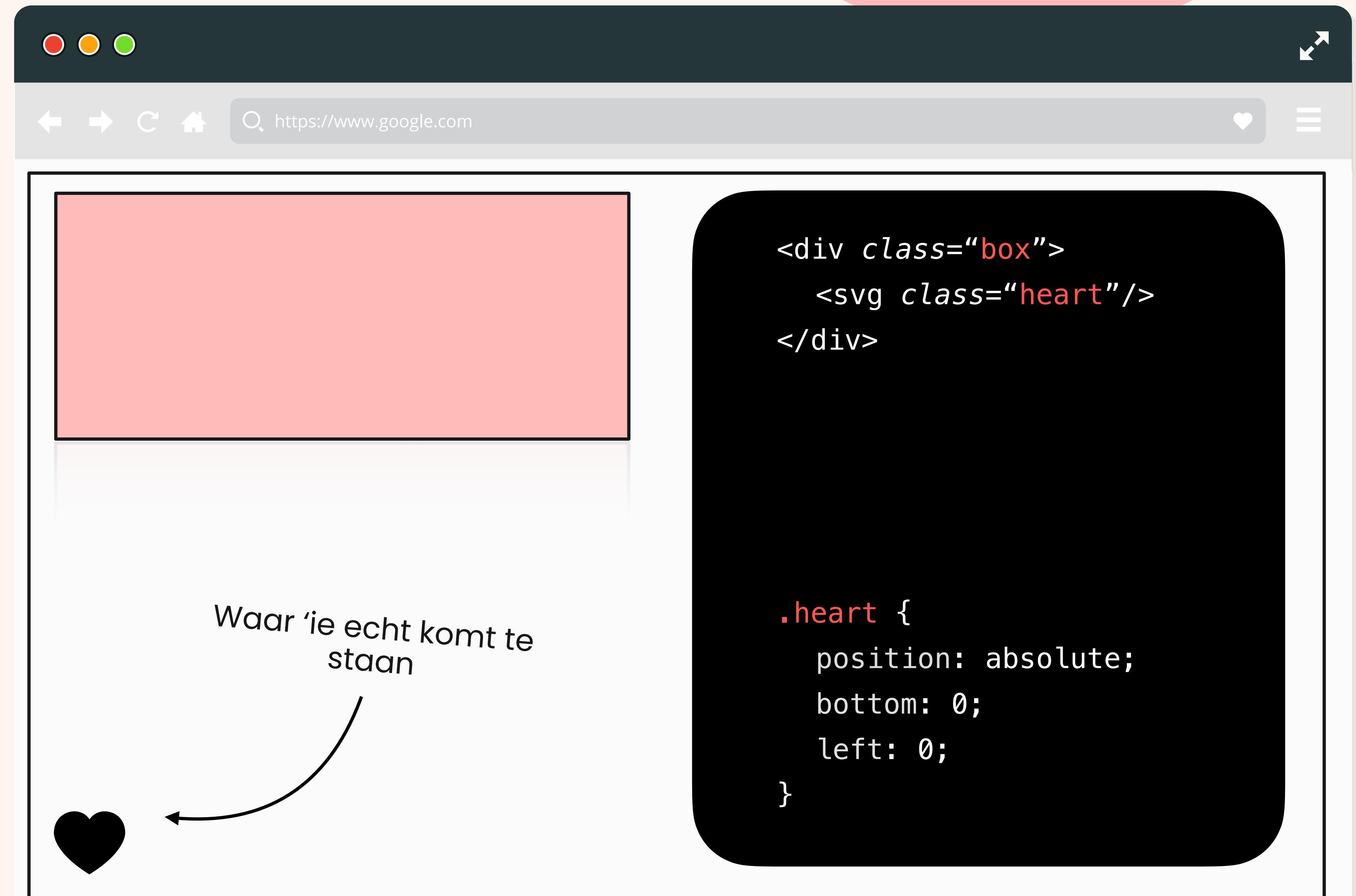
Position

Absolute

Waarom?

De `position: absolute;` positioneert zichzelf ten opzichte van het eerste parent element met `position: relative;`

Heb je dat nergens gespecificeerd? Dan positioneert hij zich ten opzichte van de body!



Position

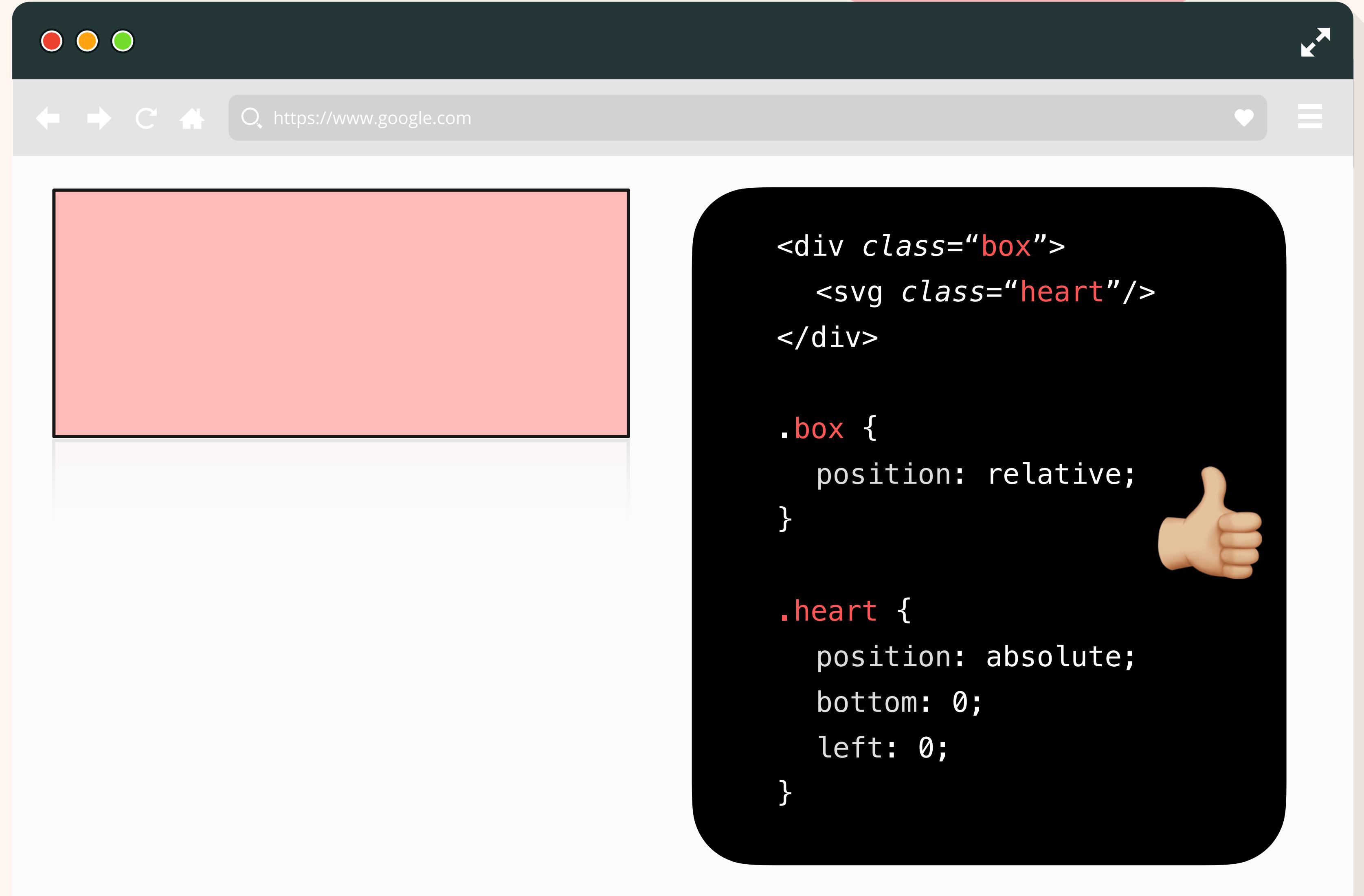
Absolute

Waarom?

De `position: absolute;` positioneert zichzelf ten opzichte van het eerste parent element met `position: relative;`

Heb je dat nergens gespecificeerd? Dan positioneert hij zich ten opzichte van de body!

De parent moet dus altijd nog een `position: relative` krijgen!



The screenshot shows a web browser window with the URL https://www.google.com. Inside the window, there is a large red rectangular box with a black border. To the right of the box, there is a small heart-shaped icon.

```
<div class="box">
  <svg class="heart"/>
</div>

.box {
  position: relative;
}

.heart {
  position: absolute;
  bottom: 0;
  left: 0;
}
```

A large black callout bubble on the right side contains the CSS code for creating the red box and the heart icon. The code uses the `position: absolute;` property for the heart icon, which is positioned relative to its parent element, the `.box` div. A thumbs-up emoji is placed next to the code.

Position

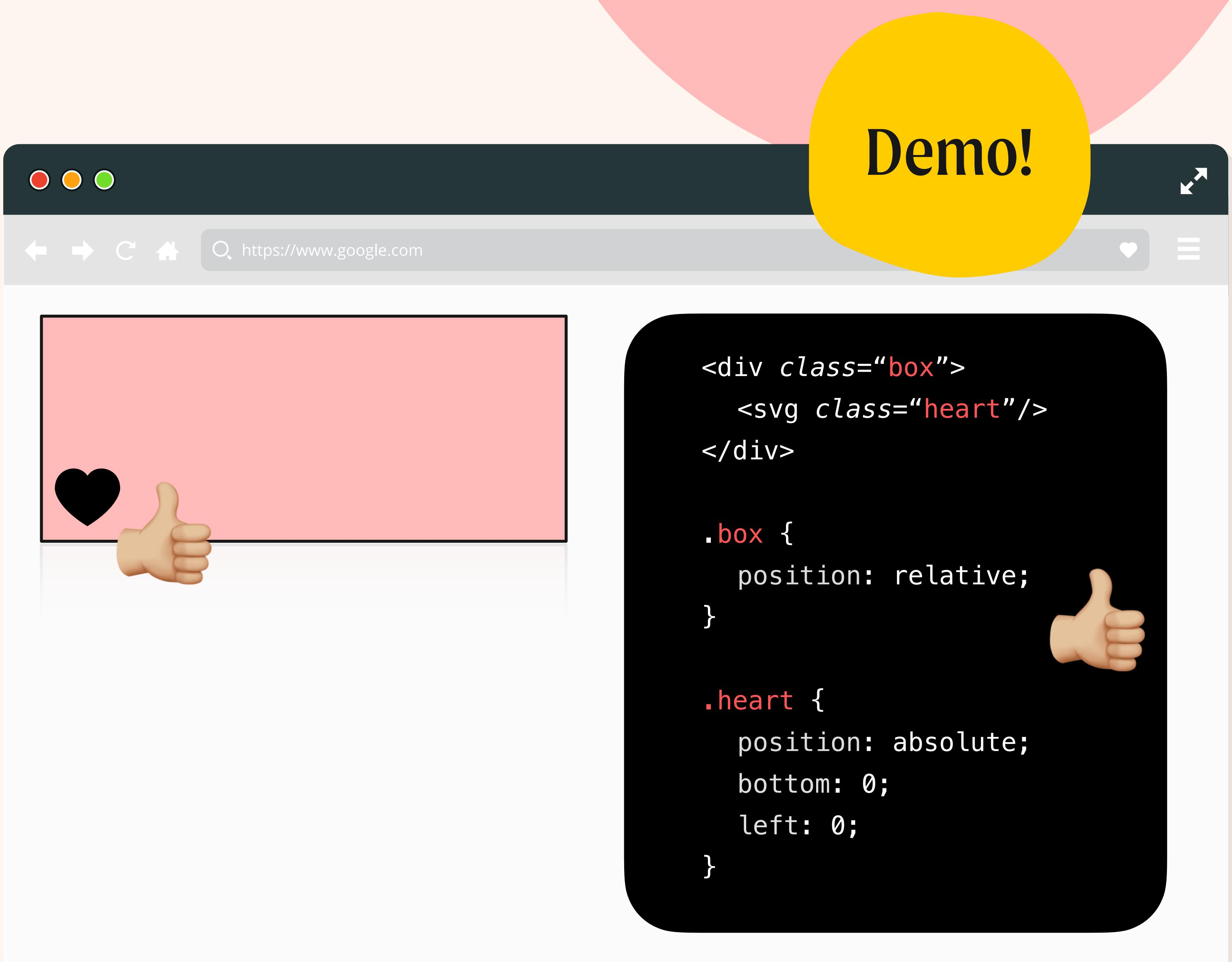
Absolute

Waarom?

De `position: absolute;` positioneert zichzelf ten opzichte van het eerste parent element met `position: relative;`

Heb je dat nergens gespecificeerd? Dan positioneert hij zich ten opzichte van de body!

De parent moet dus altijd nog een `position: relative` krijgen!



The image shows a web browser window with a yellow circular overlay containing the word "Demo!". The browser interface includes a title bar with three dots, a toolbar with back, forward, and search buttons, and a URL bar showing "https://www.google.com". The main content area displays a large pink rectangular box with a black heart icon inside. A thumbs-up emoji is positioned below the box, partially overlapping its bottom edge. To the right of the browser window is a black rounded rectangle containing the following CSS code:

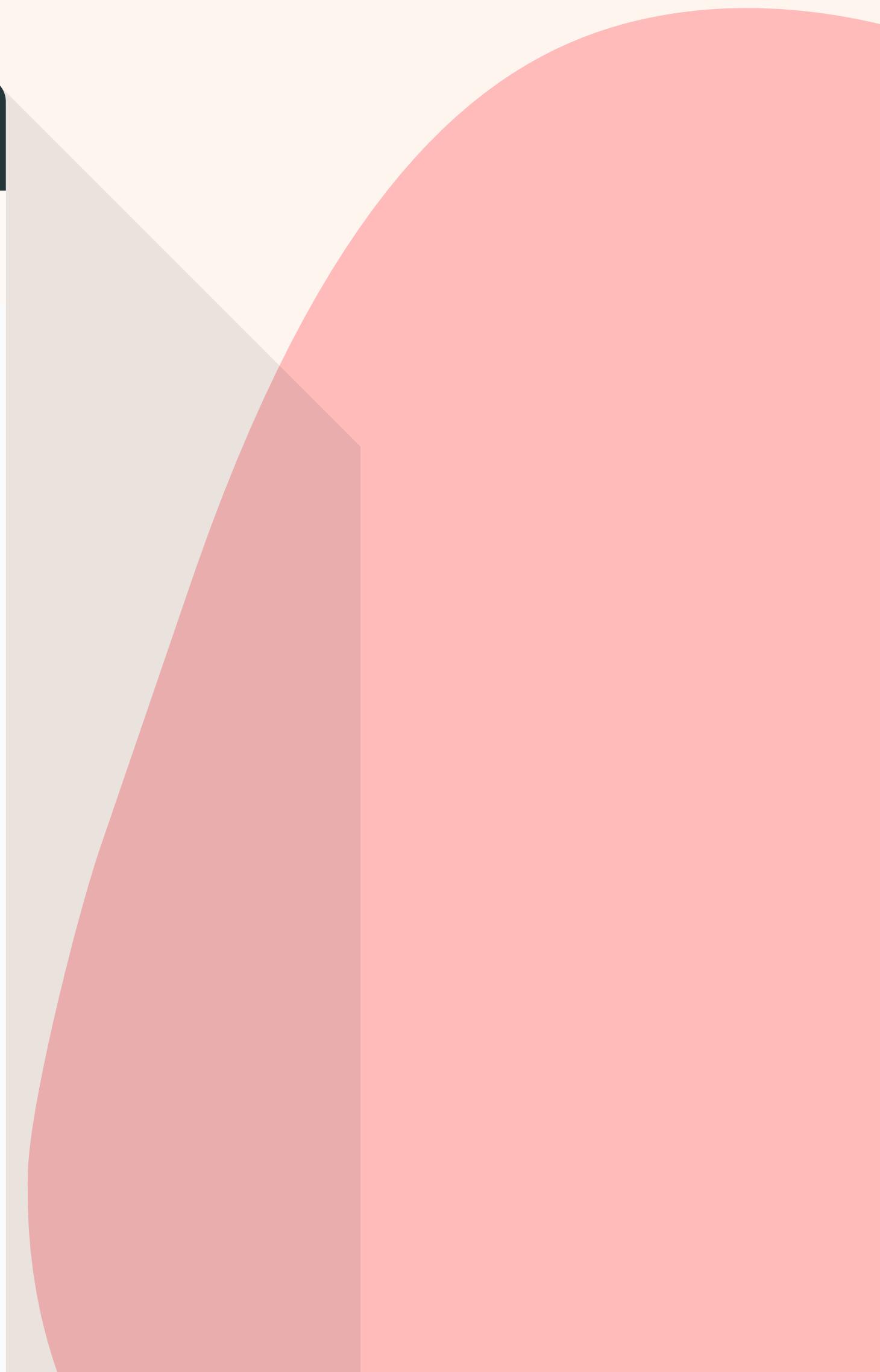
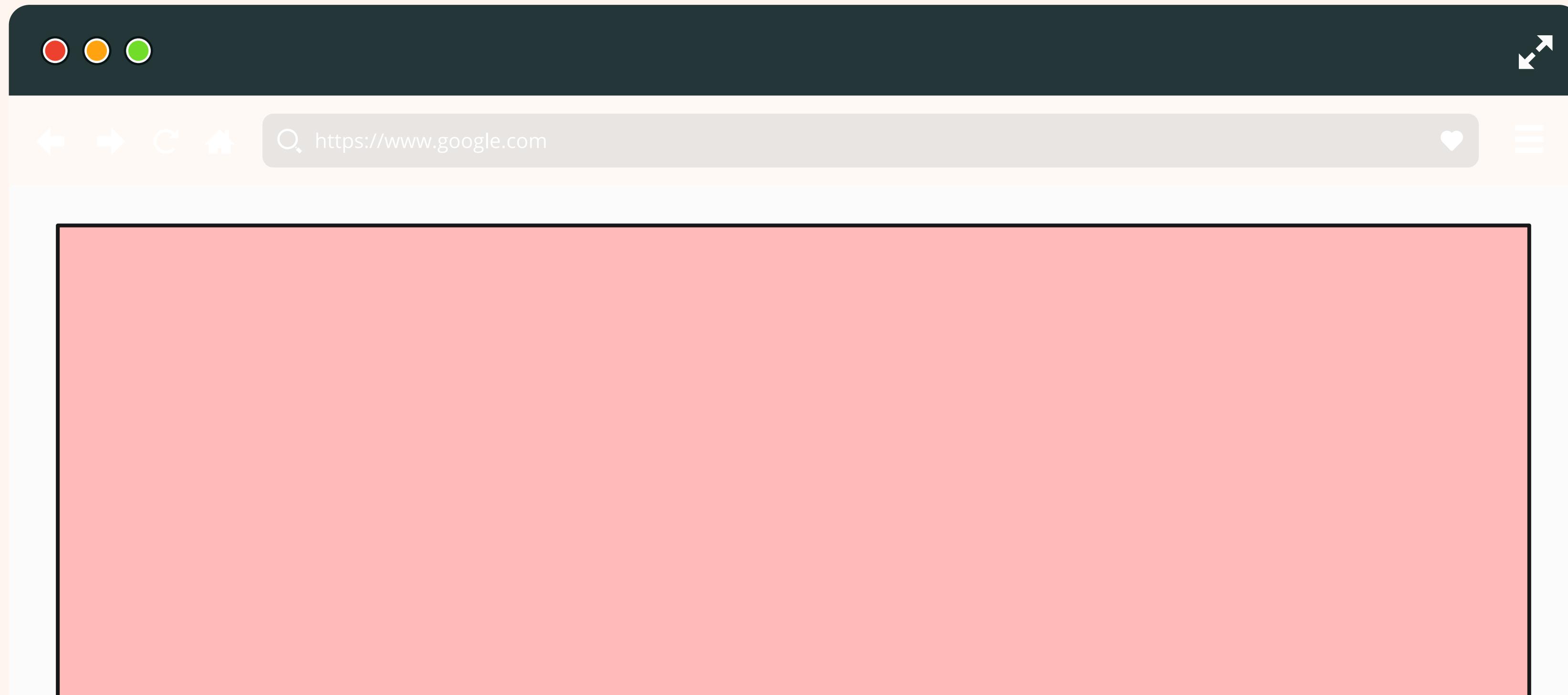
```
<div class="box">
  <svg class="heart"/>
</div>

.box {
  position: relative;
}

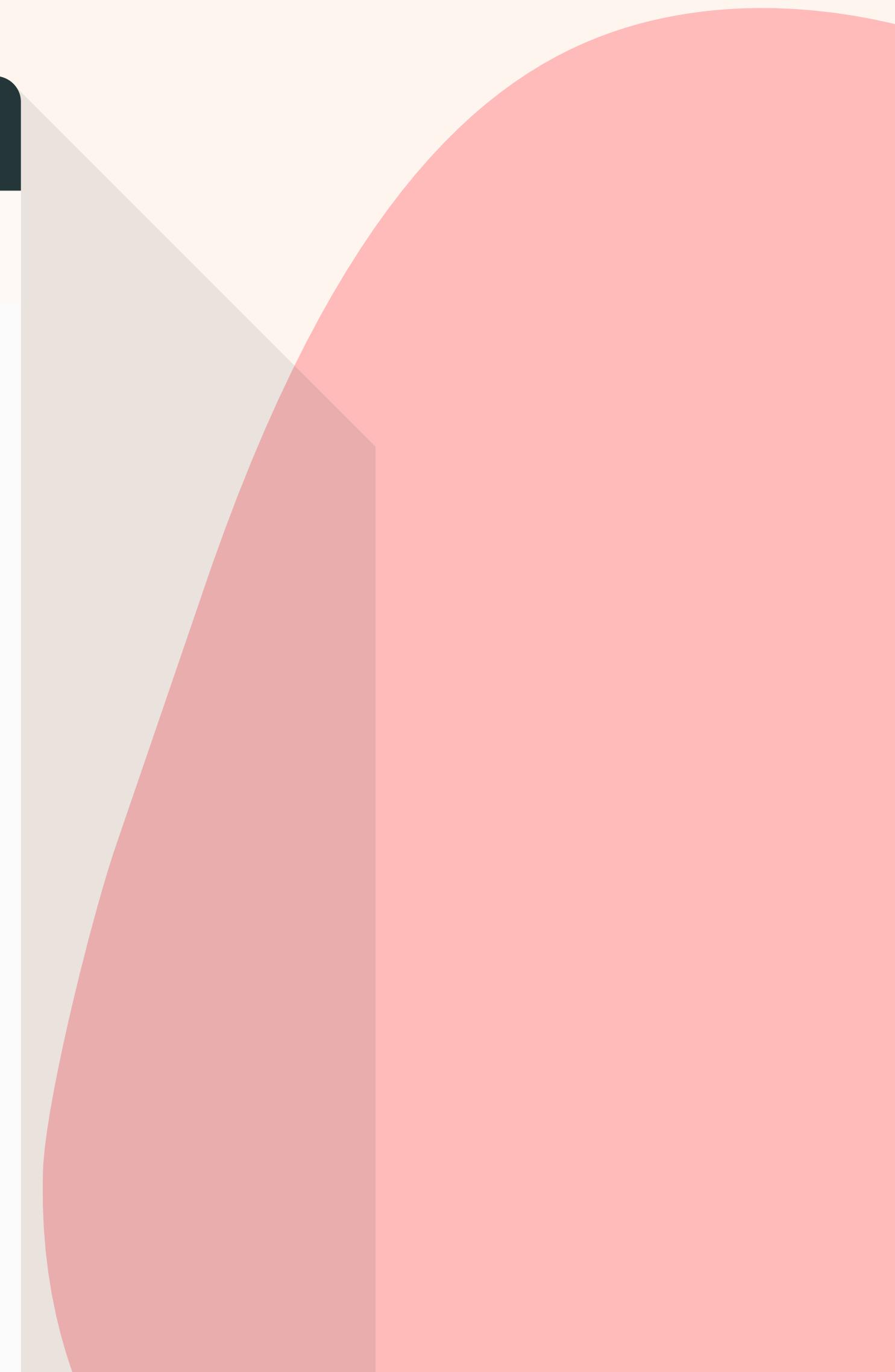
.heart {
  position: absolute;
  bottom: 0;
  left: 0;
}
```

A thumbs-up emoji is also placed next to the closing brace of the .heart selector.

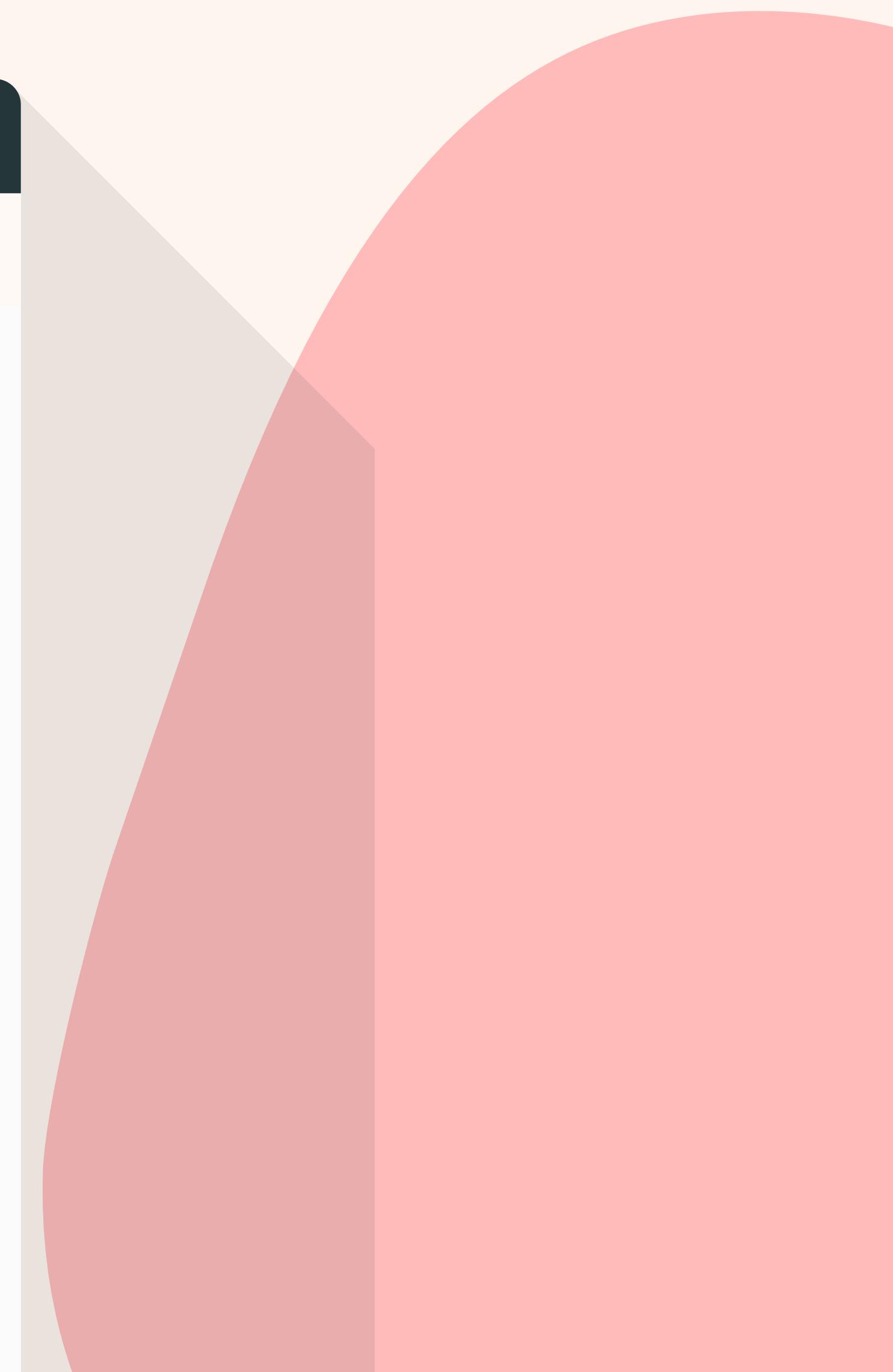
Schuine hoeken



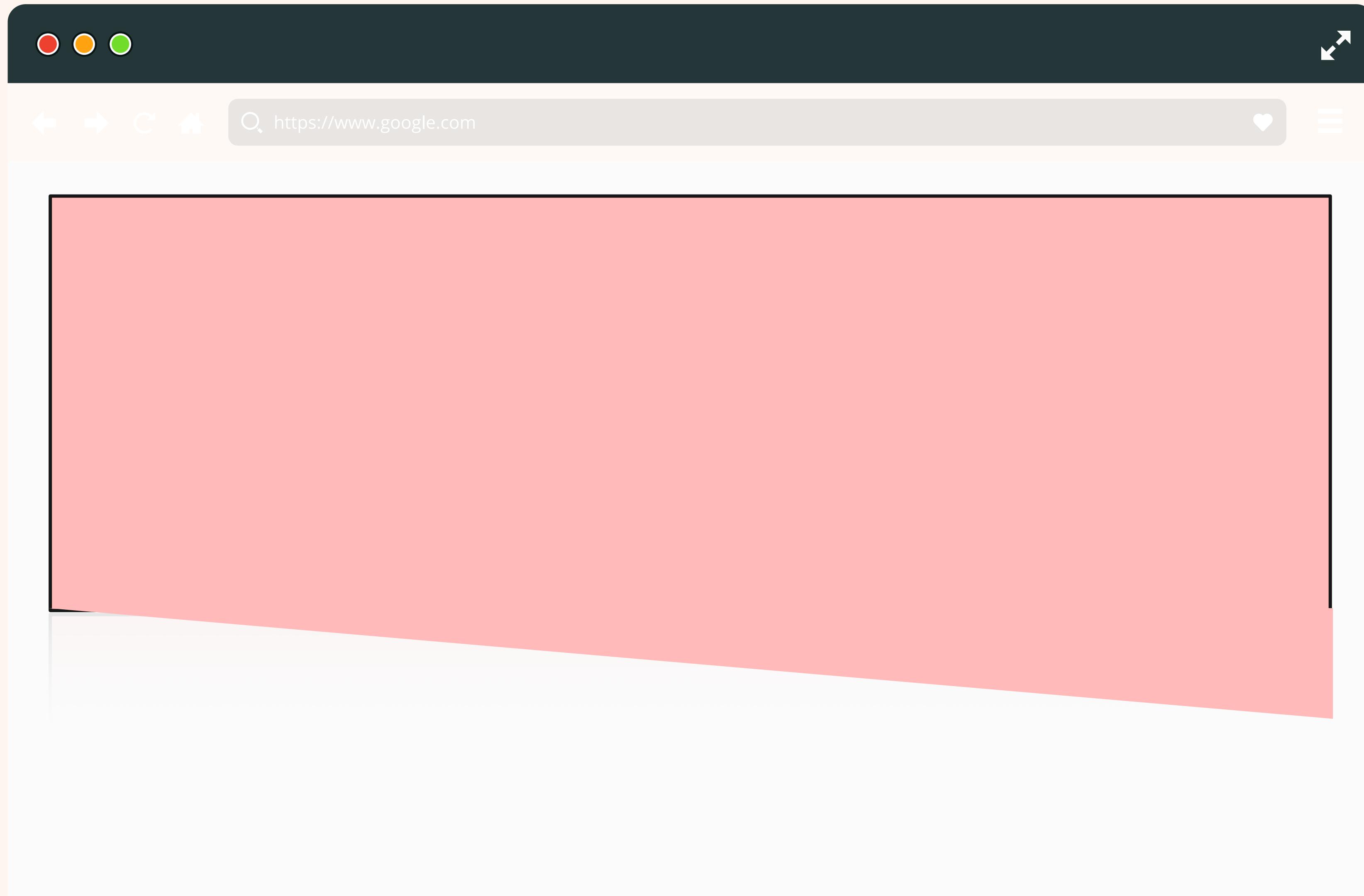
Schuine hoeken



Schuine hoeken



Schuine hoeken



90 min

Opdracht - Concept opdracht

<https://github.com/hogeschoolnovi/frontend-html-css-concept-opdracht>

- Clone de opdracht naar jouw lokale machine via VSC
- Lees de opdracht beschrijving en maak het voorbeeld na
- Je wordt ingedeeld in breakout rooms
- Overleg zo veel mogelijk met je groepje, maar tag de docent in de breakout-chat wanneer je vastloopt of ergens extra uitleg over wil!

Huiswerk volgende les

- Maak de lesopdracht af en lever deze uiterlijk woensdag 15 juni in via Teams! Dit is een feedback opdracht en je hebt er 3 weken de tijd voor!
- Lees **hoofdstuk 10** op EdHub. Het werken met een Figma is zelfstudie en zullen we niet tijdens de les behandelen.

ontwerptool als

Wil je nog veel meer oefenen met Flexbox? **Dat kan!**

<https://github.com/hogeschoolnovi/frontend-html-css-extra-practise>

Planning

Planning

- Je hebt 3 weken de tijd om deze opdracht te maken
- Lukt het niet om alle delen af te krijgen? Dan lever je gewoon in wat wél gelukt is!

Planning

- Je hebt 3 weken de tijd om deze opdracht te maken
- Lukt het niet om alle delen af te krijgen? Dan lever je gewoon in wat wél gelukt is!

Tips en tricks voor deel 2

Huiswerkklas na
[HTML & CSS] les 3

Tips en tricks voor deel 3

Huiswerkklas na
[HTML & CSS] les 4

Bespreken eindresultaat

Huiswerkklas na
[JavaScript] les 1