

HTML5 Semantic Tags

div, span, class, id

`<div></div>` Refers to an empty block element that contains no inherent style

`` Also an empty element with no inherent style, used inline

`<div class = "myClass">` Classes are used to distinguish specific style that is applied only to that class. Classes should be defined when there will be multiple elements on your page that require the same style.

`<div id = "thisGreatID">` IDs are also used to distinguish specific style but are more specific than classes and should be applied when those style elements will only be used once

Metadata: `viewport`

The user's visible area of a web page

HTML5 introduced a method to let web designers take control over the viewport, through the `<meta>` tag.

<!

- - Tells the browser to match the device's width for the viewport
- Sets an initial zoom value -->

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

<meta name="viewport" content="width=device-width, initial-scale=1.0">

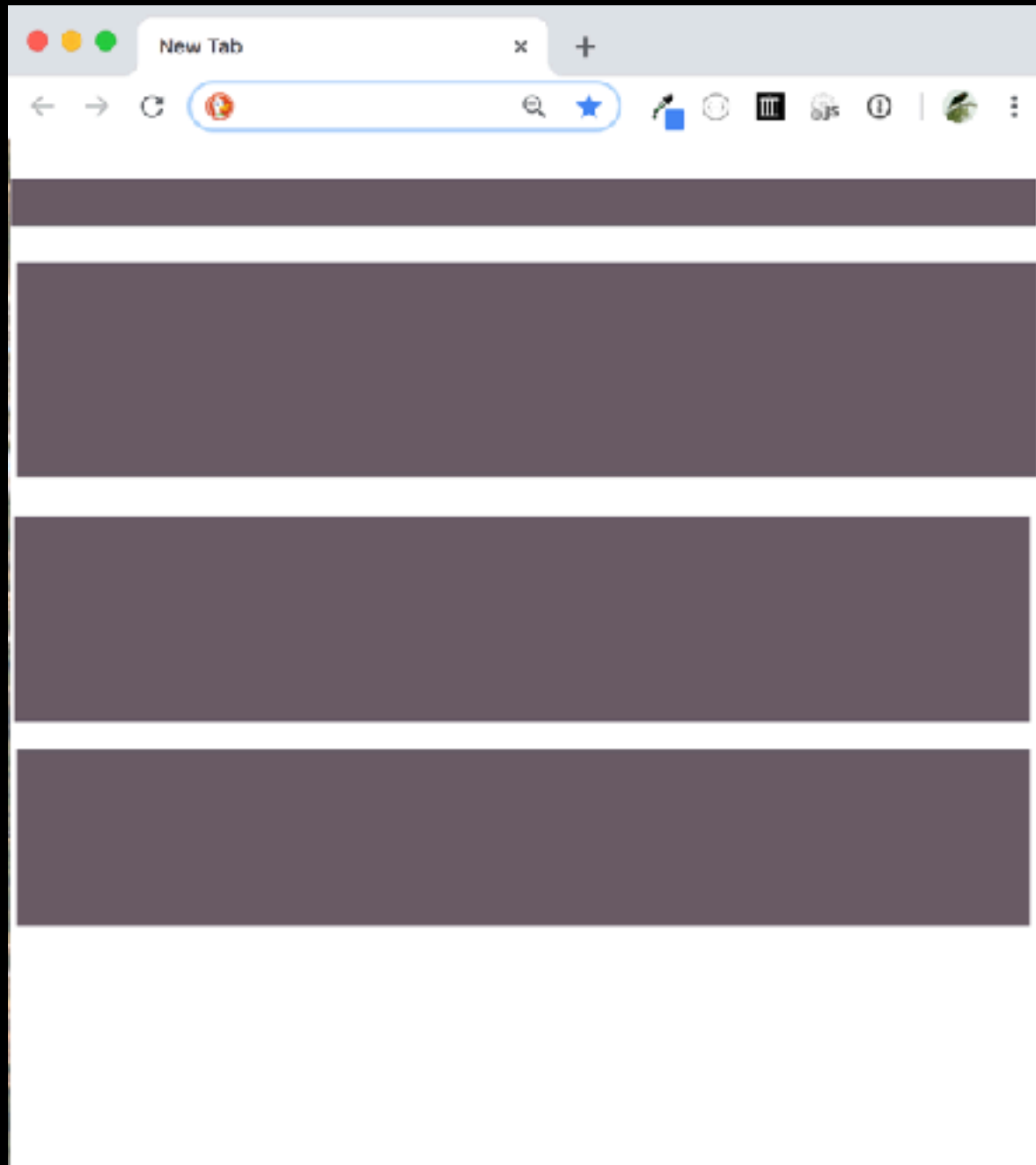


without



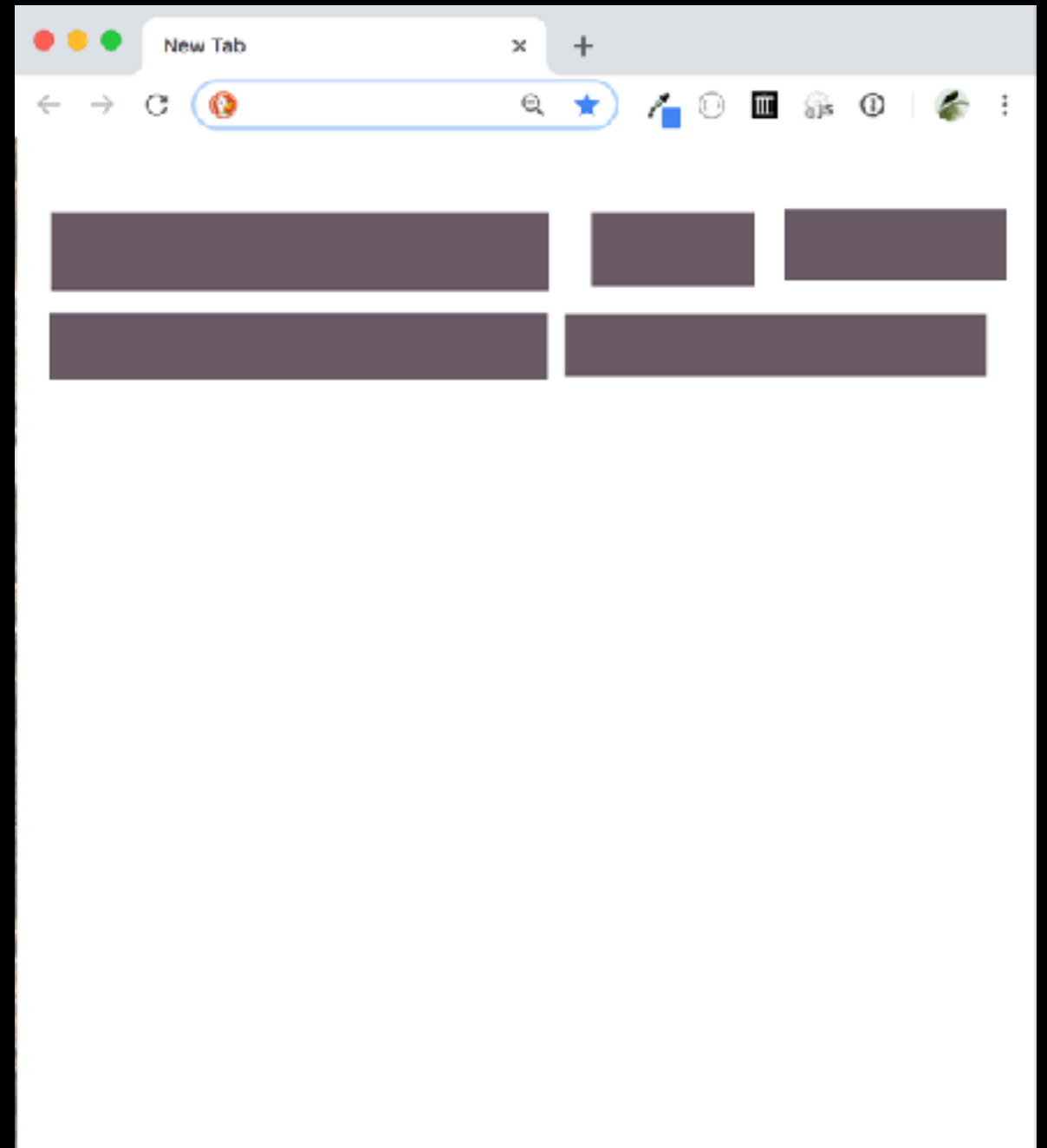
with

css Box Model



Block layout

Laying out large sections of a page



Inline layout

Laying out txt + other inline content within a section

Border

All boxes have borders even if invisible or 0px wide. It separates the edge of one box from another.

Padding

Padding is the space btw the border + any content contained within it. More padding increases the readability of its contents.

Margin

Margins sit outside the edge of the border. You can set the width to create a gap btw borders of adjacent boxes.

Box Model



Content

Box Dimensions

By default the box is sized just big enough to hold its contents. Use the **height** + **width** properties to set your own dims.

min-width, max-width
min-height, max-height

- **`px`** - traditionally, the most popular way of specifying the size of a box
- **`%`** - the size of the box is relative to the size of the browser window, or if the box is inside another box.
- **`em`** - the size of the box is based on the size of the text inside it.
Designers have recently started to use **%** + **ems** more as they are flexible across platforms + devices
- **`rem`** - Relative to font-size of the root element. [check it here](#)

vh and vw

You can define height and width in terms of the viewport

- Use units **vh** and **vw** to set height and width to the percentage of the viewport's height and width, respectively
- $1\text{vh} = 1/100\text{th}$ of the viewport height
- $1\text{vw} = 1/100\text{th}$ of the viewport width

Example:

- height: 100vh; - width: 100vw;

Border

can also specify each border individually:

border-top

border-bottom

border-left

border-right

+ set each property individually:

border-style: dotted; (all styles)

border-width: 3px;

border-color: purple;

Border

```
<p>  
  This is then a very little description of  
  feeling disillusionment in living.  
</p>
```

```
p {  
  border: 4px solid black;  
}
```

When we add a border to an element, it sits flush against the text.

This is then a very little description of feeling disillusionment in living.

Padding Padding is the space btw the border + the content.

```
p {  
  border: 4px solid black;  
  padding: 10px;  
}
```

This is then a very little description of feeling
disillusionment in living.

Can specify

padding-top

padding-bottom

padding-left

padding-right

There's also a shorthand:

padding: 2px 4px 3px 1px;

padding: 10px 2px;

<- top | right | bottom | left

<- top + bottom | left + right

```
<div class="marginEx">
Lectures
</div>
<div class="marginEx">
  Homework
</div>
```

```
.marginEx {
  border: 2px solid black;
  padding: 10px;
}
```

Lectures

Homework

When we add a border to multiple divs (w/ the same class), they sit flush against each other:

Margin

Margin is the space btw the border + other elements

```
.marginEx {  
  margin: 20px;  
  border: 2px solid black;  
  padding: 10px;  
}
```

Lectures

Homework

Can specify

margin-top

margin-bottom

margin-left

margin-right

There's also a shorthand:

margin: 2px 4px 3px 1px;

margin: 10px 2px;

<- **top** | **right** | **bottom** | **left**

<- **top + bottom** | **left + right**

Margin Collapsing

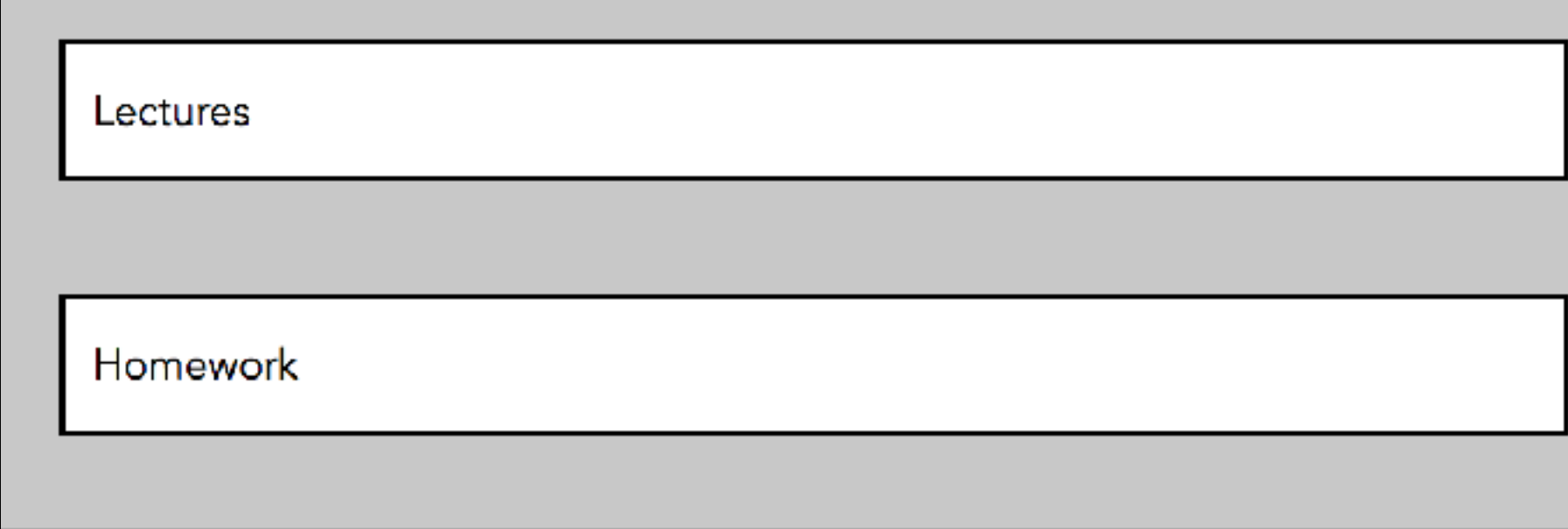
Sometimes the top + bottom margins of block elements are combined (“collapsed”) into a single margin.

Generally if:

- the elements are siblings
- the elements are block-level (**not inline**)

Then they collapse into **max** (Bottom Margin, Top Margin).

20px margin-bottom +
20px margin-top =
40 px margin?



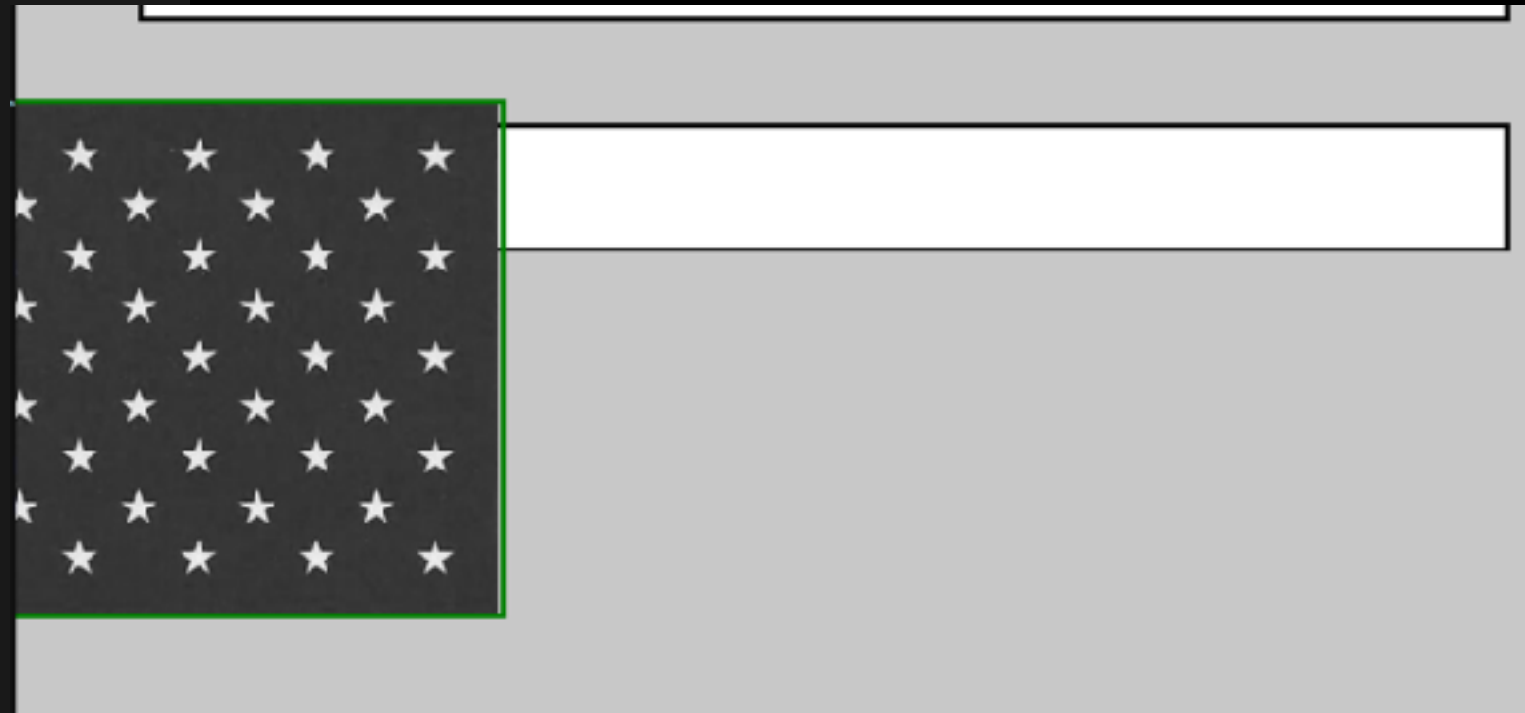
Lectures

Homework

Negative Margin

Margins can be negative as well.

```
img {  
  margin-top: -100px;  
  margin-left: -100px;  
  height: 200px;  
  border: 2px solid GREEN;  
}
```



-100 px margin-bottom
-100px margin-top

Auto Margins

If you set **margin-left** + **margin-right** to auto - you can center a block-level element

```
<html>
  <head>
    <meta charset="utf-8">
    <title>Auto Margins</title>
  </head>
  <body>
    <div>
      This is a box of text.
    </div>
  </body>
</html>
```

```
div {
  margin-left: auto;
  margin-right: auto;

  border: 2px solid black;
  padding: 10px;
  width: 300px;
}
```

This is a box of text.

Box Model for Inline Elements

Box Model applies to inline elements too, but the box is shaped differently.

```
<p> Week 03  
  <strong>"There is then as I am saying  
  complete disillusion in living, the  
  realis- ing, completely realising that  
  not any one, not one fighting for the  
  same thinking and believing as the  
  other... </strong>  
Which is a sentence my Gertrude Stein + is  
pretty long but is only part of the full one  
at that.  
</p>
```

```
strong {  
  border: 5px solid hotpink;  
  color:white;  
  padding: 8px;  
  margin: 24px;  
  background-color:pink;  
}
```

Week 03

"There is then as I am saying complete disillusion in living, the realis- ing, completely realising that not any one, not one fighting for the same

thinking and believing as the other... Which is a sentence my Gertrude Stein + is pretty long but is only part of the full one at that.

Inline Element Box Model

Margin is to the left + right of the inline element

- **margin-top** + **margin-bottom** are ignored.

Use **line-height** to manage btw spaces.

```
p {  
  width: 500px;  
  line-height: 50px;  
}  
  
strong {  
  border: 5px solid hotpink;  
  color: white;  
  padding: 8px;  
  margin: 24px;  
  background-color: pink;  
}
```

Week 03



"There is then as I am saying complete disillusion in

living, the realis- ing, completely realising that not any one, not one

fighting for the same thinking and believing as the other...



Which

is a sentence my Gertrude Stein + is pretty long but is only part of the full

one at that.

css positioning

Controlling the Position of Elements

Normal Flow

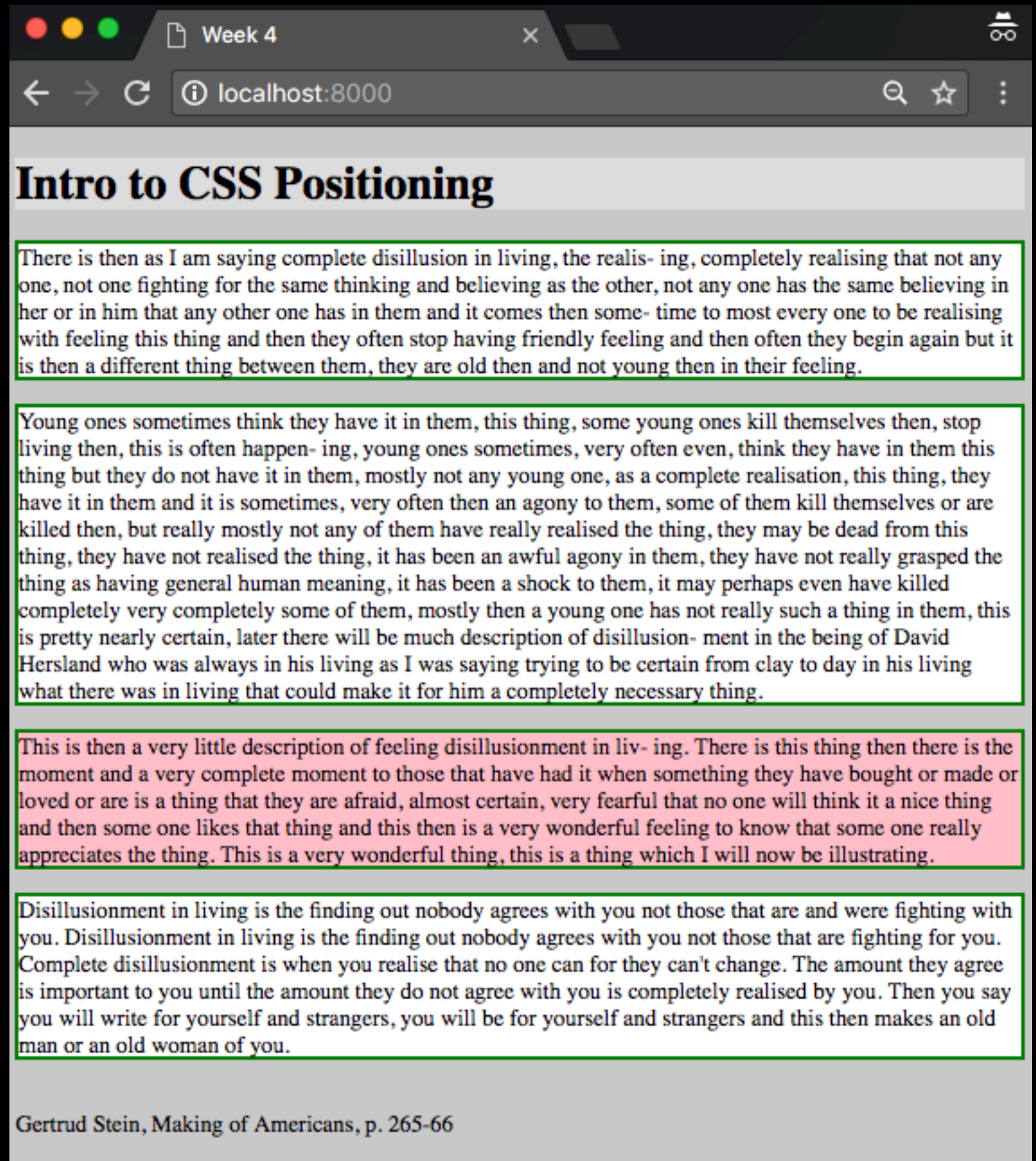
Every block-level element appears on a new line, causing each item to appear lower down on the page. Even if you specify the width of the boxes + there is space they will not appear next to each other.

```
.thePosition {
```

```
  background: pink;
```

```
  position: static;
```

```
}
```



Intro to CSS Positioning

There is then as I am saying complete disillusion in living, the realising, completely realising that not any one, not one fighting for the same thinking and believing as the other, not any one has the same believing in her or in him that any other one has in them and it comes then some- time to most every one to be realising with feeling this thing and then they often stop having friendly feeling and then often they begin again but it is then a different thing between them, they are old then and not young then in their feeling.

Young ones sometimes think they have it in them, this thing, some young ones kill themselves then, stop living then, this is often happen- ing, young ones sometimes, very often even, think they have in them this thing but they do not have it in them, mostly not any young one, as a complete realisation, this thing, they have it in them and it is sometimes, very often then an agony to them, some of them kill themselves or are killed then, but really mostly not any of them have really realised the thing, they may be dead from this thing, they have not realised the thing, it has been an awful agony in them, they have not really grasped the thing as having general human meaning, it has been a shock to them, it may perhaps even have killed completely very completely some of them, mostly then a young one has not really such a thing in them, this is pretty nearly certain, later there will be much description of disillusion- ment in the being of David Hersland who was always in his living as I was saying trying to be certain from clay to day in his living what there was in living that could make it for him a completely necessary thing.

This is then a very little description of feeling disillusionment in liv- ing. There is this thing then there is the moment and a very complete moment to those that have had it when something they have bought or made or loved or are is a thing that they are afraid, almost certain, very fearful that no one will think it a nice thing and then some one likes that thing and this then is a very wonderful feeling to know that some one really appreciates the thing. This is a very wonderful thing, this is a thing which I will now be illustrating.

Disillusionment in living is the finding out nobody agrees with you not those that are and were fighting with you. Disillusionment in living is the finding out nobody agrees with you not those that are fighting for you. Complete disillusionment is when you realise that no one can for they can't change. The amount they agree is important to you until the amount they do not agree with you is completely realised by you. Then you say you will write for yourself and strangers, you will be for yourself and strangers and this then makes an old man or an old woman of you.

Gertrud Stein, Making of Americans, p. 265-66

Relative Positioning

This moves an element from the position it would be in normal flow, shifting it to the top, right, bottom, or left where it would have been placed. This does not affect the position of surrounding elements; they stay in the position they would be in normal flow.

```
.thePosition {
```

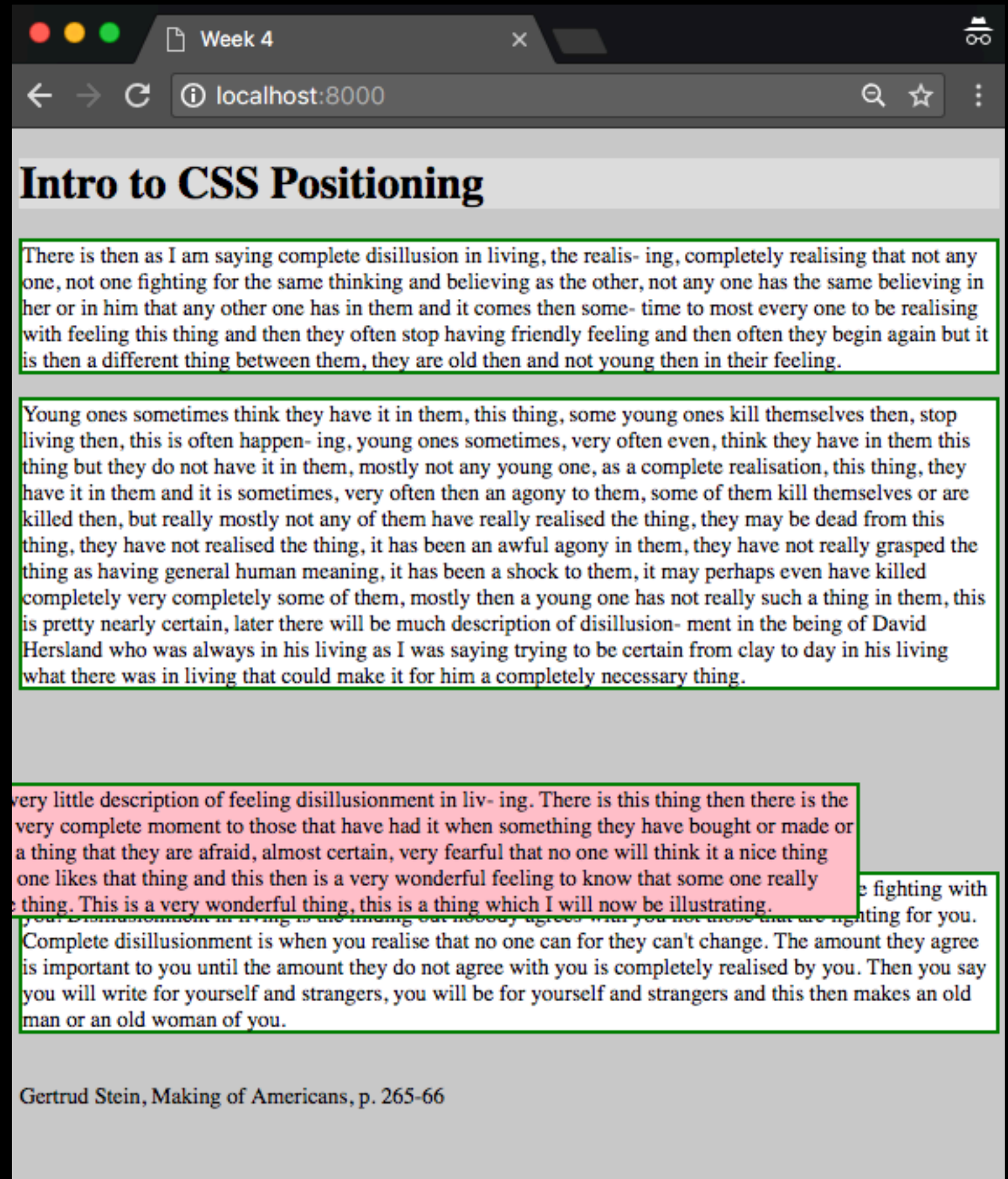
```
background: pink;
```

```
position: relative;
```

```
top: 50px;
```

```
right: 100px;
```

```
}
```



Absolute Positioning

This positions the element in relation to its containing element. It is taken out of normal flow, meaning that it does not affect the position of any surrounding elements. An absolutely positioned element no longer exists in the normal document layout flow. Instead, it sits on its own layer separate from everything else. Absolutely positioned elements move as users scroll up + down.

```
.thePosition {
```

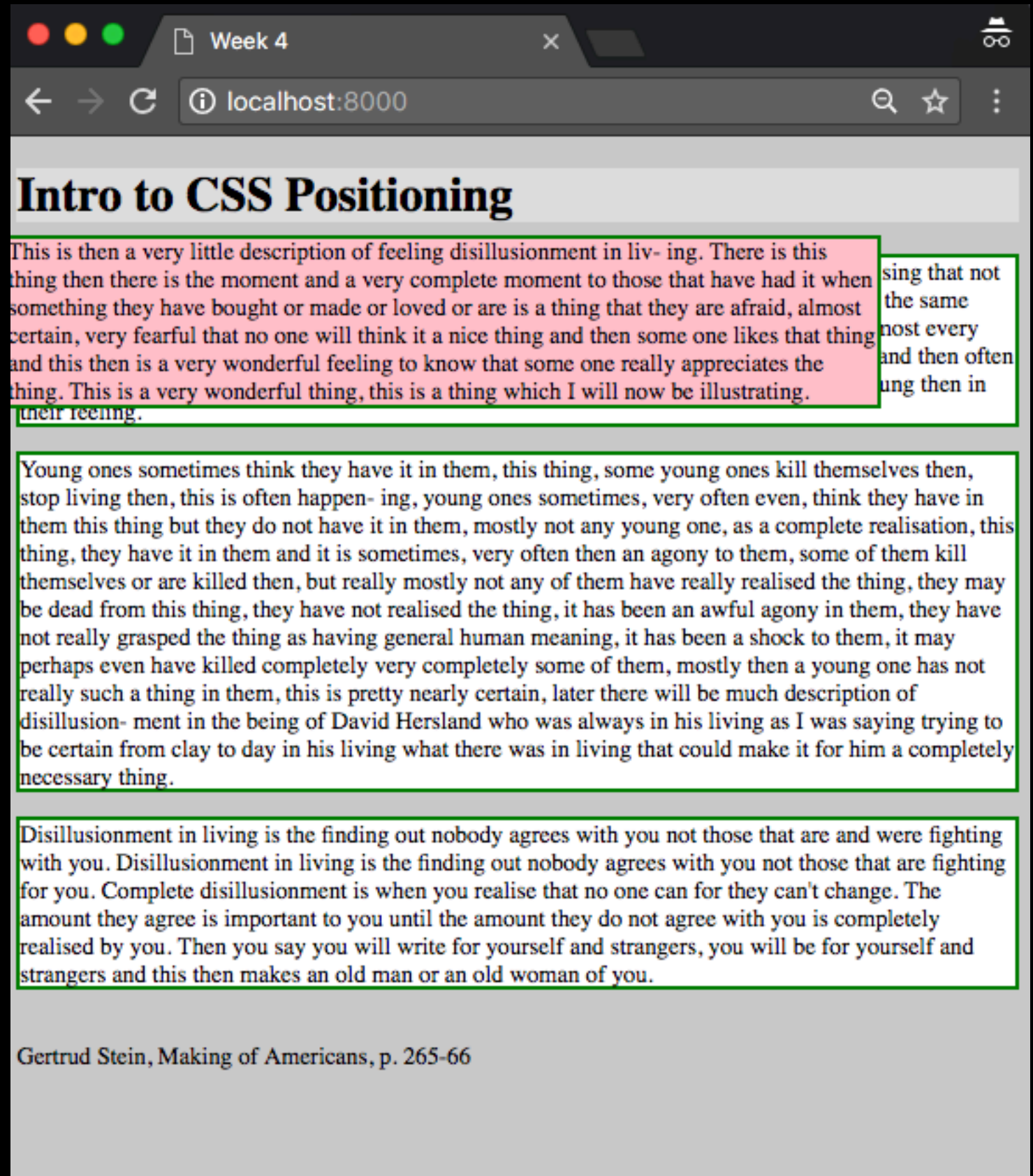
```
background: pink;
```

```
position: absolute;
```

```
top: 50px;
```

```
right: 100px;
```

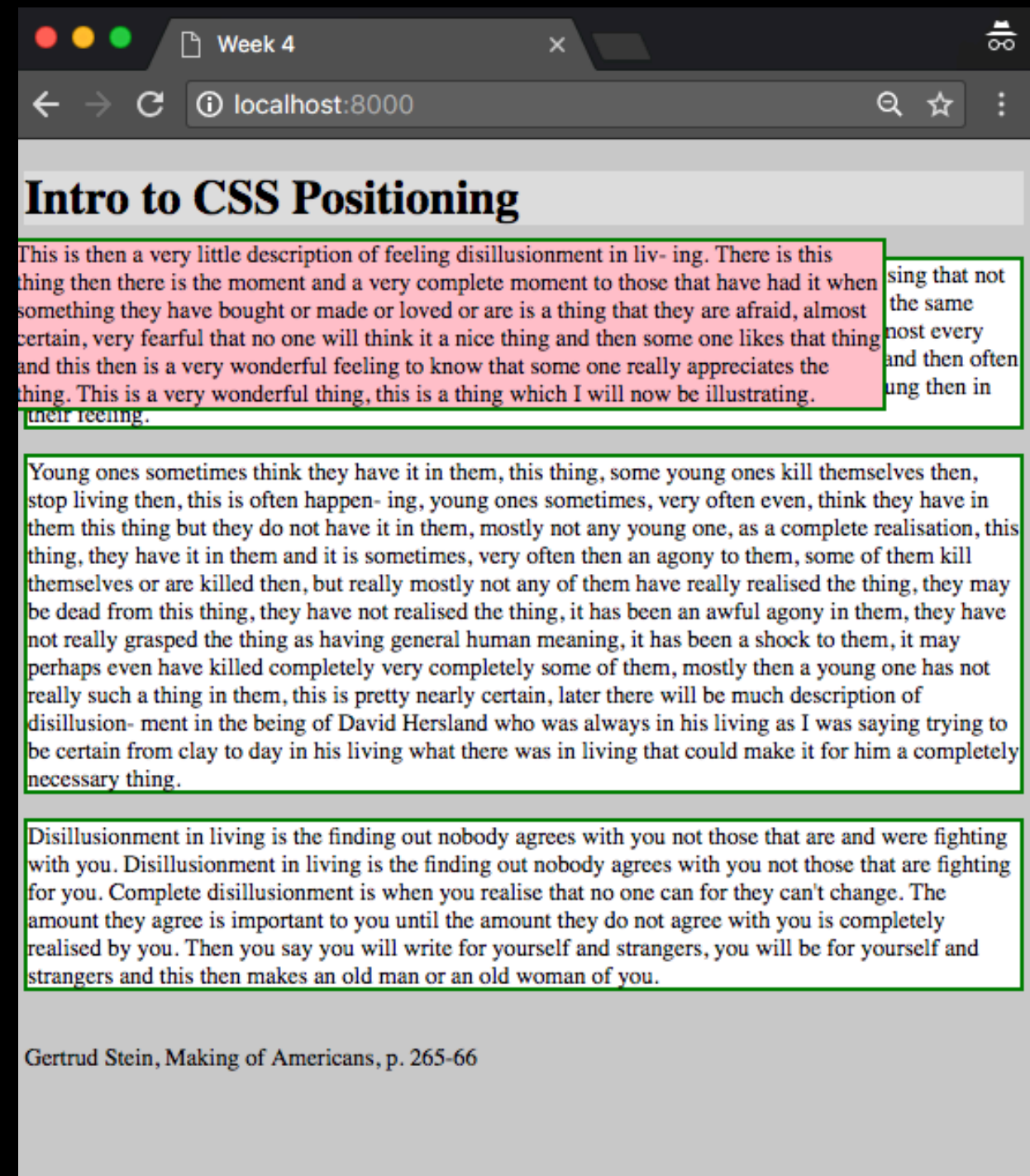
```
}
```

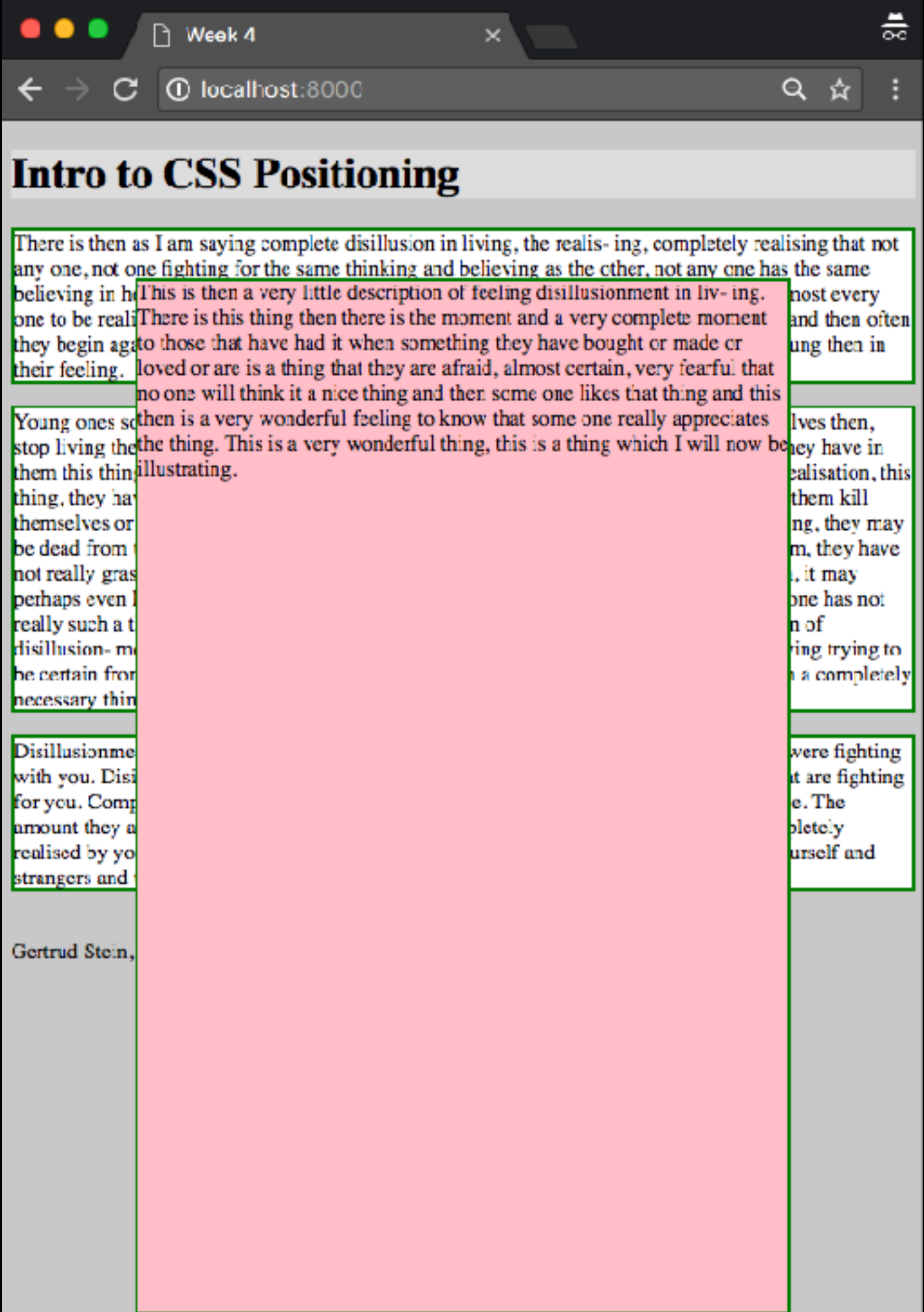


Notice that the position of the element has changed — this is because **top**, **bottom**, **left**, and **right** behave in a different way with absolute positioning.

Instead of specifying the **direction** the element should move in, they specify the **distance** the element should be from each containing element's sides.

So in this case, we are saying that the absolutely positioned element should sit **50px** from the **top** of the "containing element", and **100px** from the **right**.





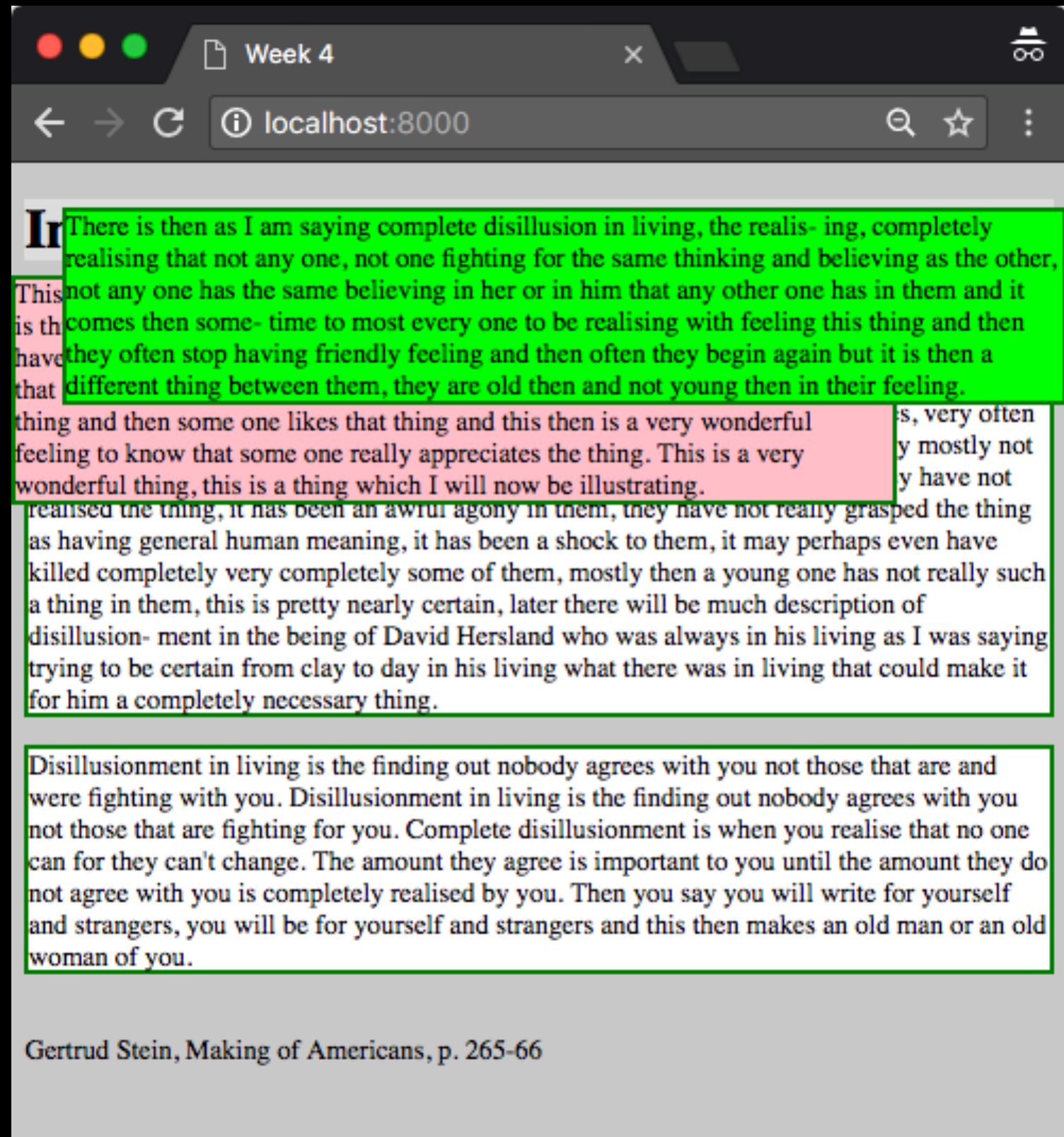
```
.thePosition {  
  
    background: pink;  
    position: absolute;  
    top: 100px;  
    bottom: 100px;  
    left: 100px;  
    right: 100px;  
  
}
```

Floating Elements

Floating an element allows you to take that element out of normal flow. The floated element becomes a block-level element around which other content can flow.

Web pages also have a z-axis: an imaginary line that runs from the surface of your screen, towards your face. **z-index** values affect where positioned elements sit on that axis; positive values move them higher up the **stack**, and negative values move them lower down the **stack**. By default, positioned elements all have a z-index of auto, which is effectively 0

```
{  
  background: lime;  
  position: absolute;  
  top: 10px;  
  left: 30px;  
  z-index: 1;  
}
```

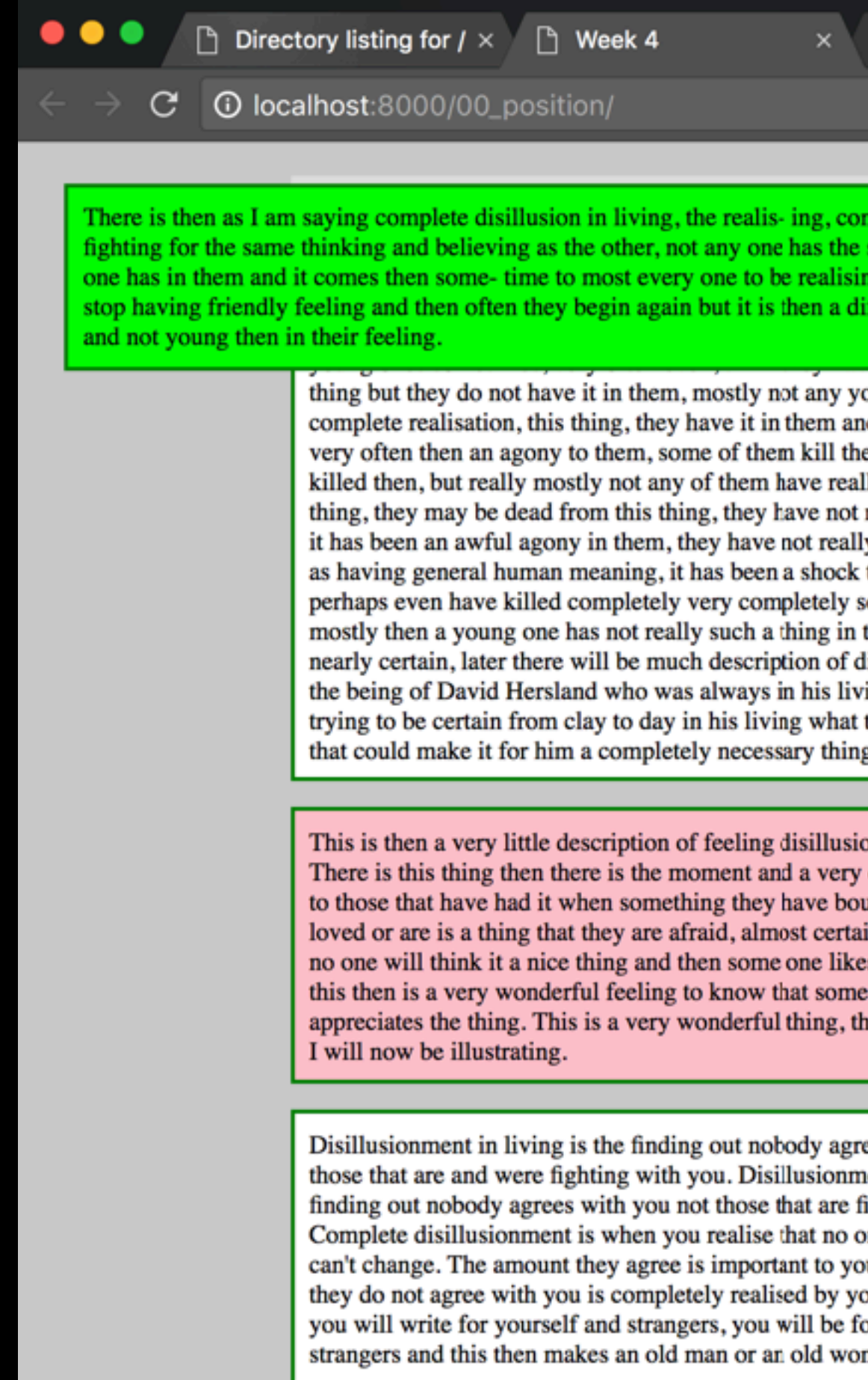


Overlapping Elements

When you use relative, fixed or absolute positioning, boxes can overlap. If they do the elements later in the HTML will sit on top of those that are earlier in the page.

When you move any element from normal flow, boxes can overlap. The **z-index property** allow you to control which box appears on top. It's value is a number, + the higher the number the closer that element is to the top.

Often referred to as **STACKING CONTEXT** + is similar to "bring to front" + "send to back features."



Fixed Positioning

This is a form of absolute positioning that positions the element in relation to the browser window, as opposed to the containing element.

Elements w/ fixed positioning do not affect the position of surrounding elements + they do not move when the user scrolls up and down the page.

```
.thePosition {
```

```
background: pink;
```

```
position: fixed;
```

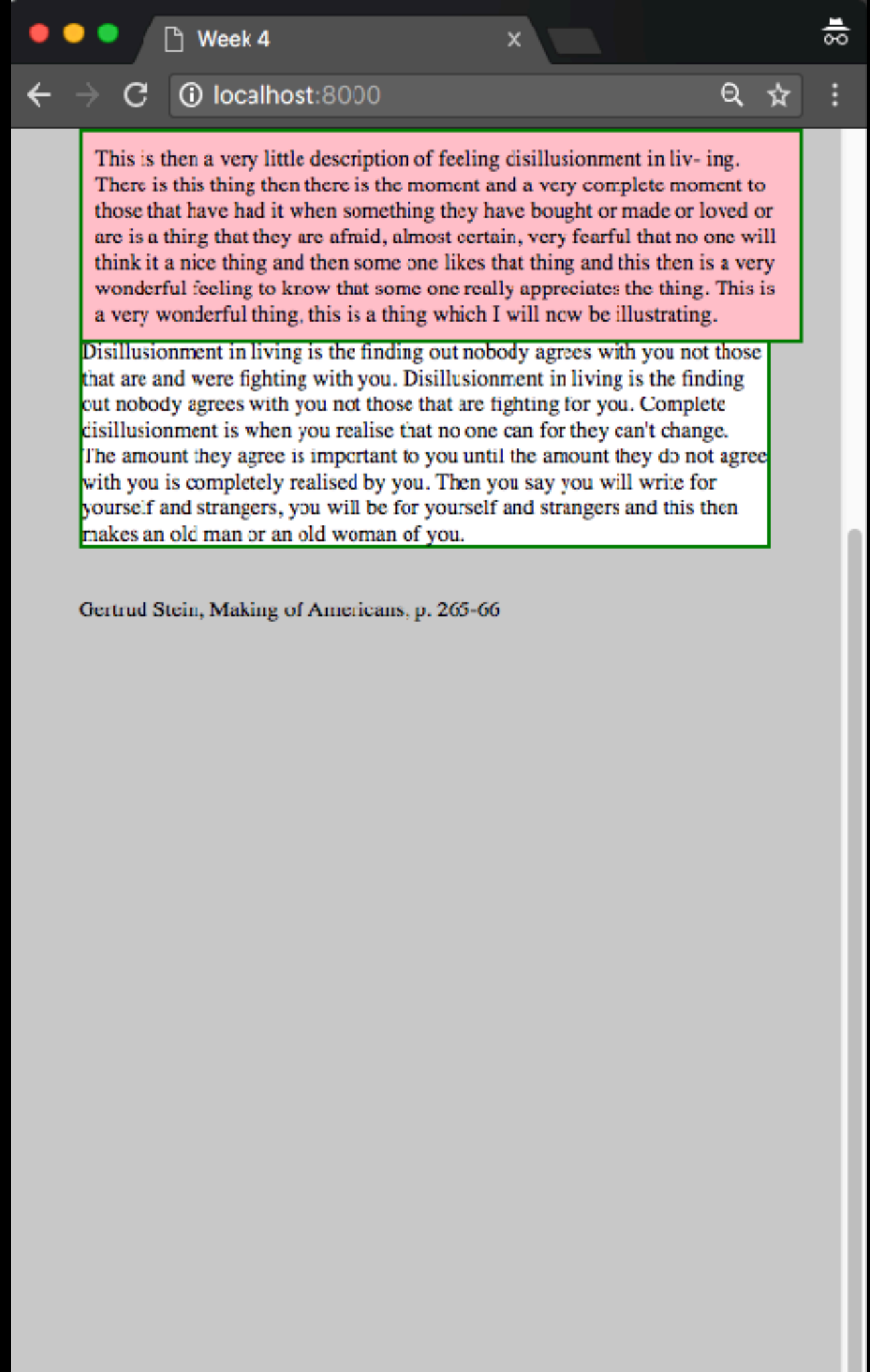
```
top: 0;
```

```
width: 50px;
```

```
margin: 0 auto;
```

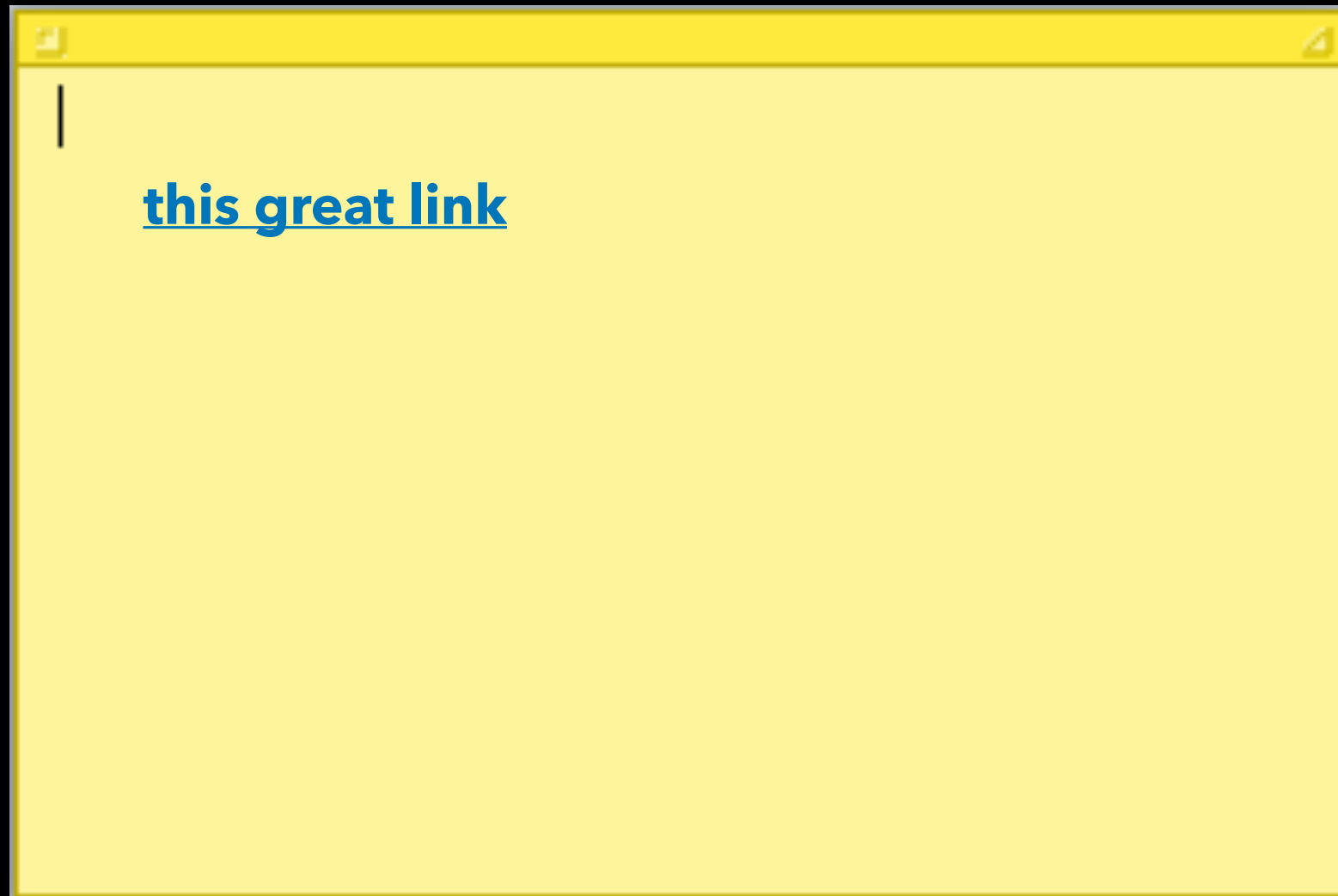
```
padding: 10px;
```

```
}
```



Gertrud Stein, Making of Americans, p. 265-66

Position Sticky



Font

Padding is the space btw the border + the content.

Some Properties:

`font-family`

`color`

`font-size`

`line-height`

`text-align`

Font

Padding is the space btw the border + the content.

text-decoration

underline, strike thru or none (eg to unset underline on hyperlinks)

text-transform

change font **case** (eg uppercase, lower, capitalize, none)

font-style

set to italic or normal

font-weight

set to bold or normal

letter-spacing

controls the space btw letters

Font stack

It's important to understand that the browser will only display font if it's installed on user's computer.

Font stack - a collection of more than one typeface in an order of preference to be displayed in the browser if some of the typefaces are not found.

```
{  
  font-family: Georgia, Courier, serif;  
}
```


Setting the font-family

- **font-family** property sets the font in your CSS
- Presented as a hierarchy of choices (1st choice, 2nd choice, 3rd choice) so it's good to have a fallback for older browsers that can't render

```
body {
```

```
    font-family: Georgia, Courier, serif;
```

```
}
```

```
h1, h2, h3 {
```

```
    font-family: Arial, Verdana, sans-serif;
```

```
}
```

Google Font API

Custom web fonts: Google Fonts

Add link in **<head>** of HTML

```
<link href="https://fonts.googleapis.com/css?
family=Roboto" rel="stylesheet">
```

Use with font-family property in CSS

```
font-family: 'Roboto', sans-serif;
```