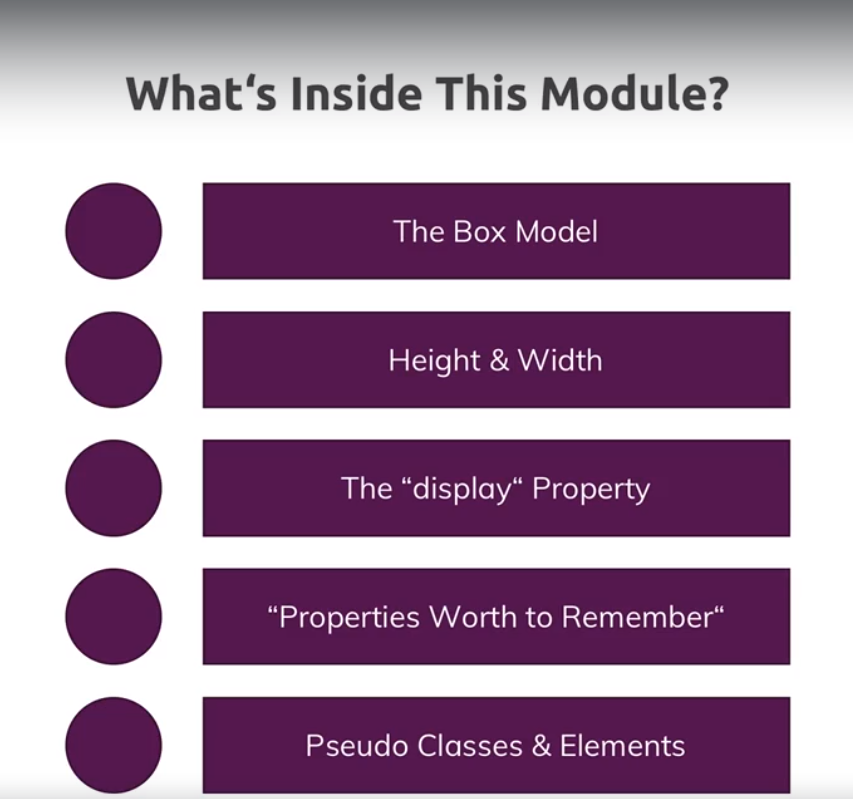
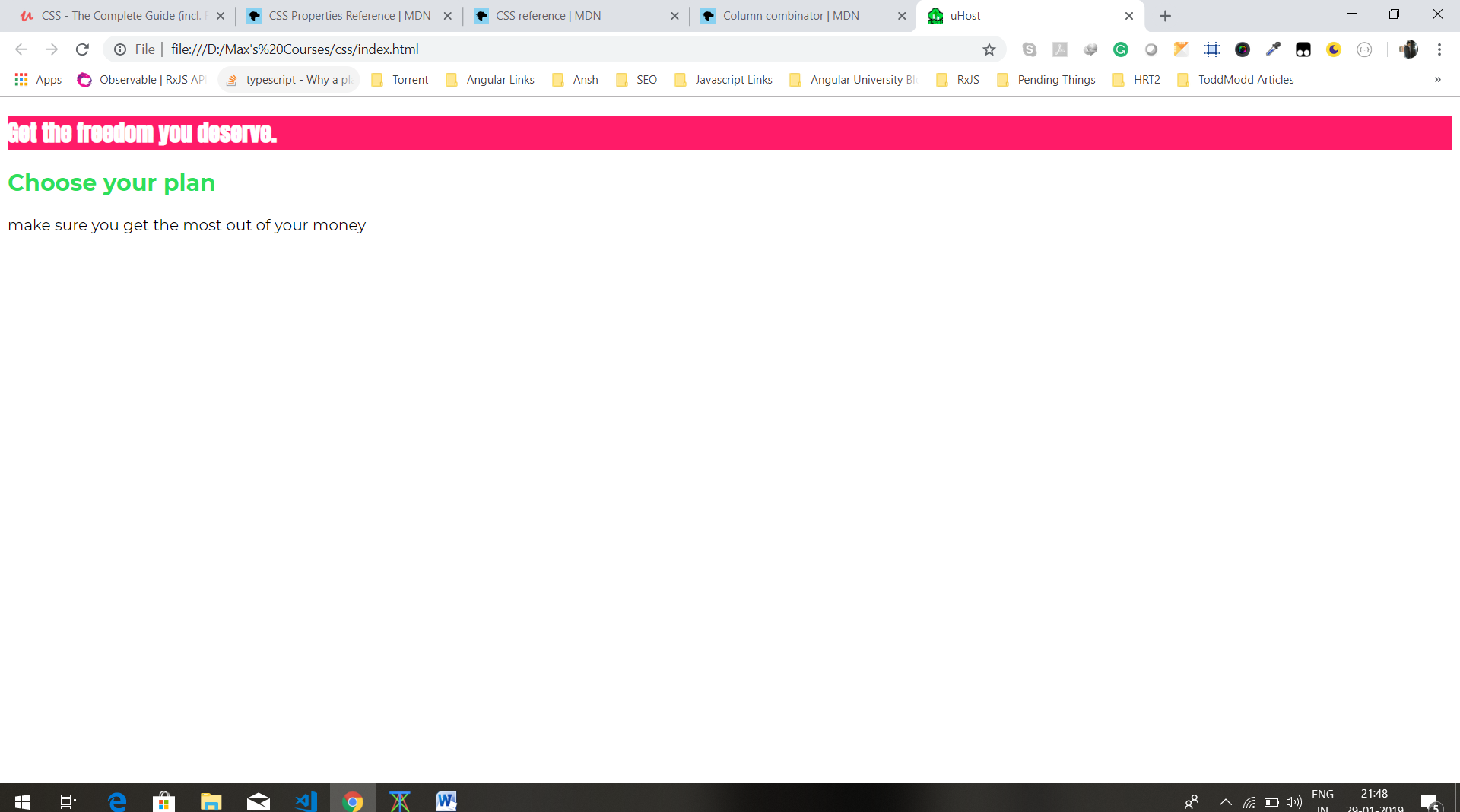
24)Module introduction



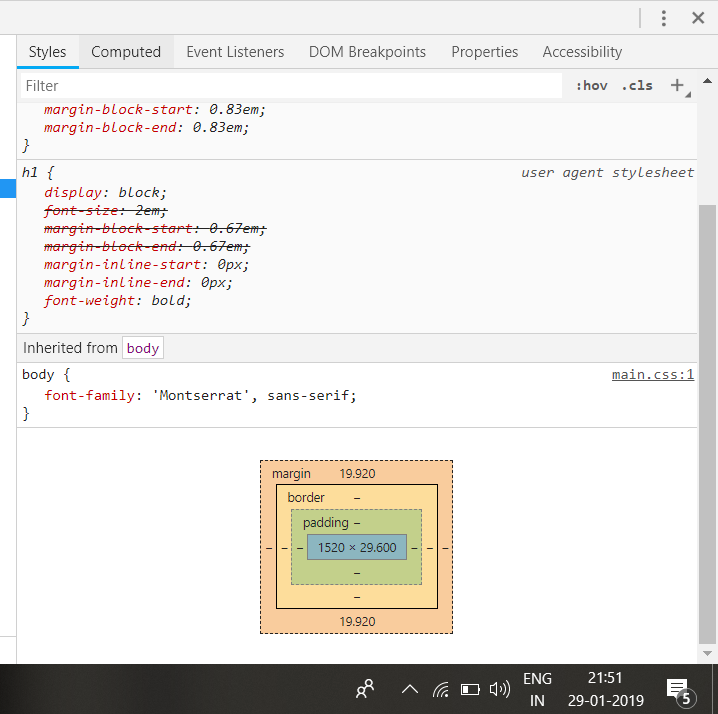
25)Introduction to the CSS Box Model

Our web page is this-



We want to control, how red block is designed. We can do that with css box model. What is this?

Every element in html is interpreted as box by css and you can see that box in chrome dev tools if you inspect that element. Select any element, in chrome dev tools you can see this box-



This is how css things about any element, evety elemnt has a content that is blue area in box. Here we have 4 parts- content, padding, border and margin. So every element is interpret like this box by css.

Now there are 2 types of elements then block level and inline(something we will cove in this module) where some of these things differ a bit. But for now lets keep in mind that every elements has these layers.

26)Understanding the box model

We saw box model in theory, now lets see it in action.in css-

body {

font-family: 'Montserrat', sans-serif;

}

#product-overview {

background: #ff1b68;

padding: 20px;

}

.section-title {

color: #2ddf5c;

}

#product-overview h1 {

color: white;

font-family: 'Anton', sans-serif;

}

Here we added padding to id selector(#prodcut overview). Here is our html-

<body>

<main>

<section id="product-overview">

<div class="sumit">

<h1>Get the freedom you deserve.</h1>

</div>

</section>

<section id="plans">

<h1 class="section-title">Choose your plan</h1>

<p>make sure you get the most out of your money</p>

</section>

</main>

</body>

Now as you can see that we have h1 tag inside section with id product overview. Now h1 tag has some margins defined by browser. now talking about section tag now, content for this tag will be content of h1 tag plus margin of h1 tag. So its just that if you add padding(we added padding to section) then content of element (section)is to be considered content plus any ,margins it might have(here h1 is content of section. but h1 has margin, so content for section will be conent of h1+ margin o h1). Because paddings and margins should’t overlap, hence padding is added after the margins of child elements.like here padding to section is added after margin of h1.

Then we give border and margin to section

#product-overview {

background: #ff1b68;

padding: 20px;

border: 5px black solid;

margin: 20px;

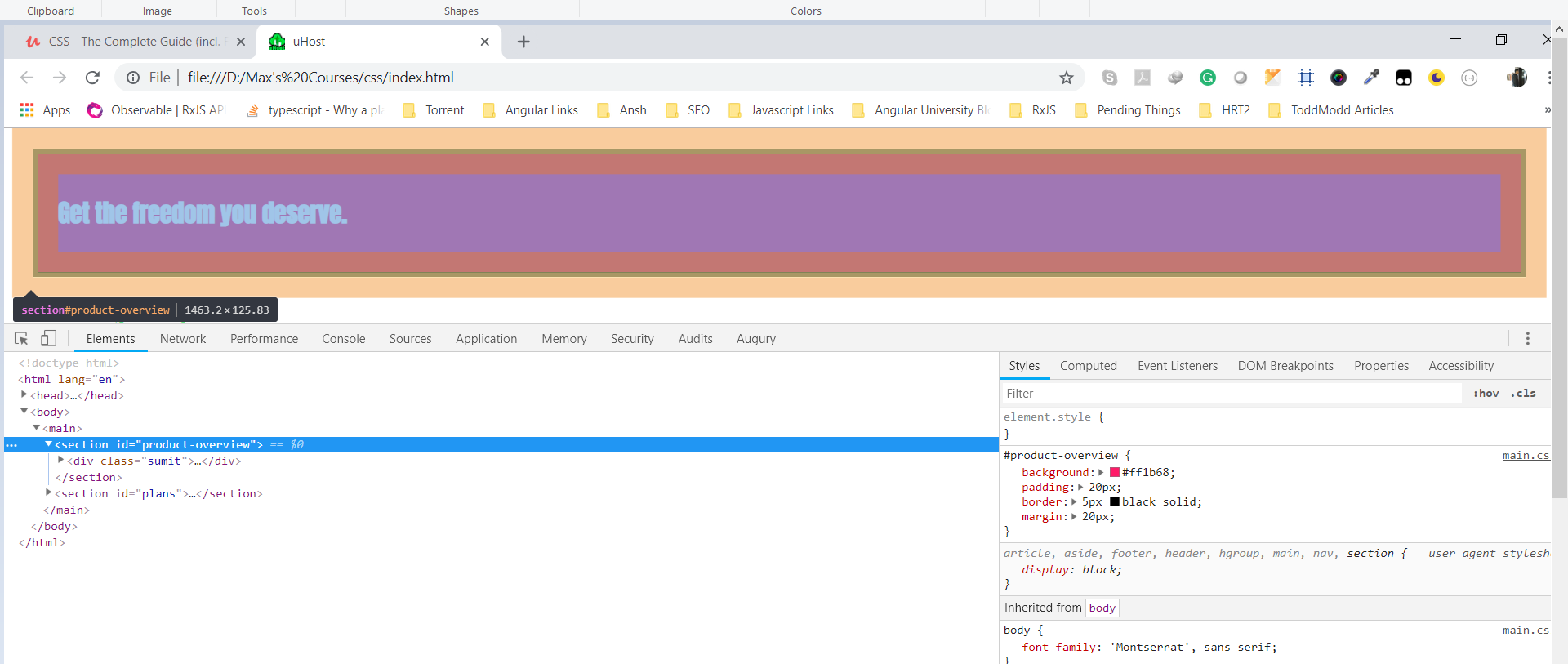
}

Now there are couple of special things you got to know about box model and default styles of web page. Lets dive into these things step by step.

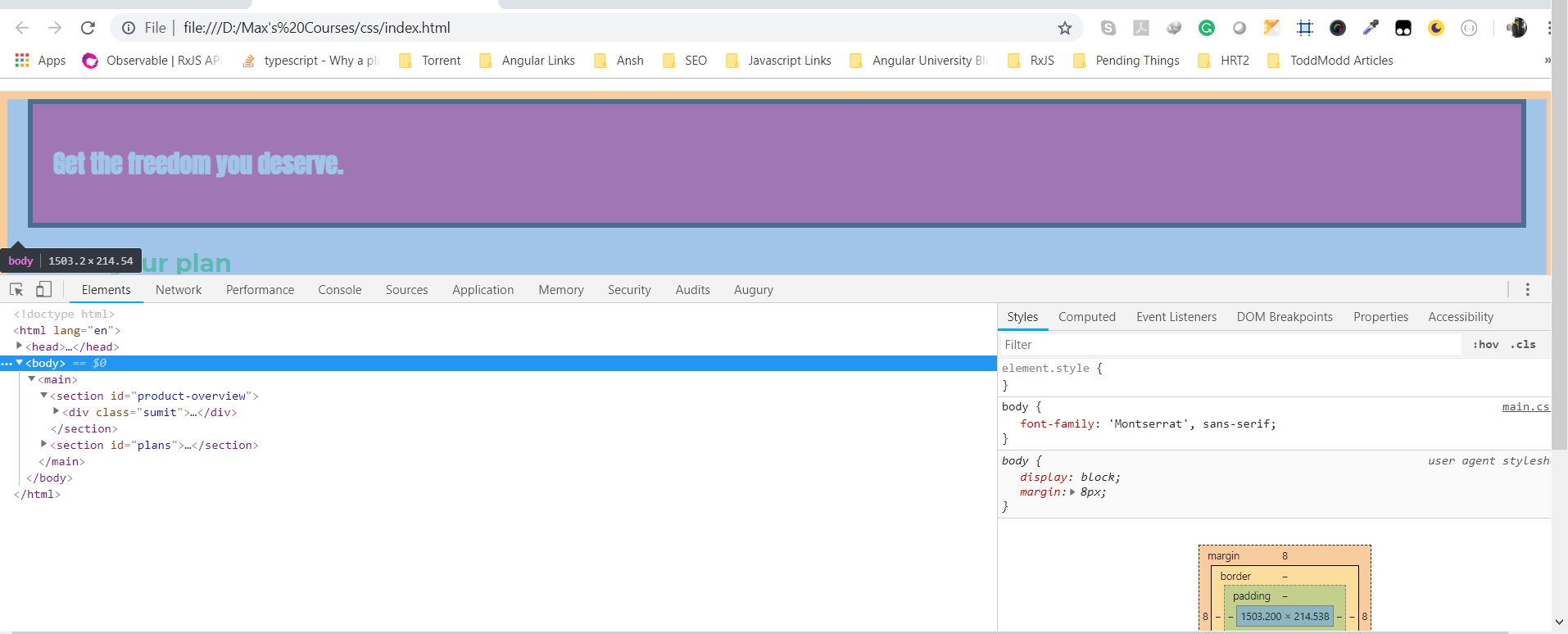
27)Understanding margin collapsing and removing default margins

In last lecture I mentioned there will be couple of special things, lets step into them step by step.

First special thing can be seen if you inspect the section-



we can see that there is white space to left and right of section with id product-overview. This white space is after orange margin. This is coming from body actually. If you hover over body then you can see that body also has some default margin.



Here you can see that body has default margin of 8 px. It is coming from browser defaults.

To fix this we can set margin equal to 0 for body like this-

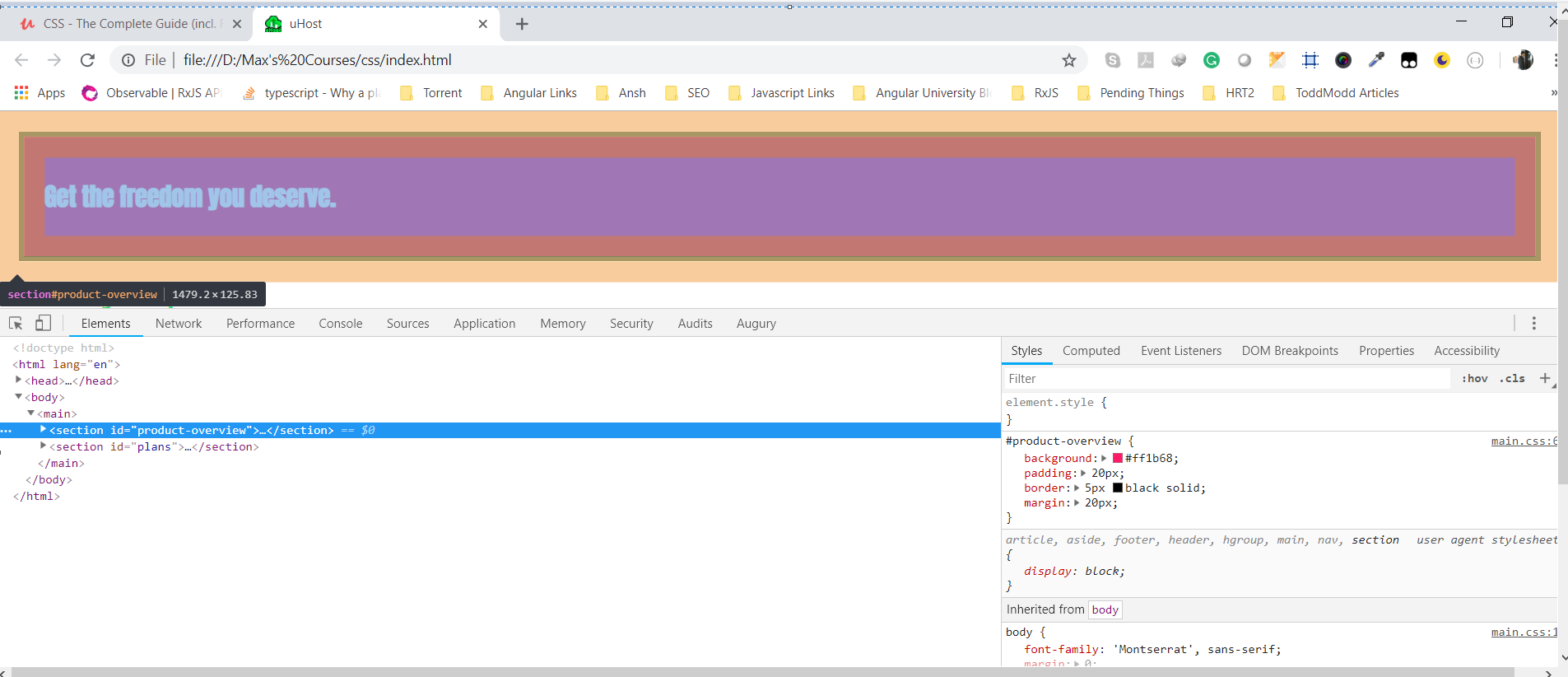
body {

font-family: 'Montserrat', sans-serif;

margin: 0;

}

Now if we see, margin (indicated by orange color) of section directly connects with corners of browser.



So first thing was that body has margin by default. Now our html is- (refer to code - marginCollapsingDemo)

<main>

<section id="product-overview">

<div class="sumit">

<h1>Get the freedom you deserve.</h1>

</div>

</section>

<section id="plans">

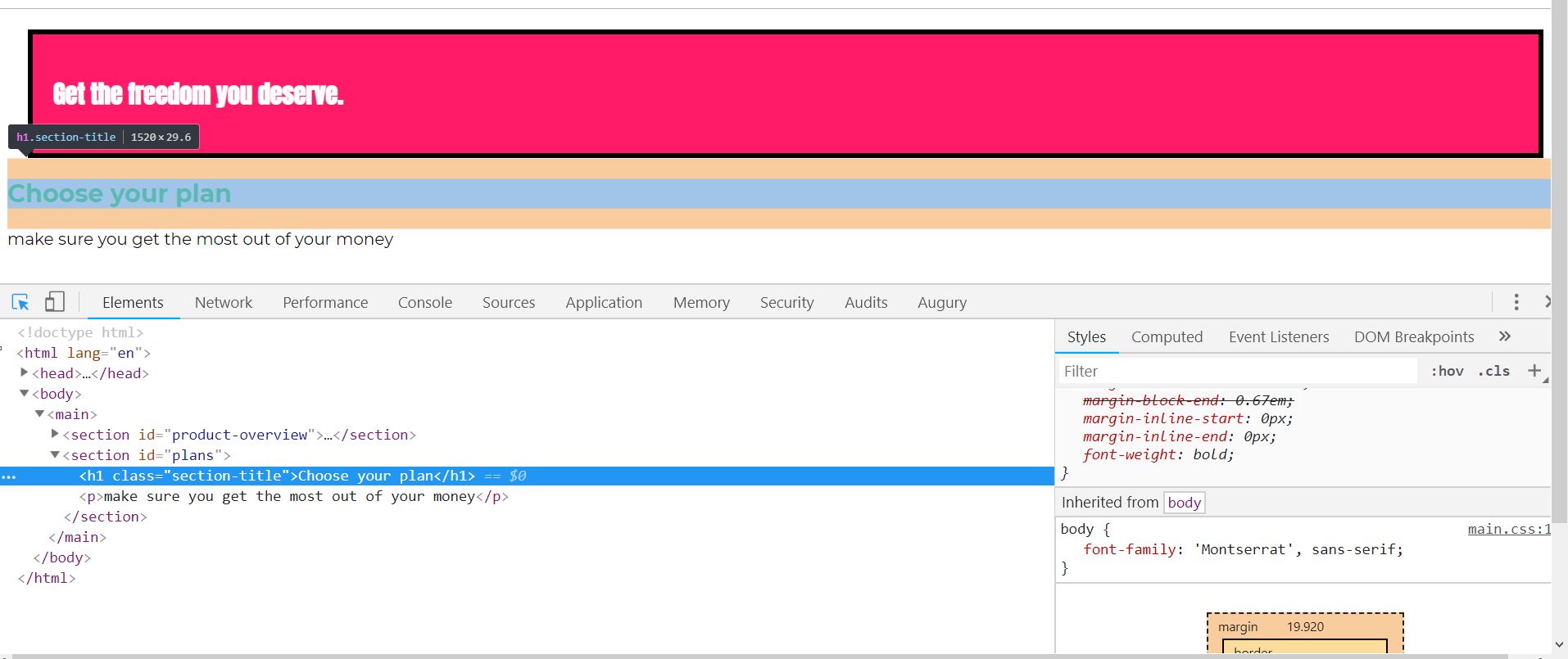
<h1 class="section-title">Choose your plan</h1>

<p>make sure you get the most out of your money</p>

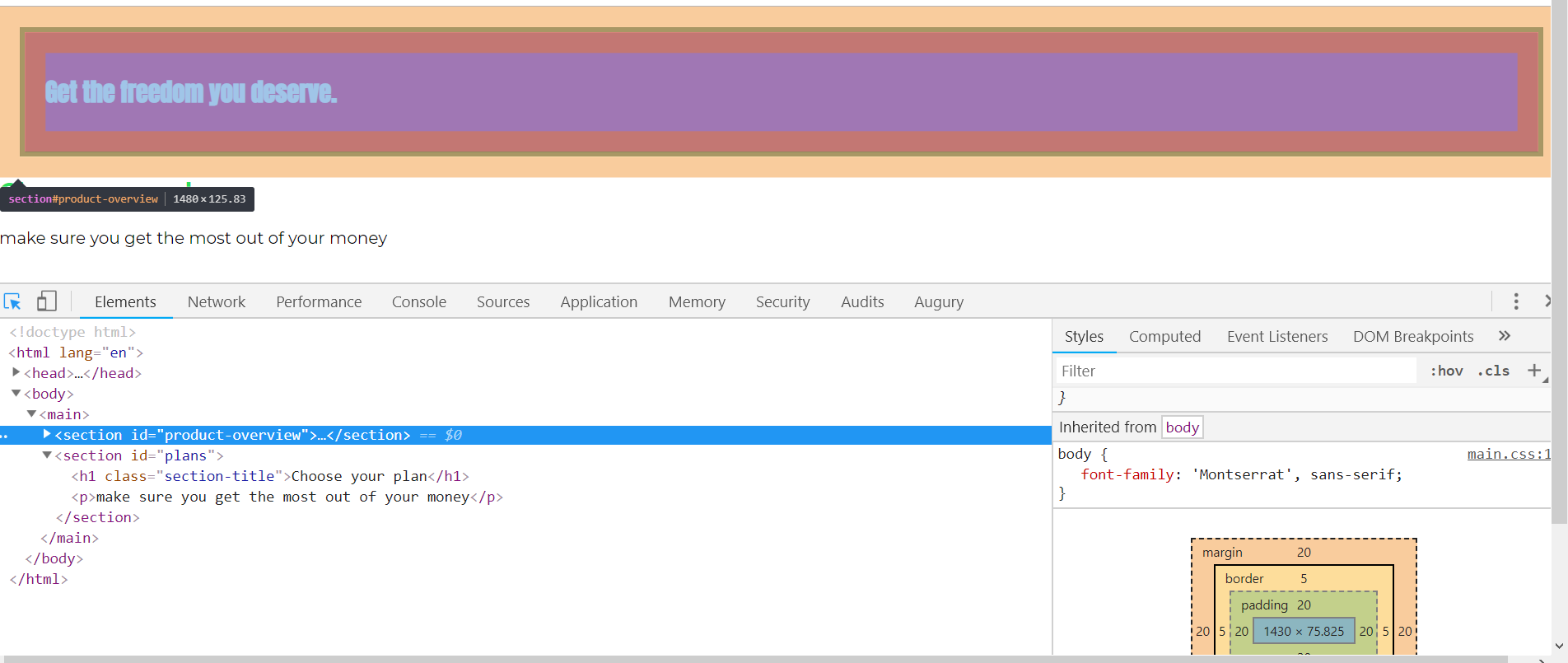
</section>

</main>

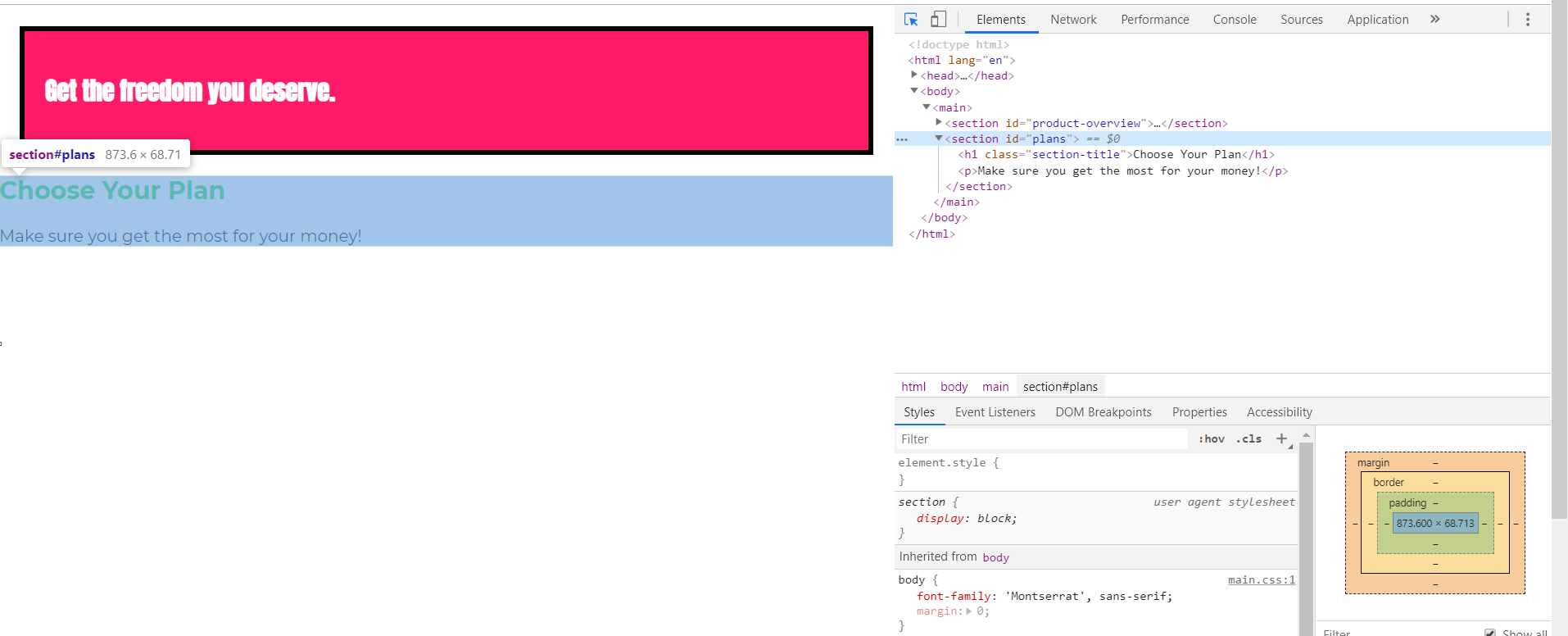
If you inspect the h1 tag in plans section, you will see that margin of this h1 and margin of section with id product-overview, they overlap. This is h1-



This is our first section-



Note that section with id plans do not have any margin-



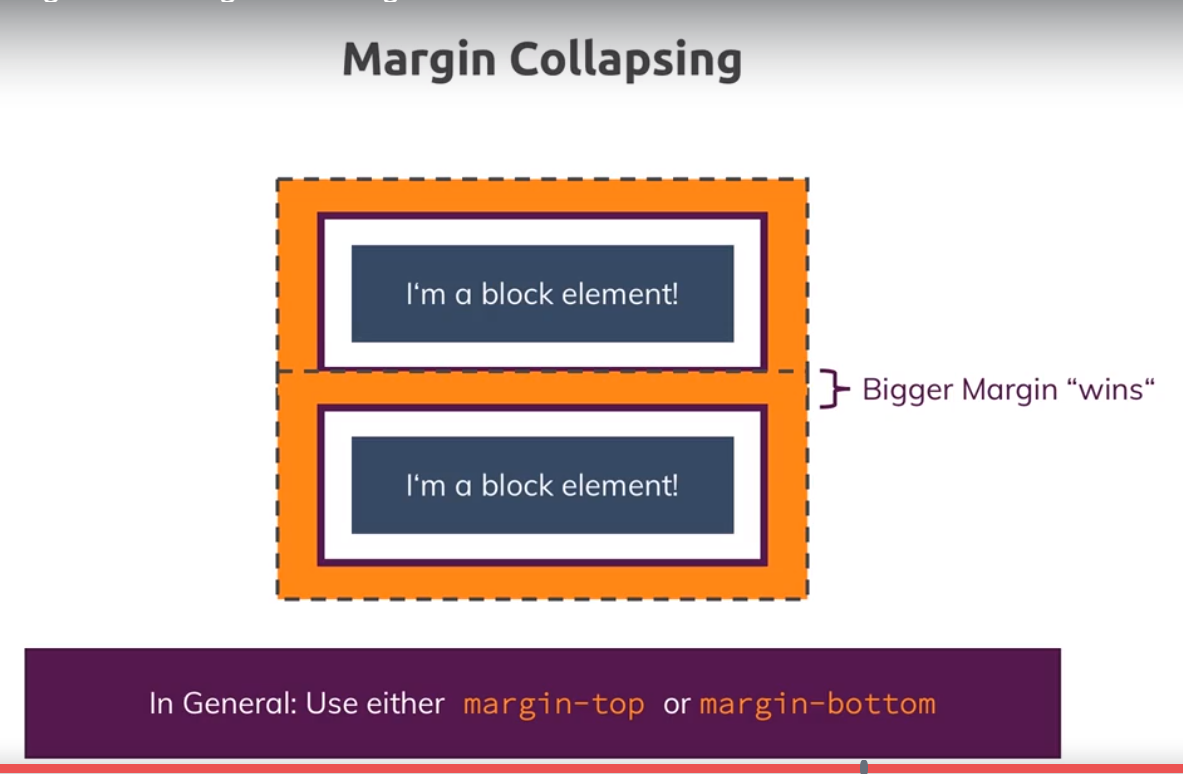
But thing to be noticed here is that margin of h1 goes outside section which contains h1.

This behaviour is called margin collapsing. It simply means following-

If you got 2 elements, block elements with it’s box model(margin is orange part here)



If you got 2 elements next to each other, then margin between them actually collapses into one margin, the bigger margin wins.



This is not abug , this is on purpose. This is ensured by css so that you dnt get too big distances between the elements. You can argue that, leave that to me, I will handle it. But this is not how css works. So this is something to be aware of. To work around it, it is good practice to use margin-top or margin-bottom unless you dnt worry about this collapsing occurring which of course also can be the case.

So in our case we got collapsing but it’s not really a problem because we have got a fine distance between elements.

I just told you this, so that you know what is going on here.

My observation-

**Here thing is there is also margin collapsing occurring between second section tag and its childs. Top margin of h1 collapses with top margin of section and bottom margin of section collapses with bottom margin of p tag. But if we inspect 2nd section tag , we cnt see margin, because it does not have margin and dev tools don’t show you collapsed margin. It will also show you margin of individual elements. But spacing between elements is decided by collapsed margin. So this is happening-**

**Top margin of h1 combines with top margin of 2nd section. then top margin of second section collapses with bottom margin of first section.**

This thing is explained in next lectures.

See this question when you are css pro-

<https://www.udemy.com/css-the-complete-guide-incl-flexbox-grid-sass/learn/lecture/9466098#questions/3959636>

28)Deep dive on margin collapsing

When working with margins, you can get unexpected results.

* Why are two adjacent elements sharing one margin even though each element has its own one?
* Why does a parent element (e.g. <section>  as in the videos) suddenly take on the margin of the child element (e.g. <h1> )?

It's always related to margin collapsing. You can dive deeply into it with the help of the following awesome article: <https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Box_Model/Mastering_margin_collapsing>

Read above link, here you can see if we give height to pafent then bottm margin collapsing does not occur.

There, three base cases are described:

1. Adjacent siblings which both have margins
2. A parent which holds one or more child elements where the first and/ or last (or the only) child has margins
3. An element without content, padding, border and height

Let's explore these cases:

**1. Adjacent Siblings**

In this case, the first element might have a margin of 10px  (on all sides let's say) and the second one has 5px  (or 20px  - the values don't matter).

CSS will collapse the margins and only add the bigger one between the elements. So if we got margins of 10px  and 5px , a 10px  margin would be added between the elements?

**2. A parent with children that have a margin**

To be precise, the first and/ or last or the only child has to have margins (top and/ or bottom). In that case, the parent elements margin will collapse with the child element(s)' margins. Again, the bigger margin wins and will be applied to the parent element.

If the parent element has padding, inline content (other than the child elements) or a border, this behavior should not occur, the child margin will instead be added to the content of the wrapping parent element.

**3. An empty element with margins**

This case probably doesn't occur that often but if you got an element with no content, no padding, no border and no height, then the top and bottom margin will be merged into one single margin. Again, the bigger one wins.

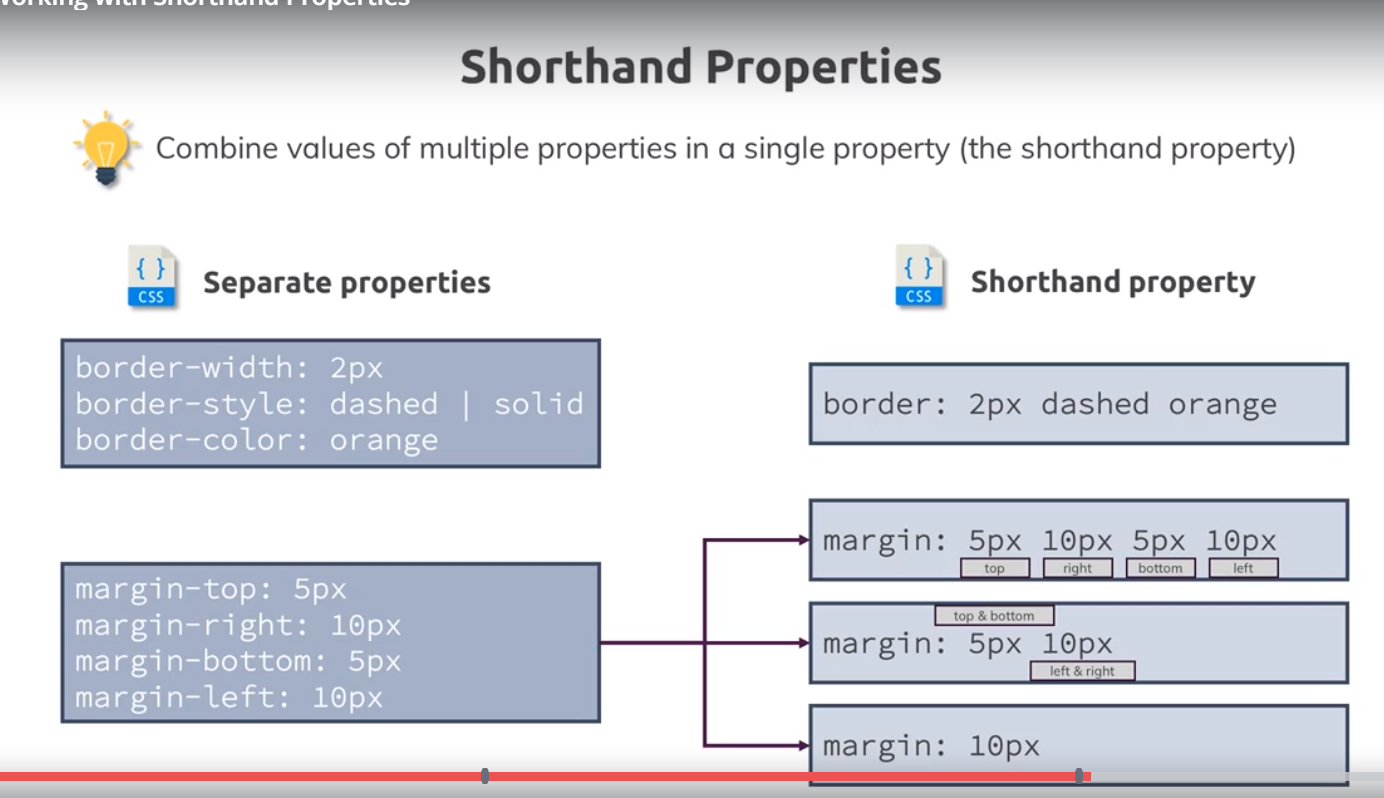
Here if you go to official docs on mozzila remember the thing, if we give height to element the bottom margin of that element does’nt collapse with top margin of last child, top margin collapses with first child.

Go through official docs.

29)Theory Time-Working with shorthand properties

Shortand proeprties combine values of multiple other properties in a single property(so called shorthand property).

The order of properties don’t matter here as long as the properties don’t use the same type of value which is’nt the case here but for other shorthands it will be the case. I will highlight this and tell you how to circumvent this once it is the case. So for border we can change the order of properties in which ever way we want.



We use lot of shorthands in css, I will mention them whenever we can use them. Again we already did it for background(we used background: red). I will dive dive deeper into what background covers or what other values can be part of background later in this course, when we also add background images.

Now lets say we used shorthand for border in one selectors, now in some other selector you want to overwrite lets say only border color, then you just define property that you want to overwrite, it will take rest values from first selector, i.e border-style and border-width will be taken from first selector.

30)Applying shorthands in practice

Now we are already using shorthands . our css file is-

body {

font-family: 'Montserrat', sans-serif;

margin: 0;

}

#product-overview {

background: #ff1b68;

padding: 20px;

border: 5px black solid;

margin: 20px;

}

.section-title {

color: #2ddf5c;

}

#product-overview h1 {

color: white;

font-family: 'Anton', sans-serif;

}

Here border, margin, padding, background are used in shorthand form.

In chrme dev tools we can expand these shorthand proeprties to see all longer hand proeprties that are combined. Like this-



We will go through background property in detail.we can use background image, which we can then position by various sub properties. In case of border we can style each side of border differently.

31)Diving into height and width properties

My observation of margin collapsing-

Block elements by default take 100% width(100% width of parent). So if we dnt specify any width then they take 100% width of parent.but we can specify width like 50%.

Html is-

<body>

<main>

<section id="product-overview">

<h1>Get the freedom you deserve.</h1>

</section>

<section id="plans">

<h1 class="section-title">Choose your plan</h1>

<p>make sure you get the most out of your money</p>

</section>

</main>

</body>

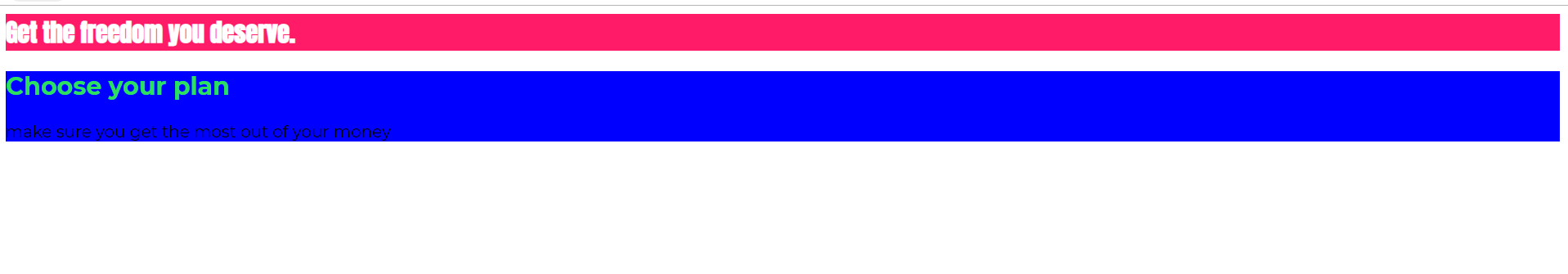
We remove all properties from id selector. so now id selector is-

#product-overview {

background: #ff1b68;

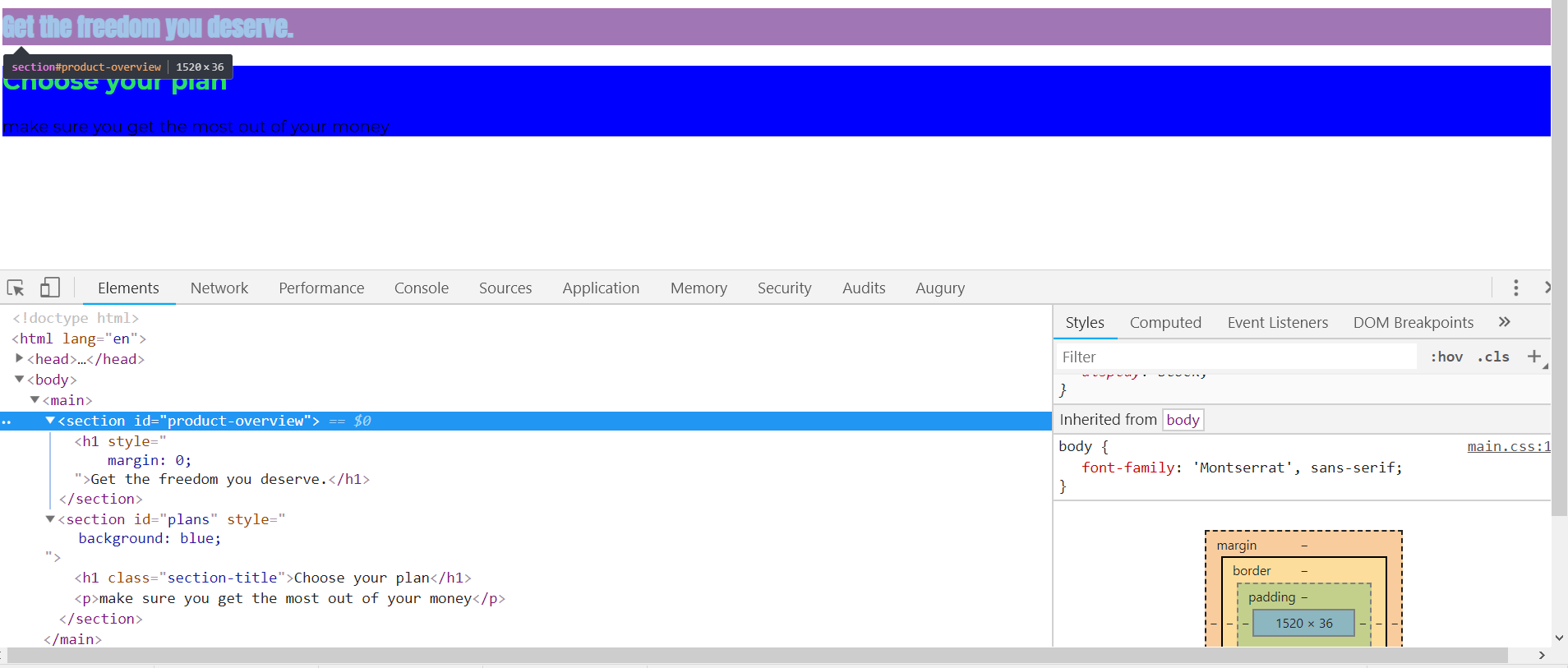
}

Now if u see 2 sections there is space between them-



Here we gave blue color to second section, just for highlighting purpose., i.e to show space between 2 sections.

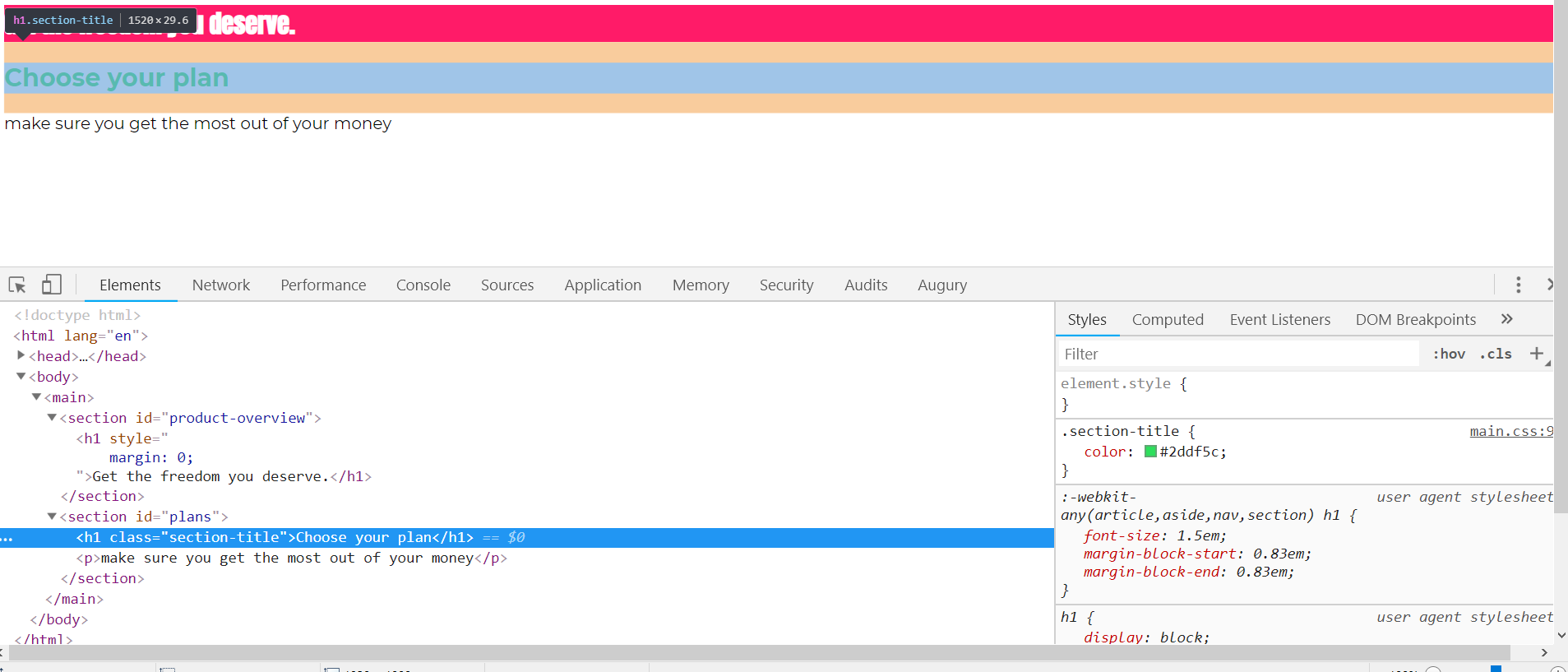
but from where this space is coming? Both sections, do not have any margins. In dev tools lets inspect section tags. In box model see no margin.



Same goes for second margin, i.e no margin.

But h1 tag in both sections have margin and because of margin collapsing, it gives margin to sections. But in chrome dev tools if we inspect section we wnt see any margin.this means dev tools wnt show collapsed margin. They will always show individual margin.

Now lets disable margin in last h1 of section one. Still we have space. This space is coming from margin of first h1,like this-



So this is what was happening-bottom margins of last h1 of section 1 and secton1 collapse.

Same happens with top-margin of section2. Then both collapsed margin again collpases to have single margin. That is why disabling the margin of last h1 of first section had no effect on gaps between section 1 and 2.

Lecture code-

Now we can set width to 100% of section, it means take full width. If we save and reload the page change we dnt see any change in webpage because that was its default behaviour because sections, like div and h1 are block level elements. block level elements unlike inline elements(like a tag) take full avalaible width by default. So lets set width to 50% like this-

#product-overview {

background: #ff1b68;

width: 50%;

}

Now width is 50% of **surrounding container**. here surrounding container is 100% of page width, so we can take that our width now is 50% of page width.(this is because surrounding element is main element, which being block element also takes full width). If surrounding container has 50% of width of web page, then section will get 50% of total width of main element.

We can either use % or we can also use px.

That was width, we can also set height. It is tricky. If you set height to 100%, it is not going to occupy whole page.

#product-overview {

background: #ff1b68;

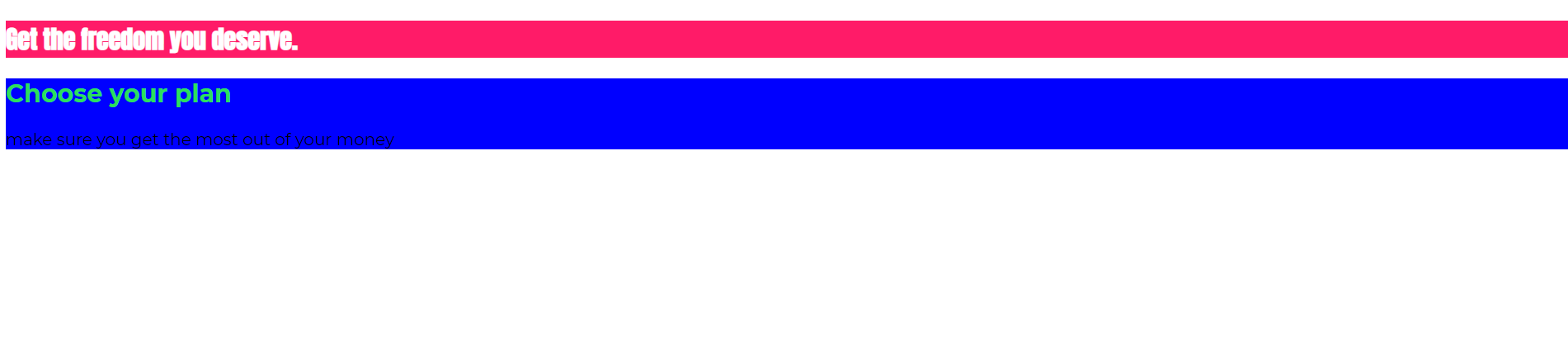
width: 50%;

height: 100%;

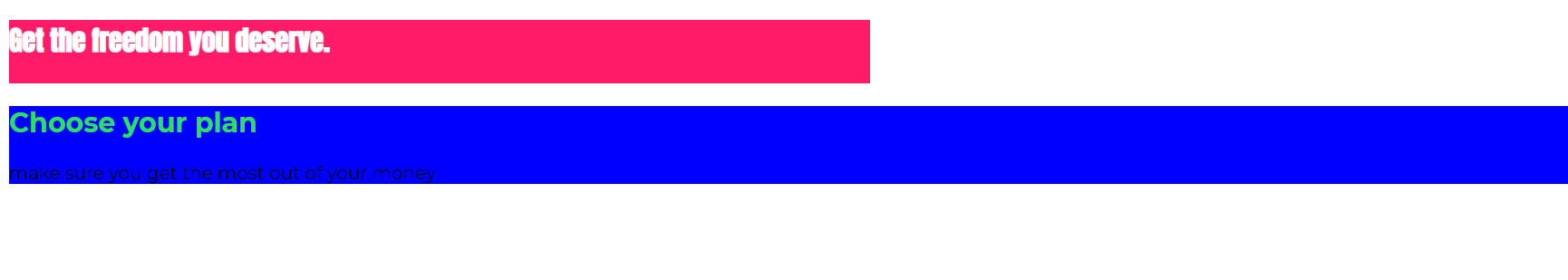
}

If we save and reload the page, we will see that section only get little bigger, only thing it does now is it also includes the height of margin of h1 element.

Before-

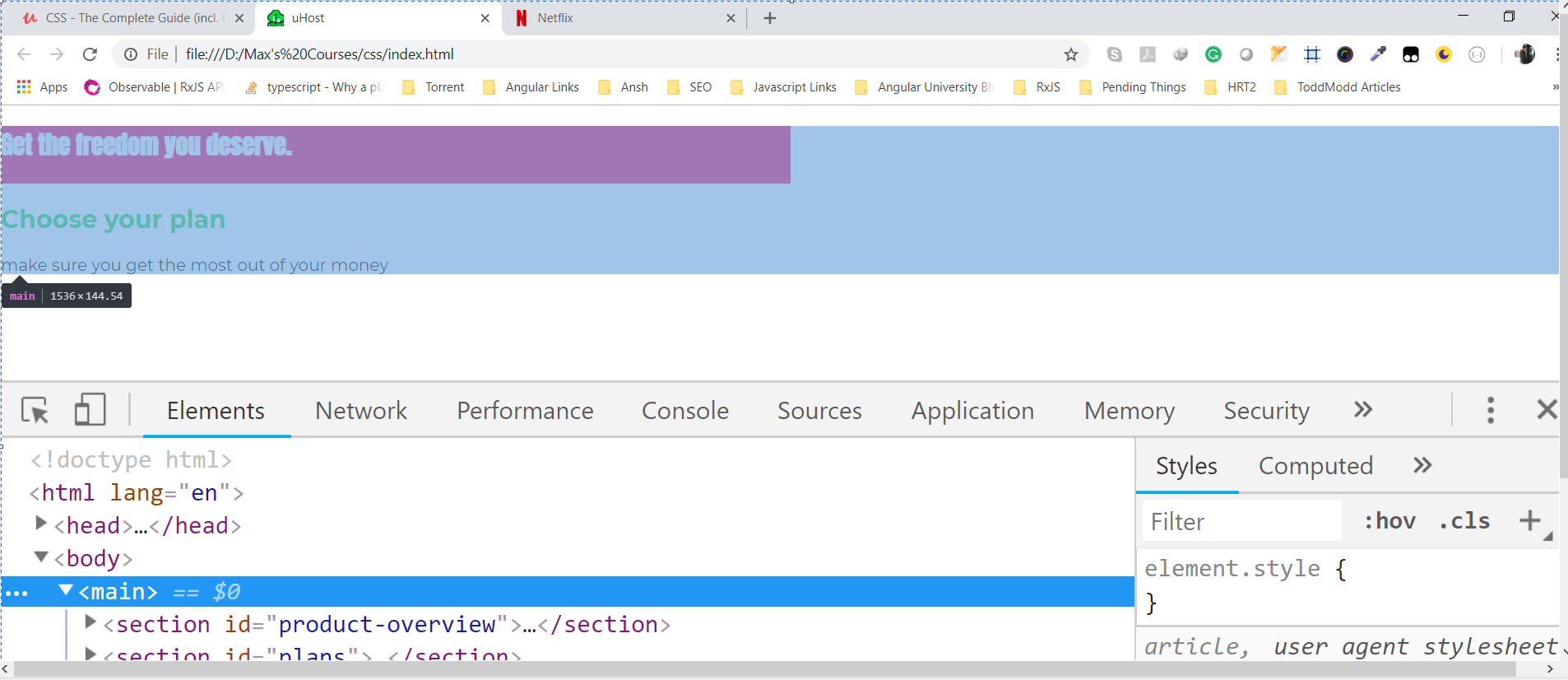


After setting width and height-

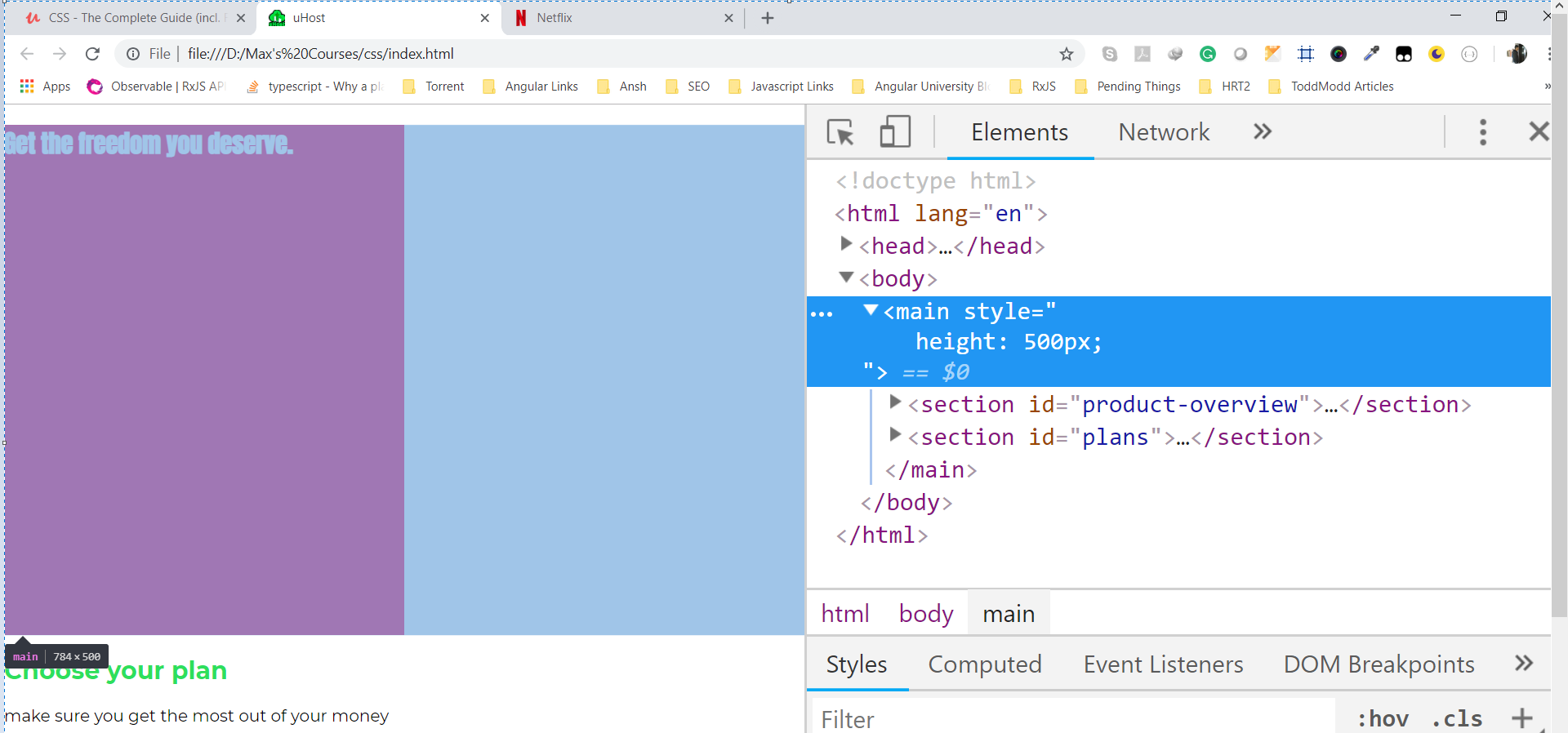


It got bigger from below is because, now we given height to section1, so now margin collapsing is not happening at bottom –margin.

Still height is not 100% of page.the reason for that is 100% refers to the avalaible height given by parent container. If we select the main element(which is parent of section) in chrome dev tools, we can see that selected area is much bigger than area of having red background(area of first section)-



But that area (height of main element) is calculated dynamically by content it holds. So it is only as big as its content requires it to be. Now that of course creates a kind of infinite loop, if we say part of content should be 100 percent of the size of main area and then main area says I am only as big as I need to be ,then 100 percent basically has no effect. If you wanted it to have an effect you would need to change the height of main area. In other words we have to change the height of main, lets set in to 500px(change it temporary in chrome dev tools by giving inline style), now red background occupies whole area. Now lets inspect main tag in chrome dev tools.



<main>

<section id="product-overview">

<div class="sumit">

<h1>Get the freedom you deserve.</h1>

</div>

</section>

<section id="plans">

<h1 class="section-title">Choose your plan</h1>

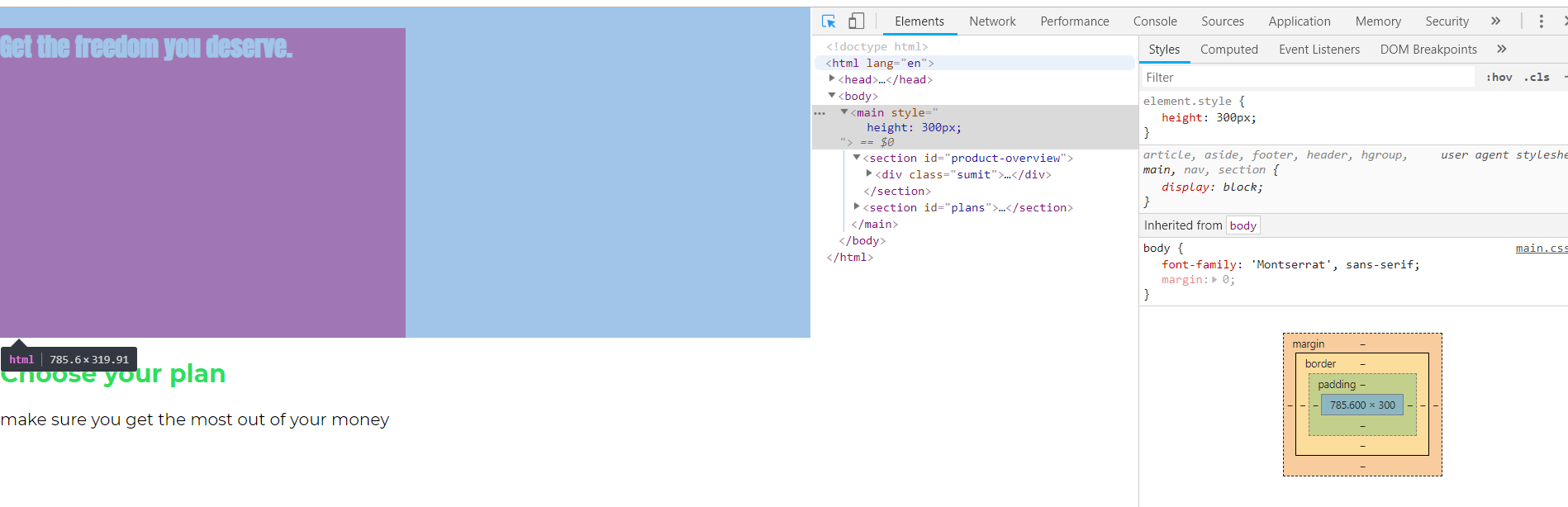
<p>make sure you get the most out of your money</p>

</section>

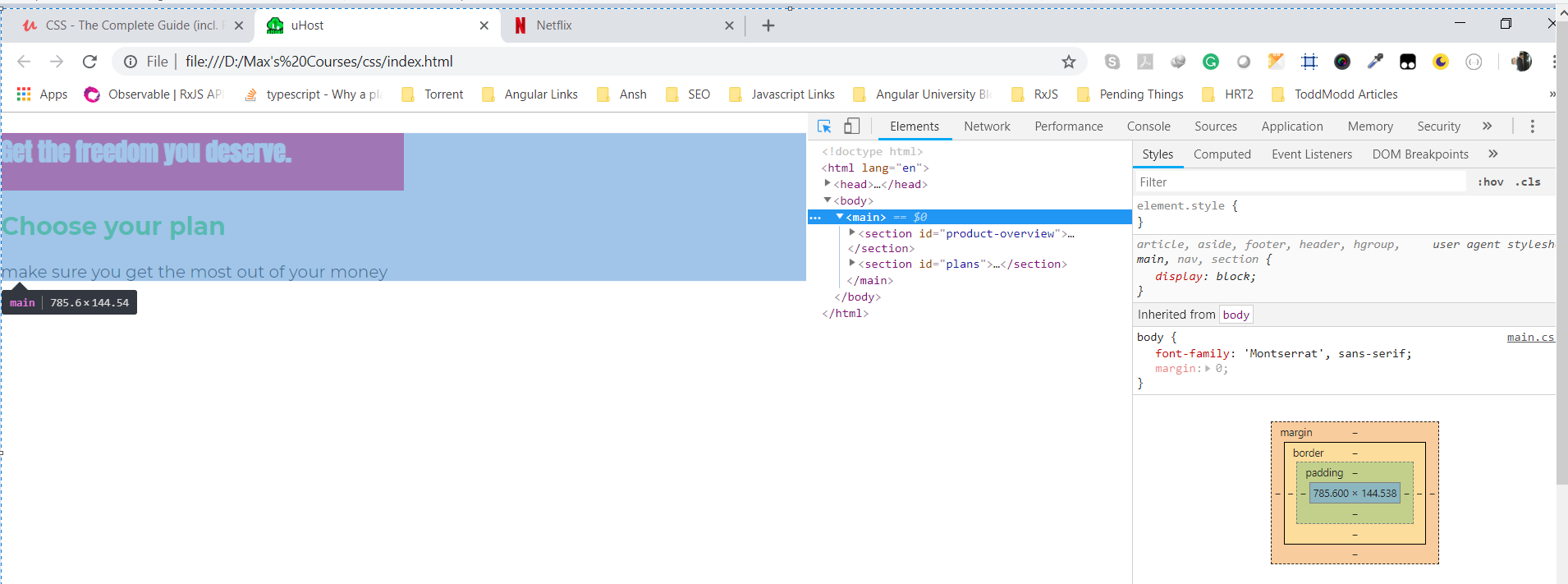
</main>

If we inspect the section, we can see that it has height equal to 500px.whoch is exactly the height we have to main area. Now 2nd sections is outside the main tag as we can see in chrome dev tools.

Like this-



Before giving height of 500px to main, it was like this(here also we had height of section equal to 100%)-

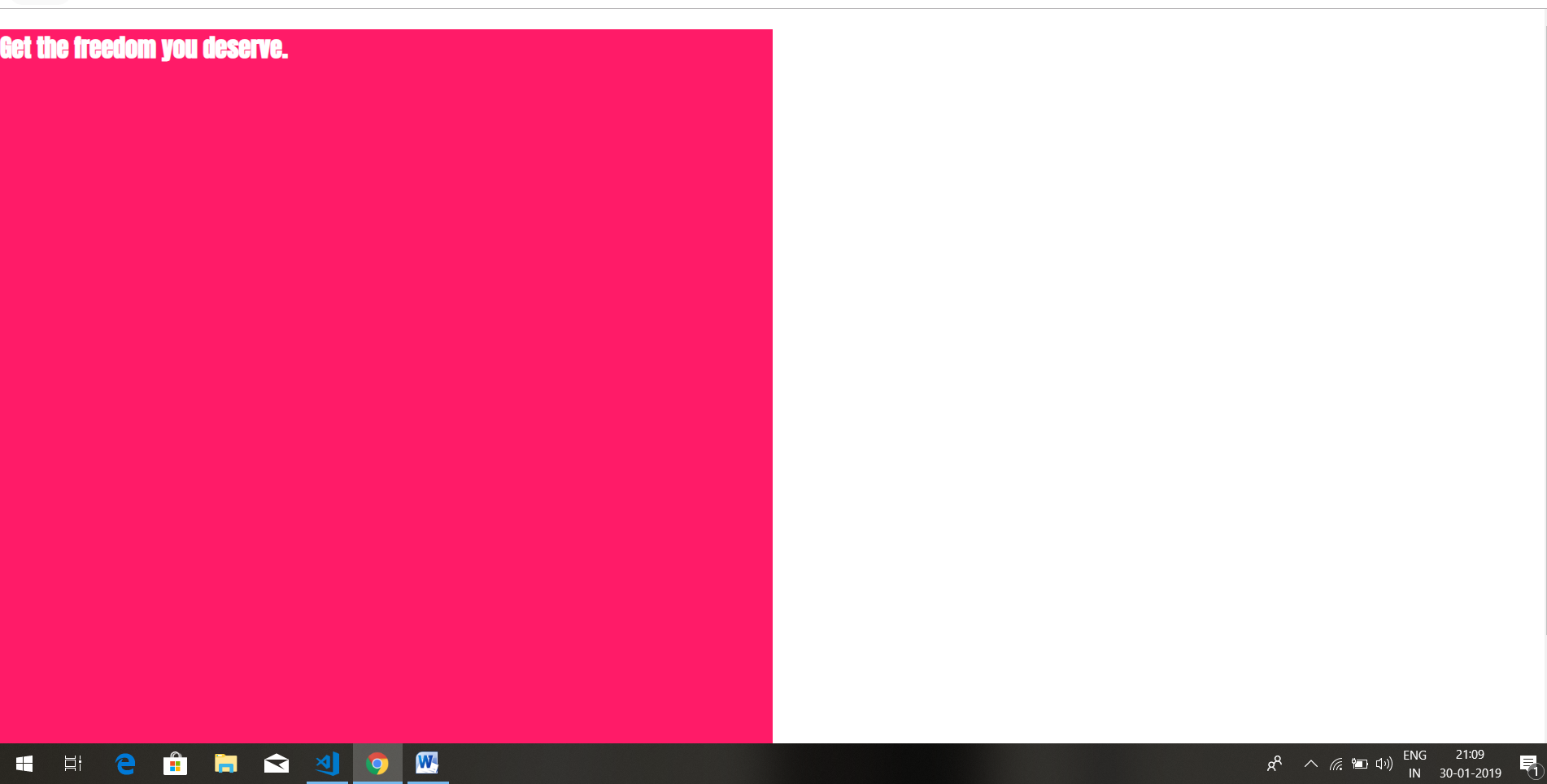


Here both sections were part of main tag, as we are seeing in chrome dev tools.

Now as we can set height of first section is exactly equal to 500px, which is height that we gave to main area. So if we give it as an explicitly height then the child (if we use percentages) will again refer to that(height that we specified) . same thing refers to width.

If we give height of 100% to main section, thn we mean 100% of body, which is again will be as long as it needs to be. So we are in same situation as before.

If we set height of main to 100% then, we are now refreing to its parent which is body, which only big as it needs to be. So we are in same situation as before. So only if we set body equal to 100% height and we do same for html , which is parent of body, which is also as big as it needs to be .after setting html to height of 100%, we can see that now section takes full hright of web page.



This is because, now , if we set equal to 100%, starting at html elemnt, which now refers to overall window(if set height of html equal to 100%), we pass relative height of 100% down to section element.

So if you ever want to style the height of an element relative to height of your page you need to create such a chain where you pass the page height down. In dimension and units section , later in course, we will also learn about a different more modern unit which allows you to achieve this a bit easier with browser support though. But for now this is cool trick to keep in mind, how you can set the height to 100 percent.

This is not something I want to do here though, I can simply set height of first secton to absolute value.

#product-overview {

background: #ff1b68;

width: 100%;

height: 528px;

}

Here we set it to 528 px. So this is how height and width work in general. We can use them with percentages or pixels or other units as we will see later. But there is actually a bit more to width and height especially if we are considering the box model.

Setting 100% height to a child can have an effect only if the parent element has an explicitly applied height. If the parent element determines its height from the heights of the children this is not possible (since it would be kind of an infinite loop).

go through all q and A of this lectures.

See this questions-

1)why second elemnt goes outside container if first elemnt has height of 100%

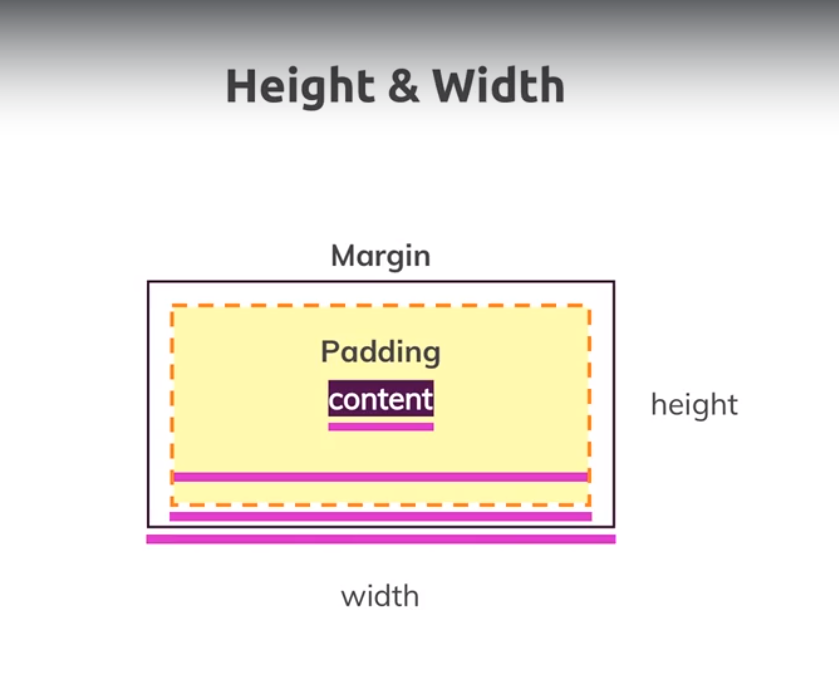
<https://www.udemy.com/css-the-complete-guide-incl-flexbox-grid-sass/learn/v4/questions/4004770>

2) <https://www.udemy.com/css-the-complete-guide-incl-flexbox-grid-sass/learn/lecture/9466108#questions/6253666>

3) <https://www.udemy.com/css-the-complete-guide-incl-flexbox-grid-sass/learn/lecture/9466108#questions/6250544>

32)Understanding Box sizing

In last lecture we set the height and width, but we mean by height and width-



Does we mean width of content, or width of content+margin and so on.

Lets see what it means. Html is-

<body>

<main>

<section id="product-overview">

<h1>Get the freedom you deserve.</h1>

</section>

<section id="plans">

<h1 class="section-title">Choose your plan</h1>

<p>make sure you get the most out of your money</p>

</section>

</main>

</body>

Main.css-

body {

font-family: 'Montserrat', sans-serif;

margin: 0;

}

#product-overview {

background: #ff1b68;

width: 100%;

height: 528px;

}

.section-title {

color: #2ddf5c;

}

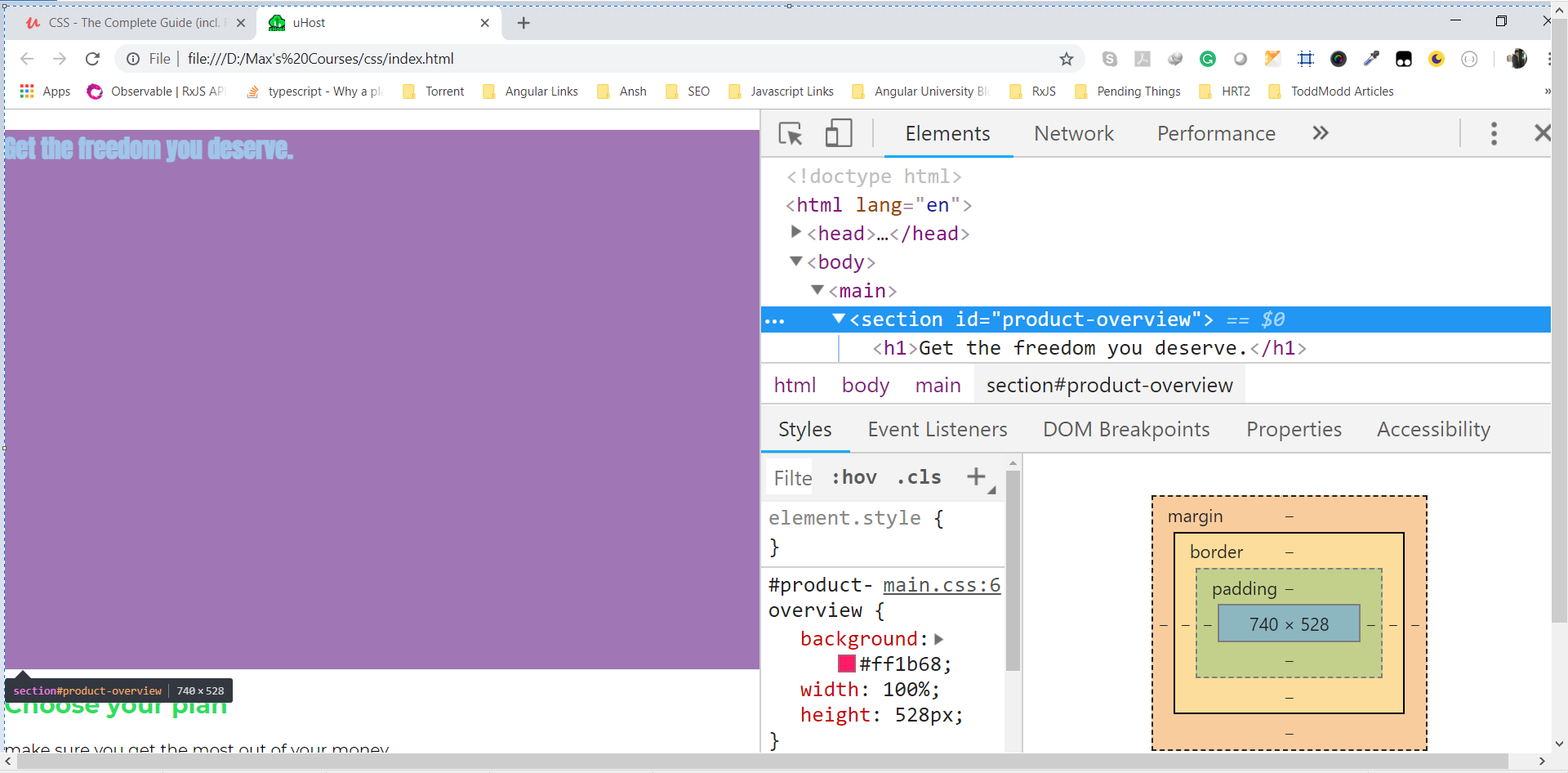
#product-overview h1 {

color: white;

font-family: 'Anton', sans-serif;

}

Here we give the height of 528 px to section with id product-overview. If we inspect this section in chrome dev tools(in box model), we can see that 528 px is height of content(it does not include border, margin and padding)-



Now lets give padding, border and margin to our section.

#product-overview {

background: #ff1b68;

width: 100%;

height: 528px;

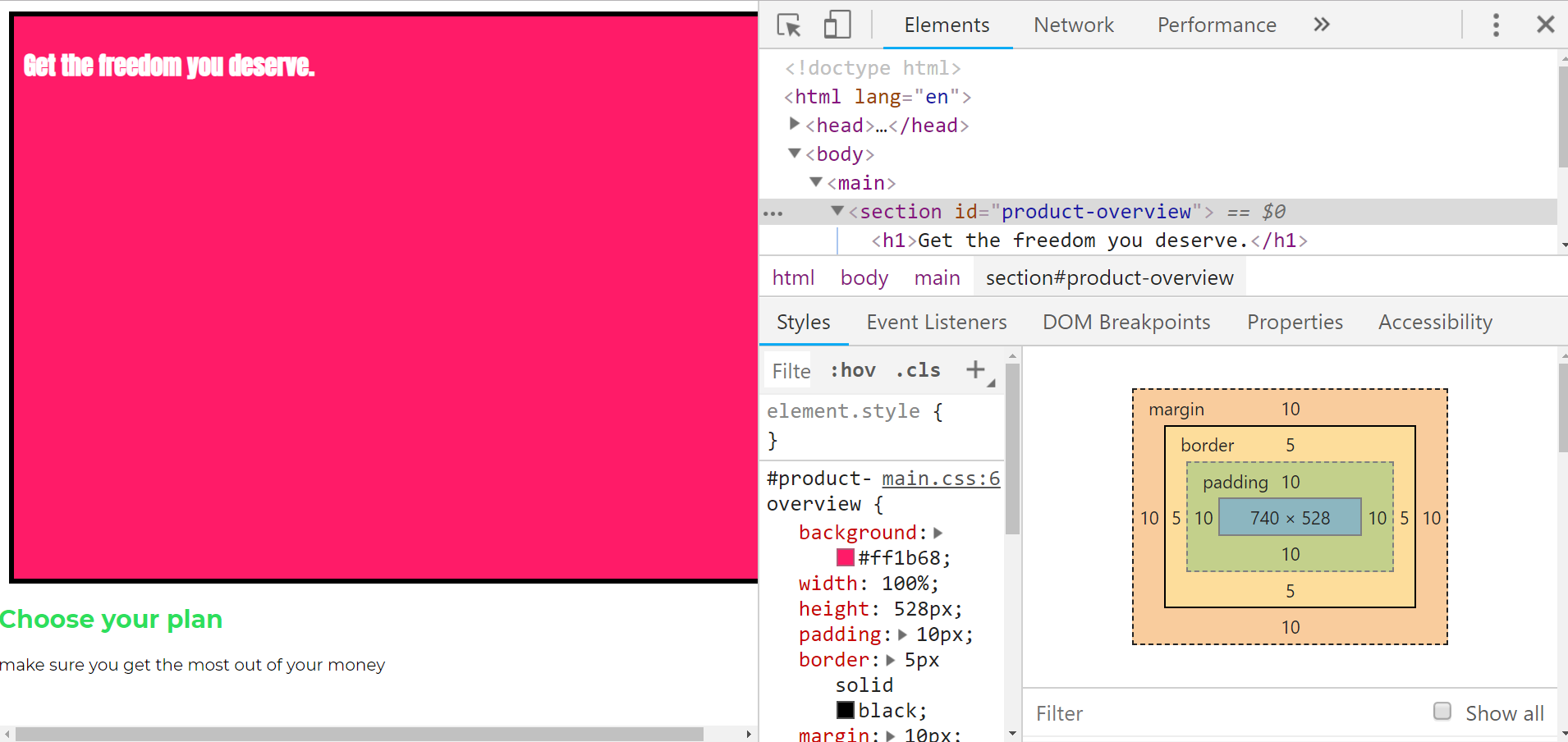
padding: 10px;

border: 5px solid black;

margin: 10px;

}

This is what we get now-



Now our element is not fitting into screen and we got scroll. Also our section is slightly moved to right. So something seems to be wrong with width. This is because width is 100% , it occupies whole screen, but we have margin, padding and border. So that is reason we got scroll.

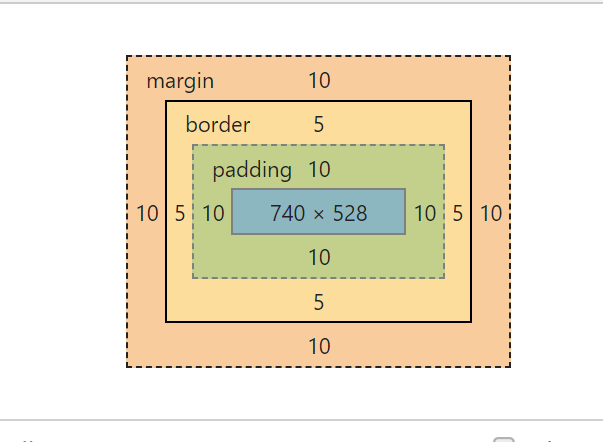
If we inspect the lement we can see that total dimension of elemnt are-



So width of elements is – content width + 2\*(padding+border). Same is case with border. Margin is not included in it.

So 740+2\*(10+5) = 770

If we see box model in chrome dev tools we can see this-



So when we specify width and height in css, we set width and height of content, padding and border is not included in it. But in the end while positon browser adds these things , so our elements is not positioned correctly.

This happens bcoz all elements by default happen to have a certain way of calculating width and height which is called **content box.** We can set it by using **box-sizing** properties. Default value of this property is content-box. Like this-

#product-overview {

background: #ff1b68;

width: 100%;

height: 528px;

padding: 10px;

border: 5px solid black;

margin: 10px;

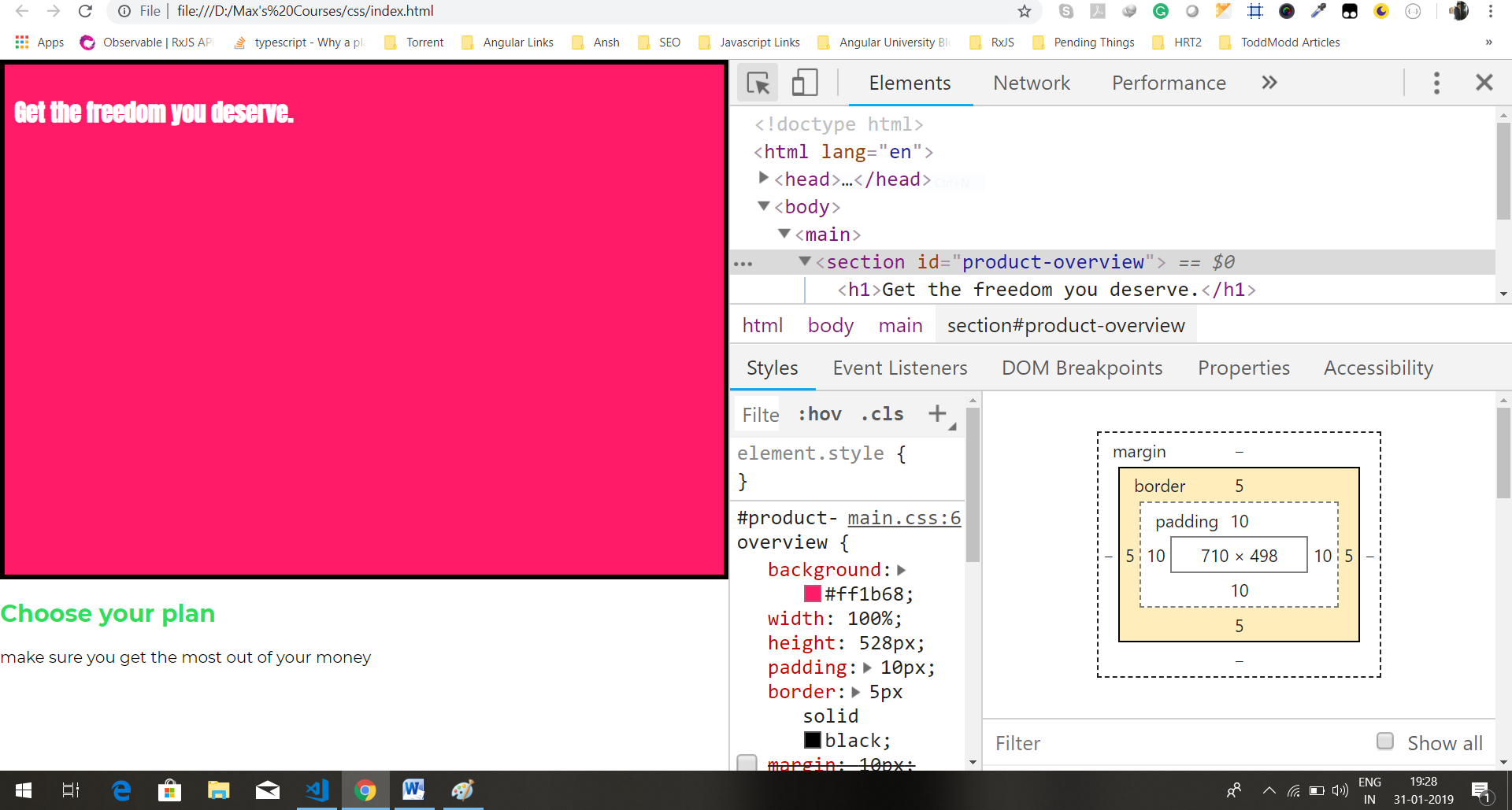
box-sizing: content-box;

}

Content-box menas when we set width and size , we set width and size of content, not of entire box(hence name of property, box-sizing) including padding and border. But if we do this-

box-sizing: border-box;

now width and height also include padding and border and that dnt include margin and we cnt make it to include that. Now if we save code and remove margin(we dnt remove margin then our content will go out from page and we will get horizontal scroll), we see this-

so width 100% means width of content+padding+border and as we have removed margin content is fitting in screen. So this is setting that we want to use, that we want width and height to include content, margin and border. It’s actually so common that you often want to override the styling for all elements to always use box-sizing as border-box. How do we do that? Lets try to add it to body selector.

body {

font-family: 'Montserrat', sans-serif;

margin: 0;

box-sizing: border-box;

}

#product-overview {

background: #ff1b68;

width: 100%;

height: 528px;

padding: 10px;

border: 5px solid black;

/\* margin: 10px; \*/

}

.section-title {

color: #2ddf5c;

}

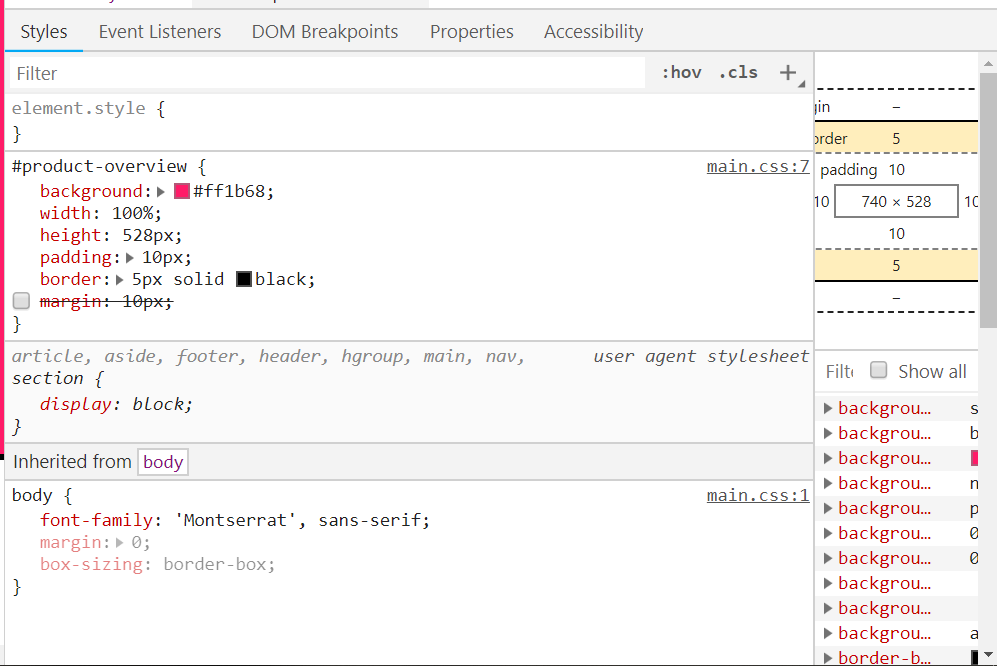
#product-overview h1 {

color: white;

font-family: 'Anton', sans-serif;

}

Now we removed it from product-overview selector and added it to body selector. we might think that product-overview will get it by inheritance. We see webpage, we are back in content-box world. If we inspects section with id product-overview, then we see-



we are back in content-box world. The reason for this is that we are now inheriting the box-sizing set up(from body), but actually we dnt see it here as a browser default style(that we cannot see box-sizing in browser defaut styles) but because it is block level element(we are talking about section), this is overwritten automatically. So our inheritance doesn’t take effect because the browser sets it own box sizing because section is block level element. So we have to use universal selector.

this is rare cases were we really use it, if you want to reset all elements to have certain behaviour like using box-sizing: boder-box. Then we use universal selector.

questions-

1)why box-sizing property is blurred-

<https://www.udemy.com/css-the-complete-guide-incl-flexbox-grid-sass/learn/v4/questions/6262878>

33)adding header to our project

Here we added a header to our html like this-

<body>

<header class="main-header">

<div>

<a href="index.html">uHost</a>

</div>

<nav>

<ul>

<li>

<a href="packages/index.html">packages</a>

</li>

<li>

<a href="customers/index.html">Customers</a>

</li>

<li>

<a href="start-hosting/index.html">Start Hosting</a>

</li>

</ul>

</nav>

</header>

<main>

<section id="product-overview">

<h1>Get the freedom you deserve.</h1>

</section>

<section id="plans">

<h1 class="section-title">Choose your plan</h1>

<p>make sure you get the most out of your money</p>

</section>

</main>

</body>

Now we want to style it so we added a class to header. We want header to be of green color, have some padding and width 100%.

.main-header {

width: 100%;

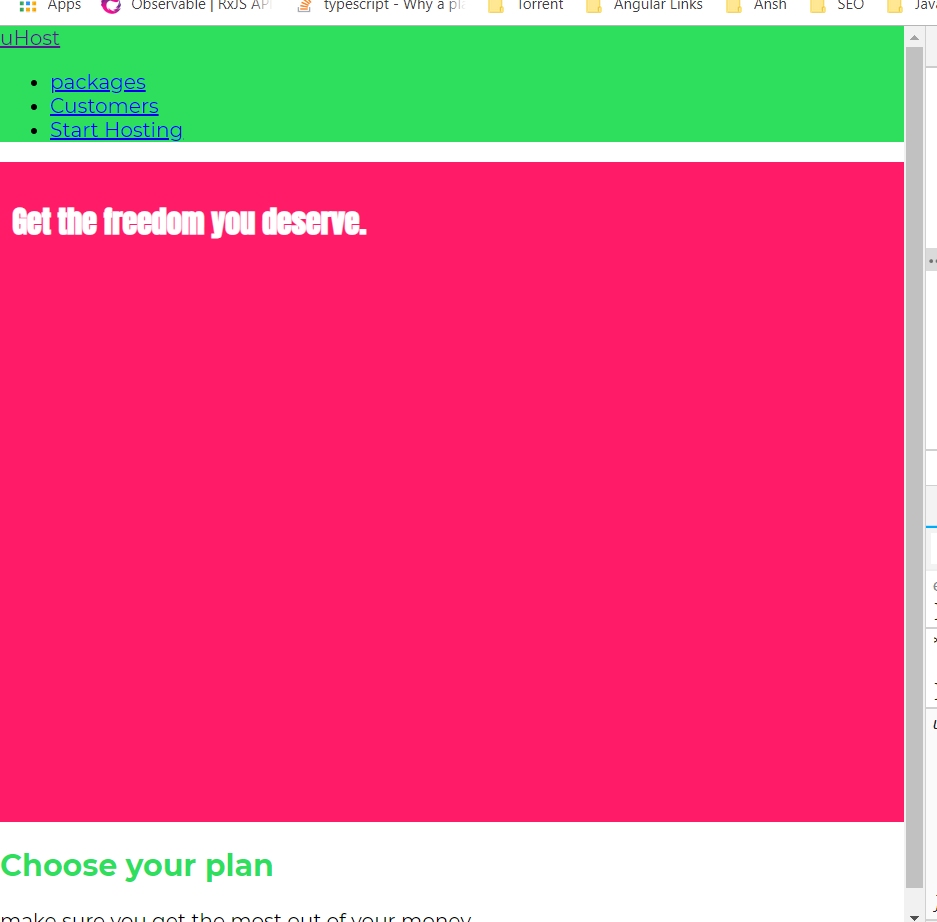
background: #2ddf5c;

padding: 8px 16px;

}

We set width to 100%, but it is optional as block level elements always takes 100 of width.

One thing is if we remove padding from main-header class, we will see gap between header and main section. it is because of margin collapsing. here is question regarding that. Like this-



Queastion-

1)margin collapsing

<https://www.udemy.com/css-the-complete-guide-incl-flexbox-grid-sass/learn/v4/questions/4887812>

now we want list items to sit next to each other, not on top of each orther. To achieve this we need to dive into display property.

34)understanding display property

We need to use display properties to change how our navigation links look. First of all we need to target . for that we give them class names. I will us certain way of writing classes which is called BEM(block element modifier), something I which will explain at the end of this coyrse in pattern and best pratices part. It’s just a specific way of writing classes that should ensure that we don’t accidentally reuse class name and they are very clear about what they do.

<header class="main-header">

<div>

<a href="index.html">uHost</a>

</div>

<nav class="main-nav">

<ul class="main-nav\_\_items">

<li class="main-nav\_\_item">

<a href="packages/index.html">packages</a>

</li>

<li class="main-nav\_\_item">

<a href="customers/index.html">Customers</a>

</li>

<li class="main-nav\_\_item">

<a href="start-hosting/index.html">Start Hosting</a>

</li>

</ul>

</nav>

</header>

To nav we give class main-nav. To ul we give class main-nav\_\_items. And to li we give class main-nav\_\_item. We are in main-nav, there we are in sigle item, hence class main-nav\_\_item. \_\_ means we are in main-nav, after \_\_there is a section of main\_\_nav

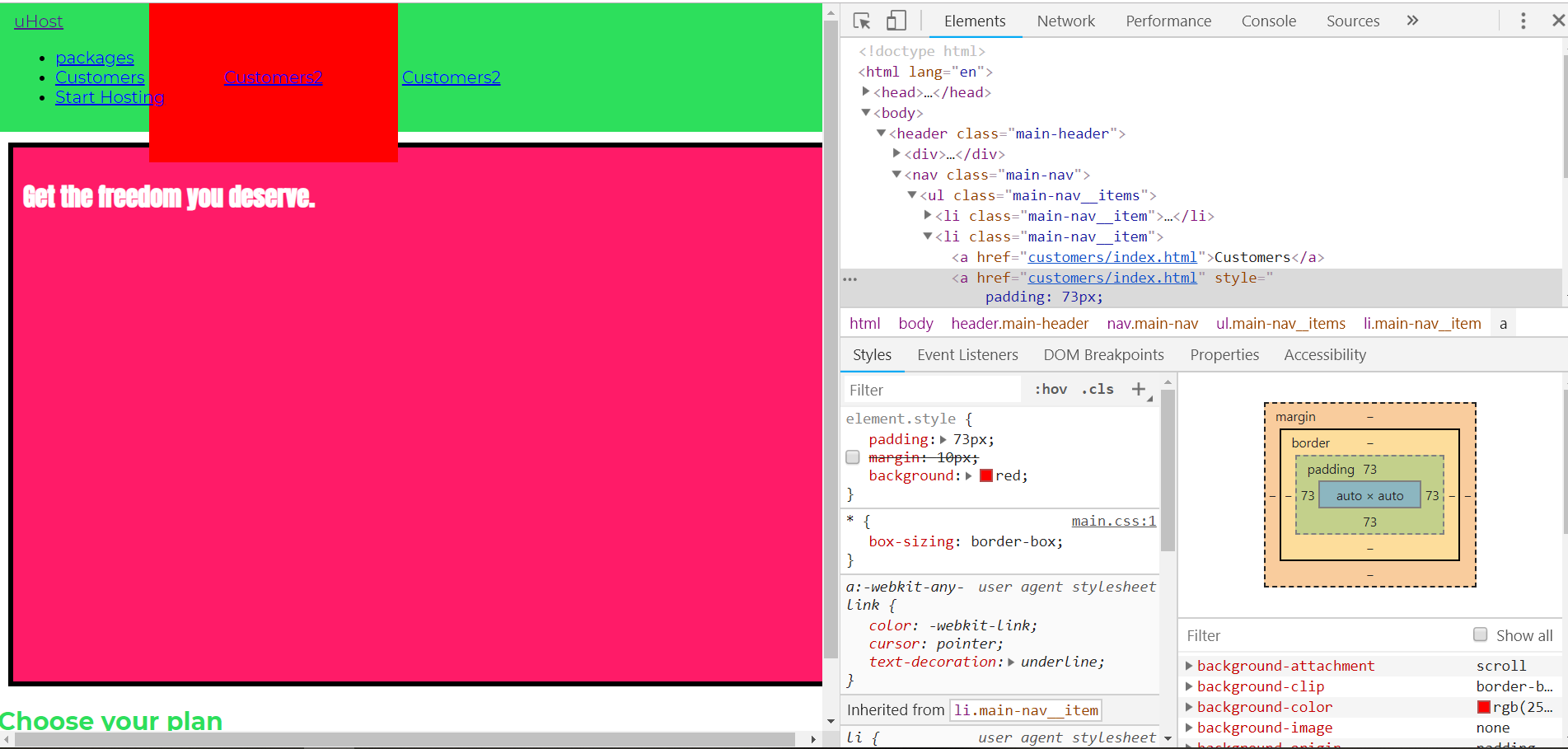
now lets see display property. Display property allows to change the behaviour of element from block to inline or even to inline-block, which is mixture between 2 or to none(to remove it from DOM). In html we got inline and block element. anchor tag(<a>) is inline element. If we have 2 anchor tags sitting next to each other , they will come in same line. They dnt take full width like block level elements. they only take the width that their content requires. If we inspect li tag in chrome dev tools, we can see that it takes full width avalaible(minus padding , margin, what’s taken by other element surrounding it).

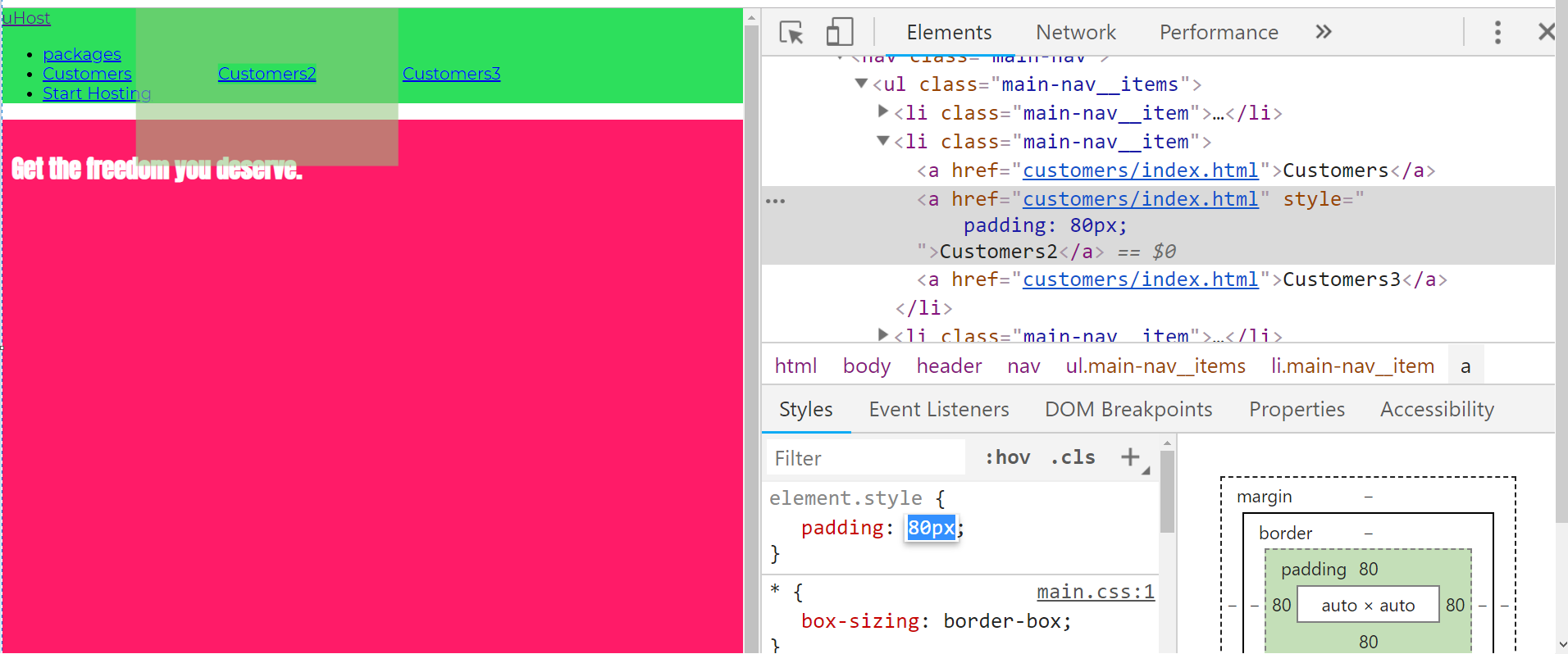
Here is question-

1)What is meant by element surrounding it?

<https://www.udemy.com/css-the-complete-guide-incl-flexbox-grid-sass/learn/v4/questions/6325360>

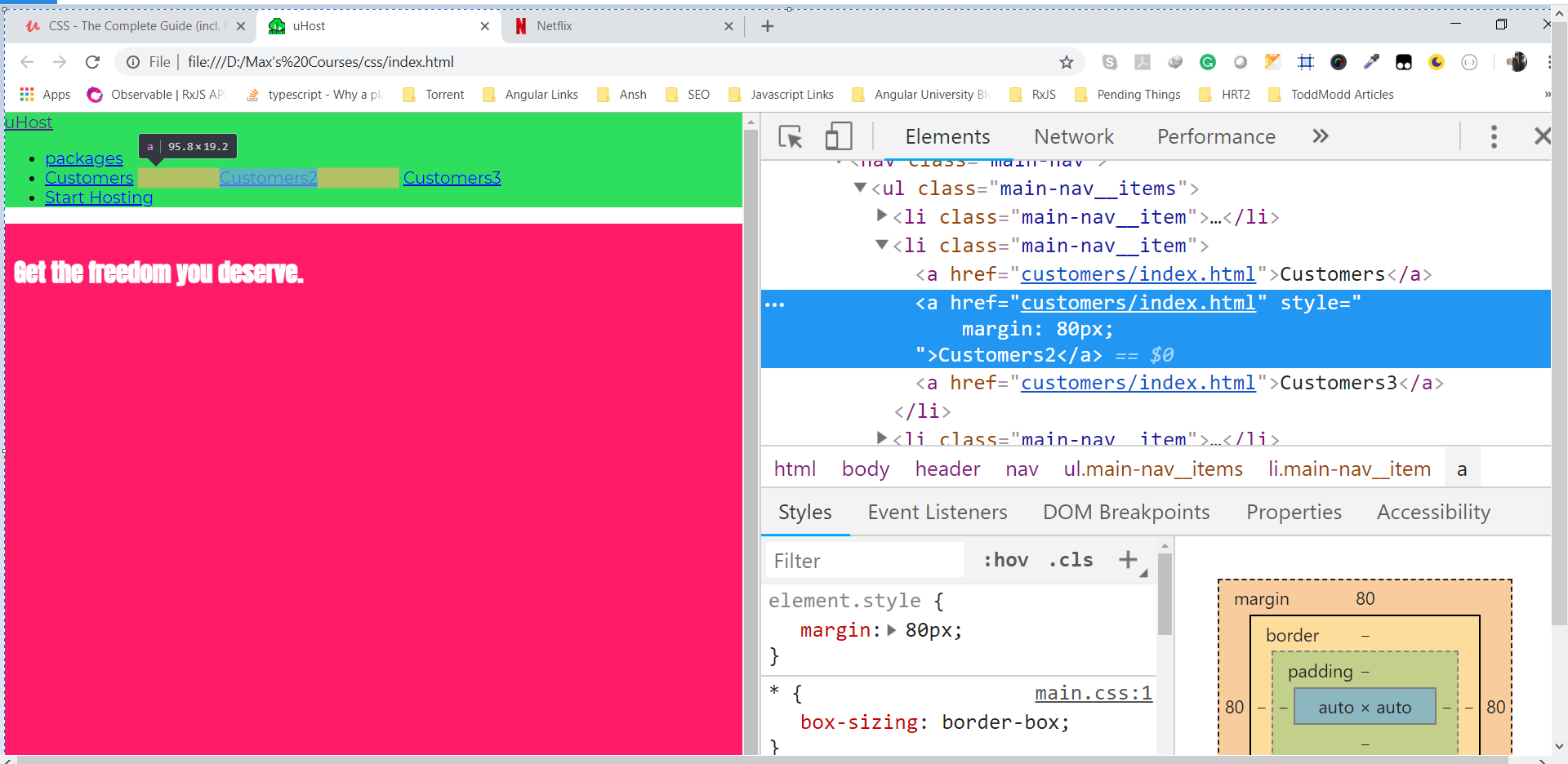
a tag only takes the width, its content needs. Both actually use the box-model(you can see box model for both in chrome dev tools). **though for inline element we can’t set the margin top and bottom for example**, bcoz that is not how inline elements work for example. They are not positioned in the flow like block level element. They don’t take new line necessarily. Hence margin top and bottom is difficult because they might be inline with other element. Lets try to set padding on a tag, this is what we get-

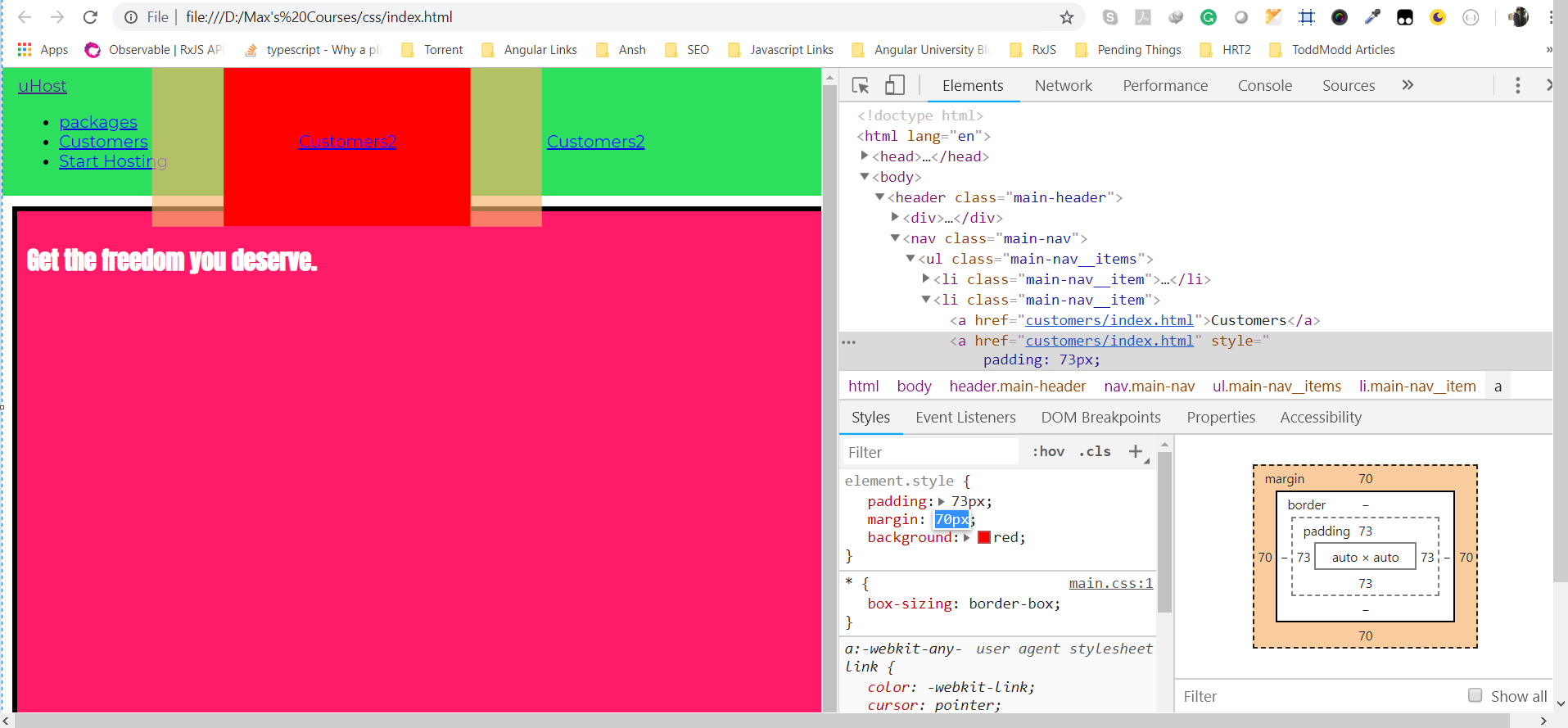




Here you can see padding on all 4 sides are applied. but in case of left and right side padding, elemnts are pushed away. But in case of top and bottom padding, elements are not pushed.

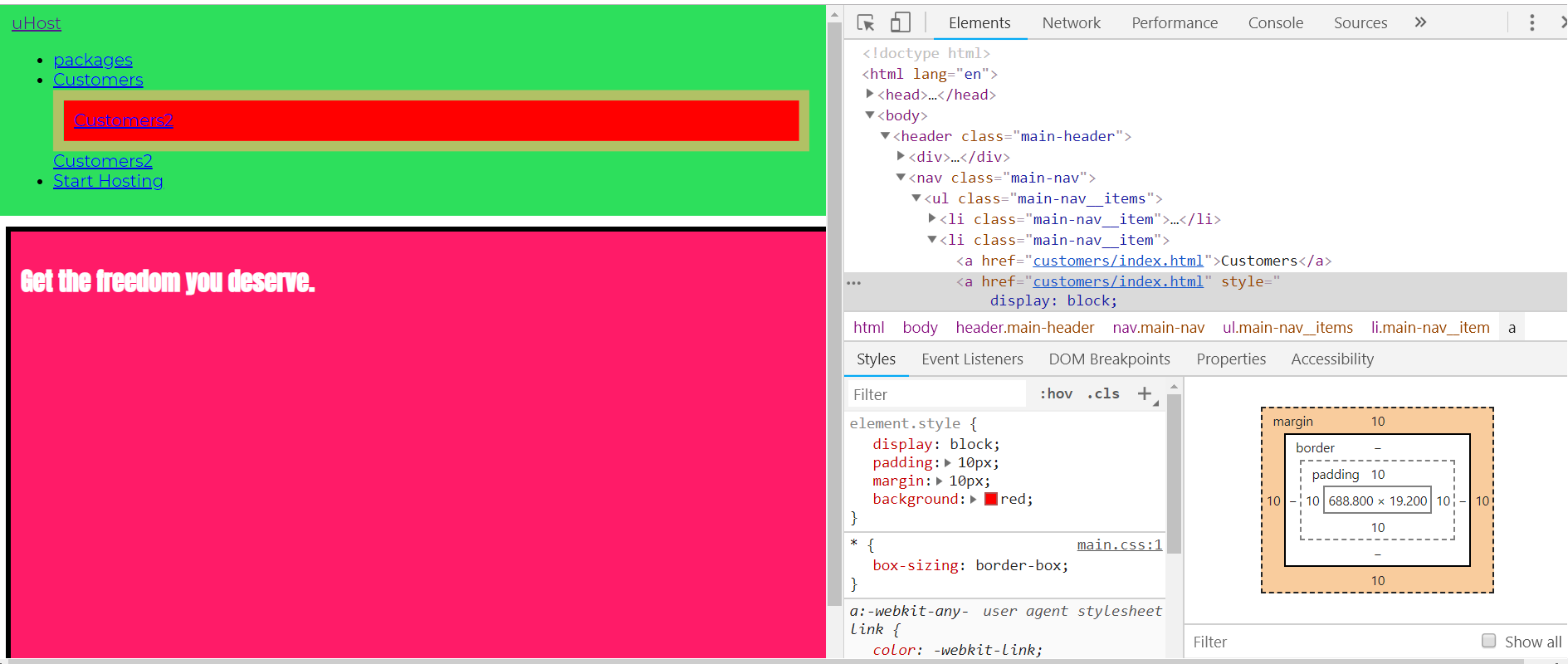
If we give it margin, then-





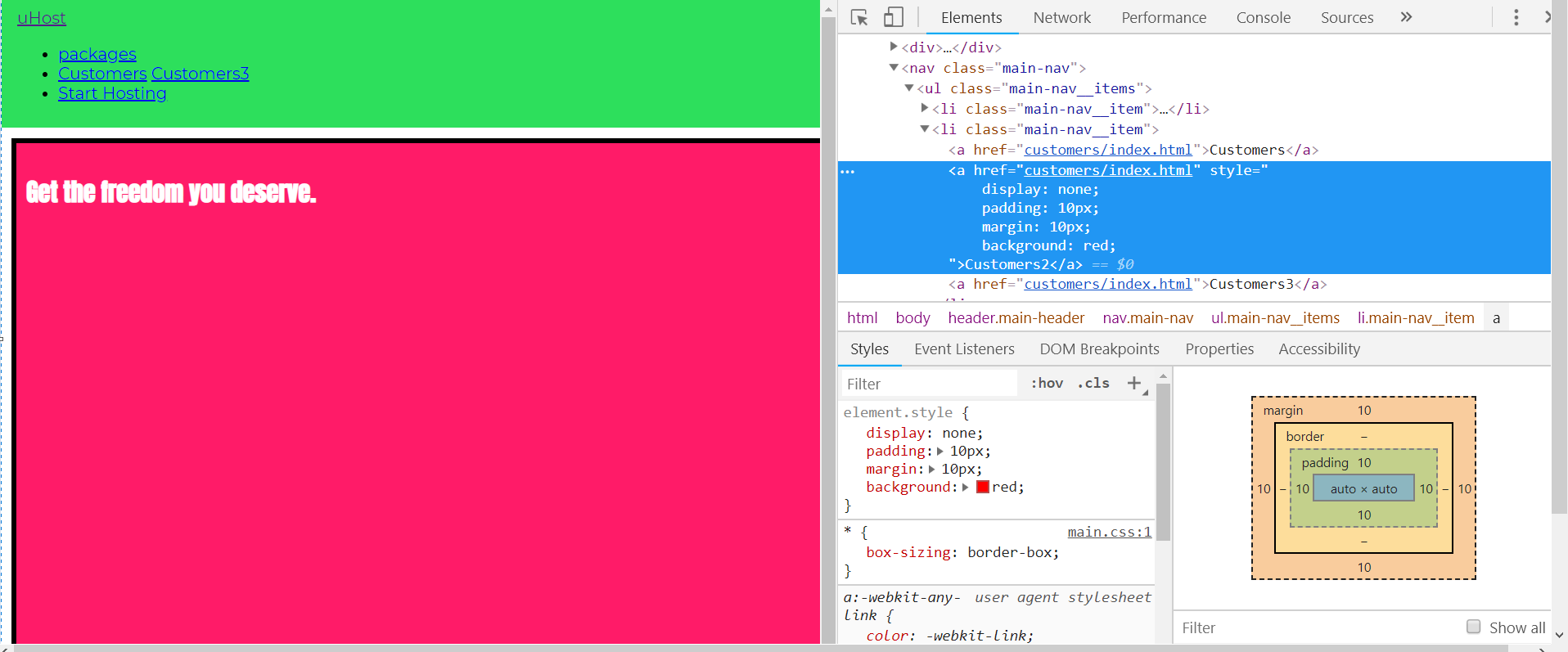
As we can see top and bottom padding is not even applied to inline elements.

So that is the difference(b/w inline and block elements) and we can change the behaviour with display property. If we set display equal to block for a(anchor) tag, we will see that it starts on new line and occupies full width. Lest try to give it margin or padding, this is what we get-



2nd a tag is block level , so it also pushes 3rd a tag(which is still inline) to next line. Also top and bottom padding takes effect as we can see. Elements which are block level by default are like ul, li , nav, section and so on.

We can also set display property to none. Now element will be removed and it’s not only invisible or transparent, it’s taken out of document flow. As there is no padding and margin of customers2 a tag.



But in chrome dev tools we can find that element. So it os stil part of DOM. So it’s not removed from there, just from visible document flow.

The idea behind the display none is , it is used in conjuction with javascript. To show some something when a button was clicked. It is used in modal, which we will see later. So that is values of display property – none, block, inlne. Changing to inline to block and other way around is not useful in most cases because elements behave in certain way for a good reason. But there is also 4th option and that is really useful to us.

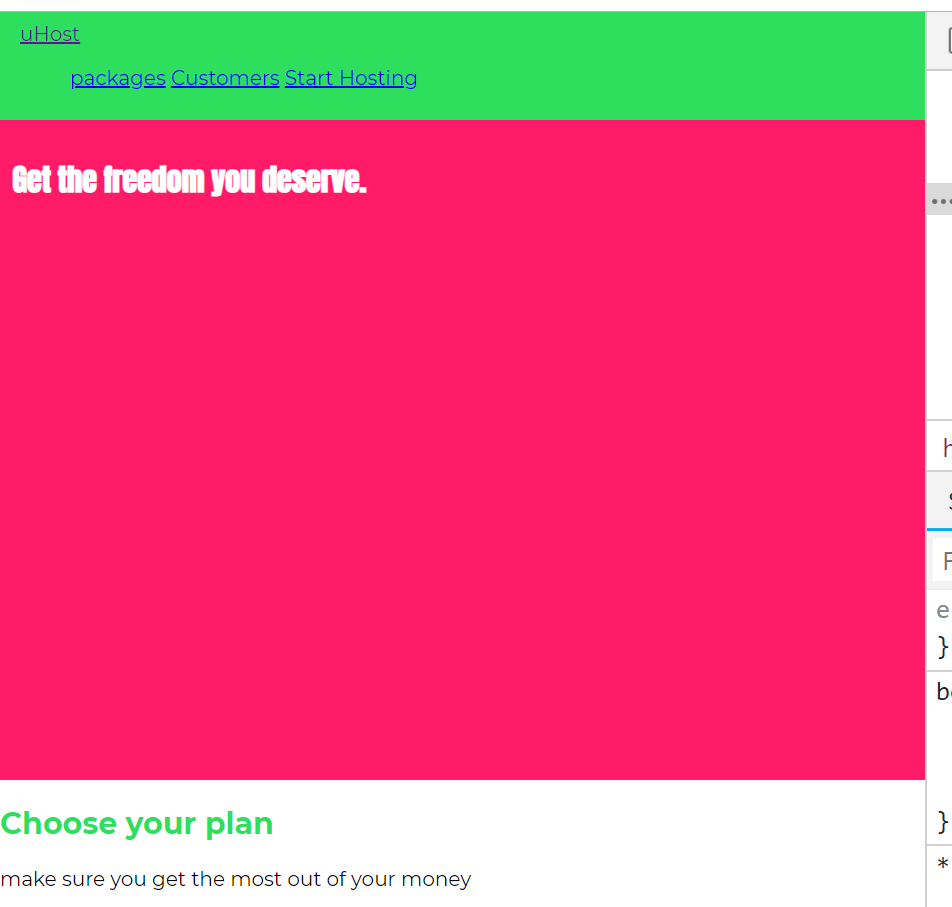
Lets set display of li to inline-block. It mixes the behaviour of both inline and block**. Like inline element now, they can go next to each other. but they still behave like block elements when it comes to setting top and bottom margins, paddings, things that are possible for inline elements.**

.main-nav\_\_item {

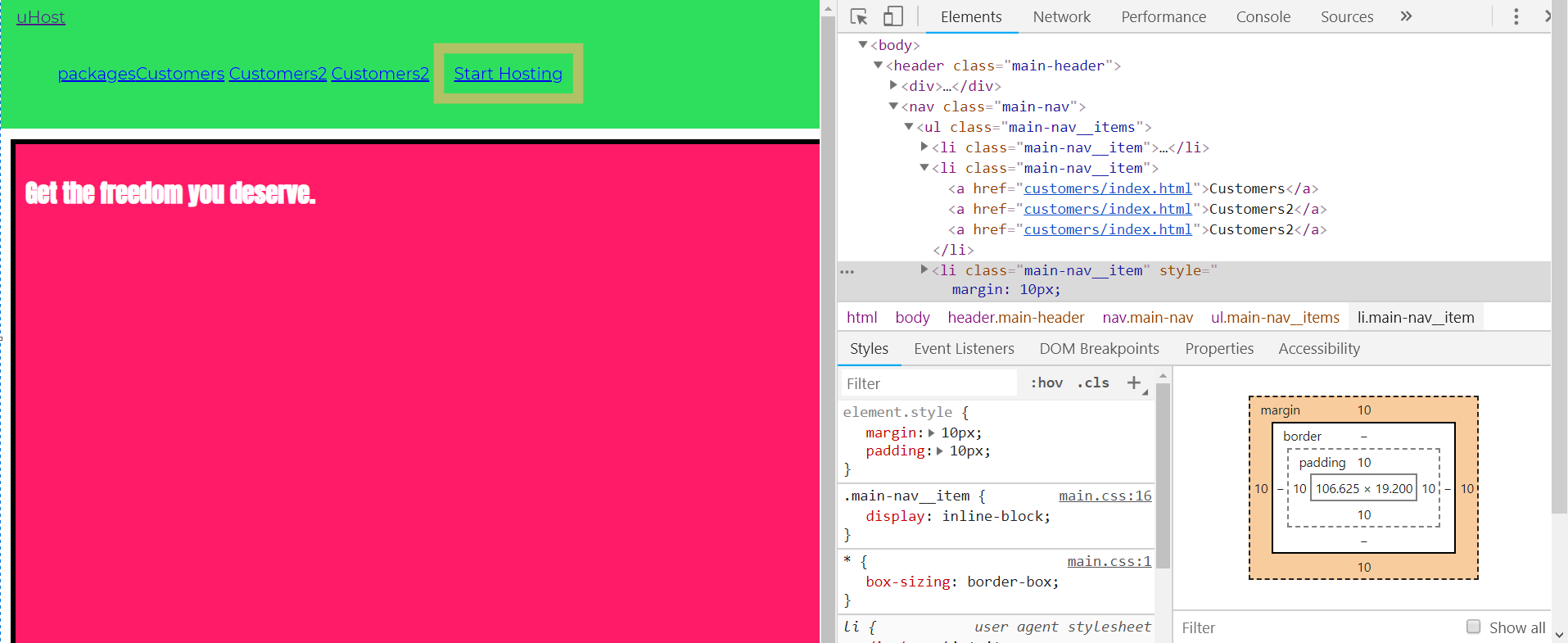
display: inline-block;

}

Now we can see that list items are setup in one line, because again, they behave like inline like inline element such that they only that they only take as much width as their content needs and they fit into one line but they have a box where we can setup padding and so on.



Lets try to setup padding and margin on inline-block. We saw how padding and margin work in case of inline element. If we apply them to a tag, which is inline they will work in same way. But if we apply them to li, which is inline –block-

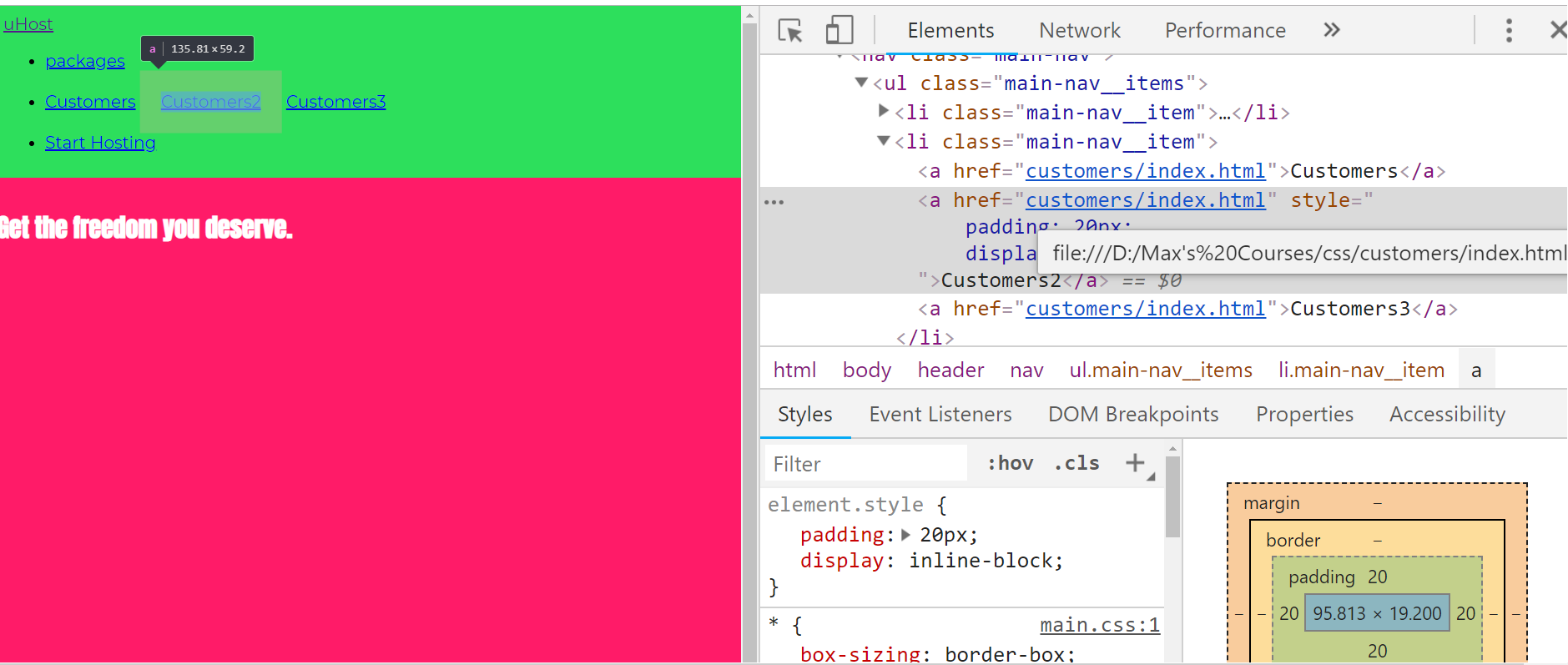


Here we can see that top and bottom margin is also applied. also whole line is moved (vertically)because of margin. Same thing for padding, here also whole line is moved. we saw in case of linline elements, there was no vertical movement of line because of padding and top and bottom margin not even apply for inline lements.

**Now we can still style li as block level element but they sit next to each other like inline elements**. Later in course we will learn about other tools to position our elements, mainly flex box, wihch is even better to position elements next to each others. For now inline elements are great start.

Observation-

1)



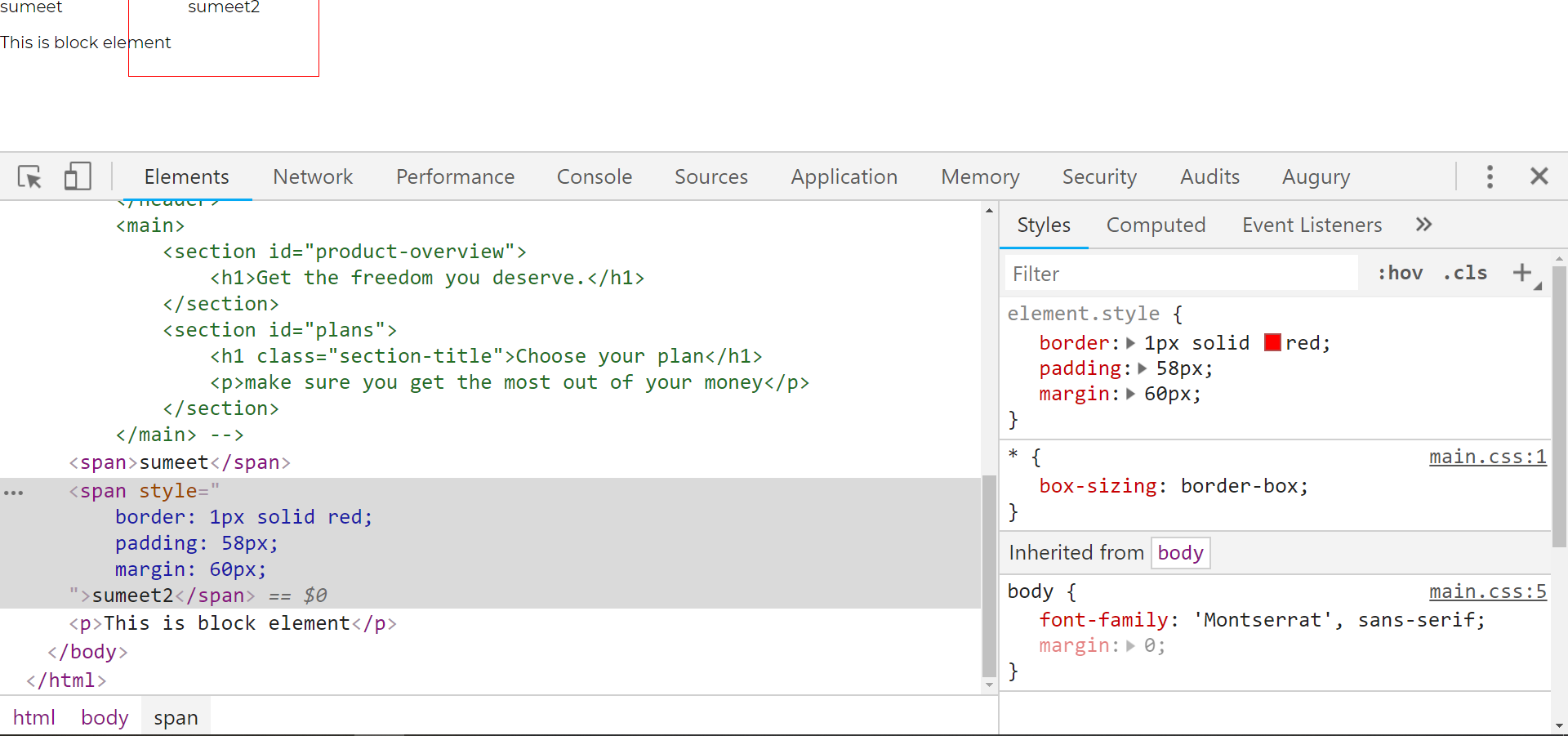
Here display is default. Out in second li we have 3 anchor tags. In chrome dev tools I changed display of cutomers2 a to inline-block and then gave it padding of 20px. Now you can see top and bottom padding is also applied. also note that customers and customers3 a tag(which are still inline) also moved. So whole row has moved.

2) this our html-

<span>sumeet</span>

<span>sumeet2</span>

<p>This is block element</p>



Here I am giving margin and padding to span, which is inline element. you can see that, when we give padding, border is expanded but next element is not pushed. If we increase padding, p tag stays at same position. At this point we can also see that border has also surpassed the p tag. That is how padding behaves, in case of inline element. Element is centered, i.e element is exactly in middle of its box, as we have even padding. But next element which is p tag, it is not moved because of this padding.

When we apply margin to inline element, then only top and bottom margin is not applied.

Question-

1)Here I was applying padding, but top padding was not getting applied(top border was getting hidden). Read this answer

<https://www.udemy.com/css-the-complete-guide-incl-flexbox-grid-sass/learn/v4/questions/6271392>

if inner element is block element instead of inline(lets ay we have section, instead of span) then we would get this-



This question proves a very important point,inline elemnt does not move vertical nor it moves other elements either by padding or margin. This explains why we cannot see top border and top padding in question.

35)display: none vs visibility: hidden

We had a look at display: none;  - this value removes the element to which you apply it from the document flow. This means that the element is not visible and it also doesn't "block its position". Other elements can (and will) take its place instead.

There is an alternative to that though.

If you only want to hide an element but you want to keep its place (i.e. other elements don't fill the empty spot), you can use visibility: hidden;

Here's a visual example:

1. .box-1 {
2. display: none;
3. }
5. .box-2 {
6. display: inline-block;
7. }

Will render:

x

where x  has the class box-2 . The first element just isn't displayed. **It's still part of the DOM though**, **you can still access it via JavaScript for example.**

Here's an example for visibility: hidden :

1. .box-1 {
2. visibility: hidden;
3. }
5. .box-2 {
6. display: inline-block;
7. }

Will render:

\_x

where \_  simply is an empty spot and x  has the class box-2 .

The element is only invisible, **it's not removed from the document flow and of course also not from the DOM.**

36)HTML Refresher: Block level vs inline Element

It's not really a CSS topic, though it's related to it: The difference between **block-level** and **inline elements**.

You can read a more detailed article (which also includes a YouTube video about HTML at the top of the page) here: <https://academind.com/learn/html/beginner-s-guide/diving-deeper-into-html#block-level-vs-inline-elements>

*Here's the executive summary:*

**Block-level elements** are rendered as a block and hence take up all the available horizontal space. You can set margin-top and margin-bottom and two block-level elements will render in two different lines.

Some examples are: <div> , <section> , <article> , <nav>  but also <h1> , <h2>  etc and <p> .

**Inline elements**on the other hand only take up the space they require to fit their content in. Hence two inline-elements will fit into the same line (as long as the combined content doesn't take up the entire space in which case a line break would be added).

They also use the box-model you learned about but margin-top  and margin-bottom  have no effect on the element. padding-top  and padding-bottom  also have a different effect. They don't push the adjacent content away but they will do so with the element border. You can read more about that behavior in the following article: <https://hacks.mozilla.org/2015/03/understanding-inline-box-model/>

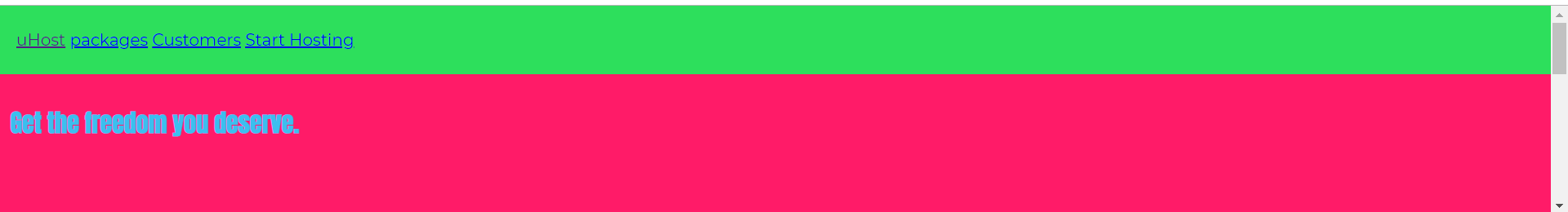
Additionally, setting a width  or height  on an inline element also has no effect. The width and height is auto to take as much space as required by the content.

Logically, this makes sense since you don't want your inline elements to destroy your multi-line text-layout. If you want to do so or need both block-level and inline behavior, you can set display: inline-block  to merge behaviors.

Some example elements are: <a> , <span> , <img>

37)Appling the display property & styling our navigation bar

We want uHost and other links to to be in same line. So we turn both of them(div and nav) into inline-block elements. Now they are in same line.



Now we want our navigation links to be right align. Our html is-

<header class="main-header">

<div>

<a href="index.html">uHost</a>

</div>

<nav class="main-nav">

<ul class="main-nav\_\_items">

<li class="main-nav\_\_item">

<a href="packages/index.html">packages</a>

</li>

<li class="main-nav\_\_item">

<a href="customers/index.html">Customers</a>

</li>

<li class="main-nav\_\_item">

<a href="start-hosting/index.html">Start Hosting</a>

</li>

</ul>

</nav>

</header>

Now we have ul tag, in that we have li tag. Since li tag is turned into inline-block element. So we could use text-align property on nav to align all our text on right(inline-block elements will be treated as text). So we do this-

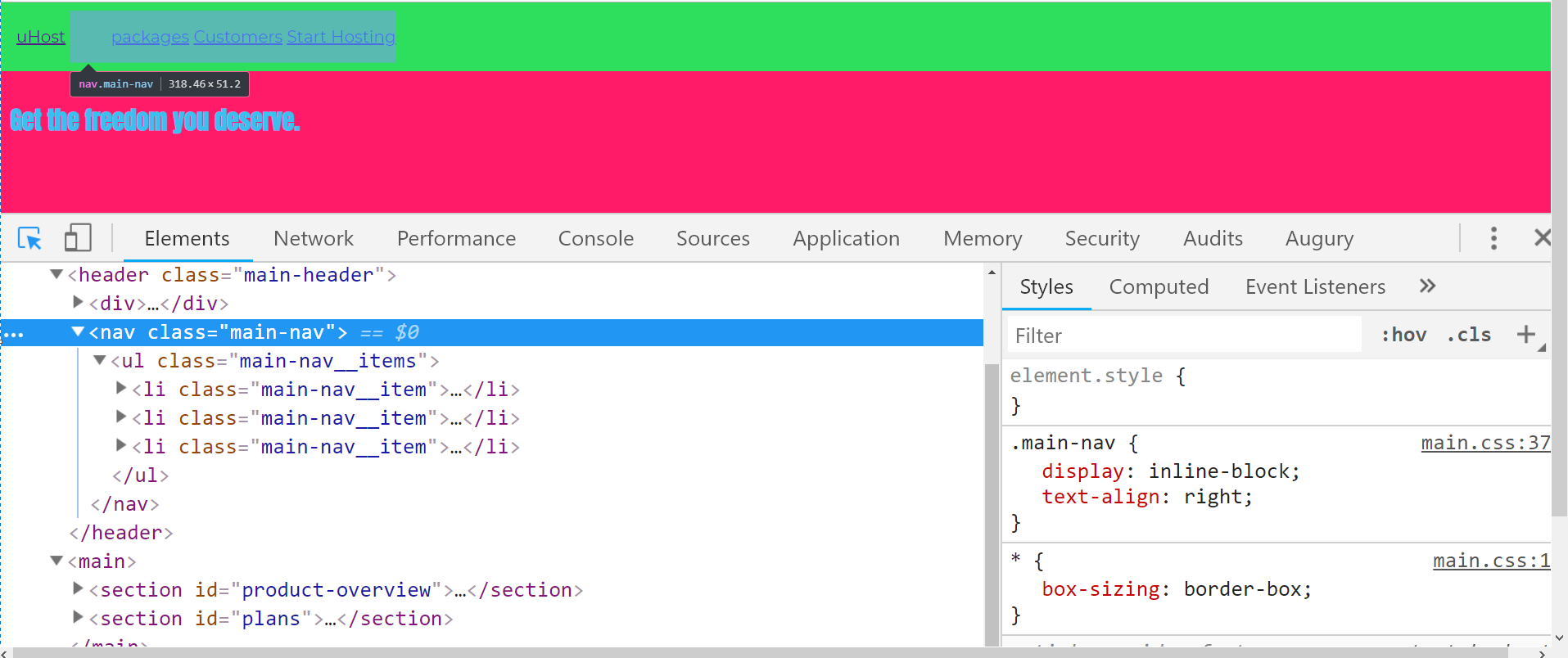
.main-nav {

display: inline-block;

text-align: right;

}

But this has changed nothing. Our webpage is-



Now as we can see that all elements are aligned to right. As you can see in chrome dev tools. You can say that we applied style to nav but li are child of ul. But if you inspect ul, it also has style text-aligned: right due to inheritance. It inherits it from nav.

But as nav is inline-block element, it only occupies as much space as it needs. So now it is not occupying full width.

Note-later in found that text is aligned to right ony because of default padding to ul tag.so as of now size of ul is exactly as it needs to be. So there is no space to see the effect of tet aligned.

So we need to set the width and width should be length our header ,minus width of first link. You can argue we cant we just set width of nav to 100%. So that it occupies whole width. But if we do this then it nav to next line. Like this-



This is because lets say screen width is 500px, it means nav width needs to be 500px. But as some width is occupied by div, 500px is not available on that line. So hence our nav moves to next line.

Then we remove our padding and margin of our ul elements.

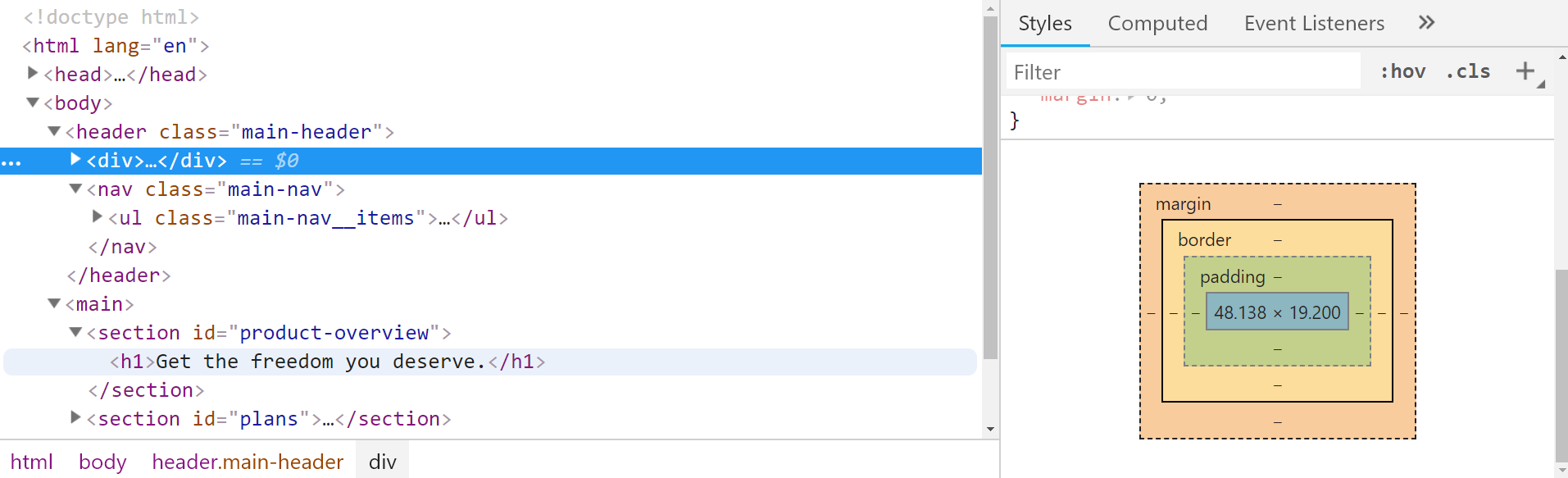
.main-nav\_\_items {

padding: 0;

margin: 0;

}

now we want width of nav as per our calculations. Now we want nav to 100%- width occupied by div. now how do we get width of div. we get it from chrome dev tools, in content box we can see width of div.



This is not correct way of getting width of div. later we will see another methods of getting width, when we have some advance tools. To give this calculation a value, we use calc function. lets take 49 as value(round off)

.main-nav {

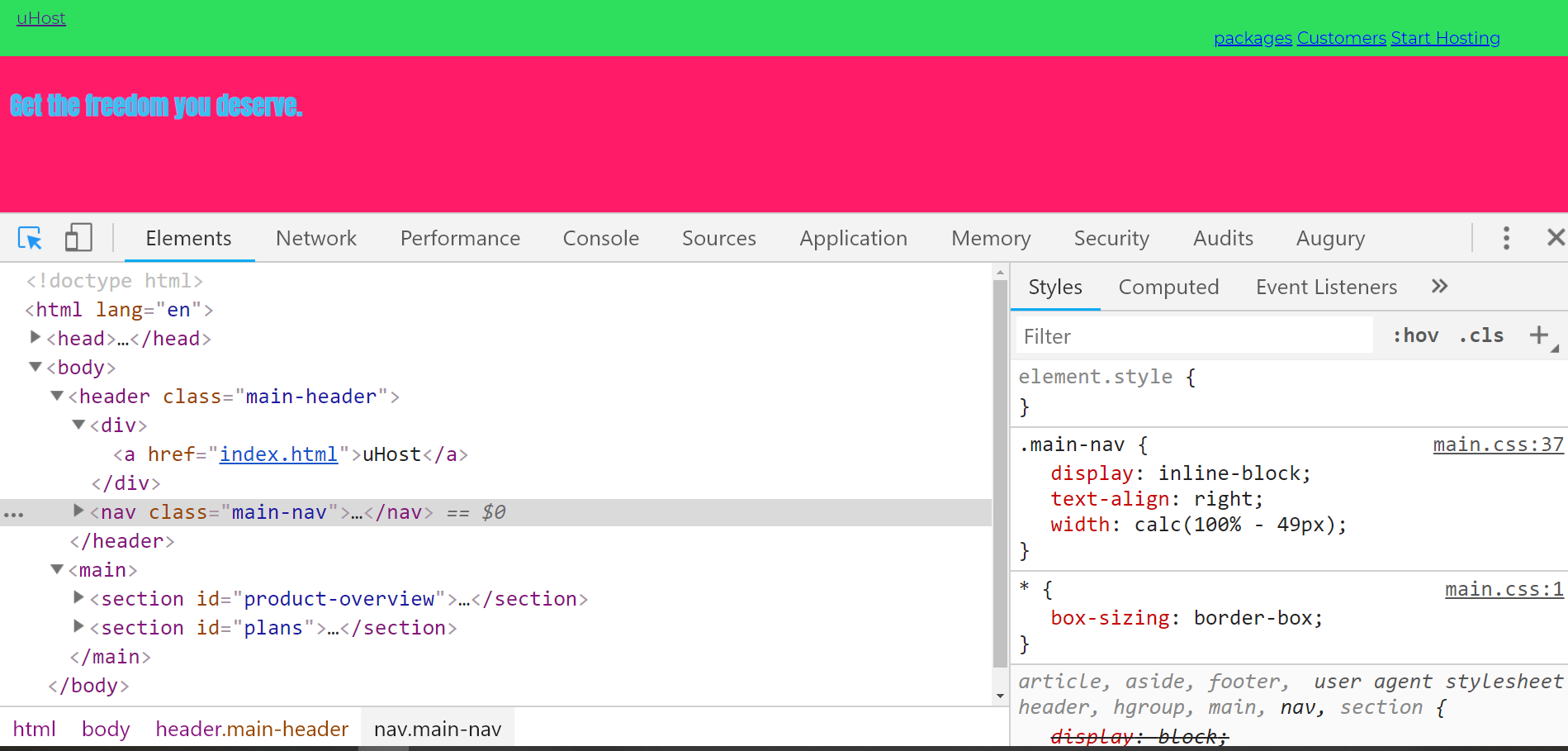
display: inline-block;

text-align: right;

width: calc(100% - 49px);

}

Even after doing this we can see that nav moves to next line.



Now this is something related to inline-block and it’s super important to be aware of this behaviour because that can cost you hours otherwise.

38)Understanding unexpected “inline-block” behaviour

Reason for nav going t next line is, something related to inline block.

in our editor we have this-

<header class="main-header">

<div>

<a href="index.html">uHost</a>

</div>

<nav class="main-nav">

<ul class="main-nav\_\_items">

<li class="main-nav\_\_item">

<a href="packages/index.html">packages</a>

</li>

<li class="main-nav\_\_item">

<a href="customers/index.html">Customers</a>

</li>

<li class="main-nav\_\_item">

<a href="start-hosting/index.html">Start Hosting</a>

</li>

</ul>

</nav>

</header>

We have white space between div and nav. this whitespace is considered a character and is added as an extra line element. So we not only have to substract width of div from 100% but also width of this extra element. How do we get it’s width. Because we dnt even see it in chrome dev tools. One way to handle this is to remove space from ide. Like this-

<header class="main-header">

<div>

<a href="index.html">uHost</a>

</div><nav class="main-nav">

<ul class="main-nav\_\_items">

<li class="main-nav\_\_item">

<a href="packages/index.html">packages</a>

</li>

<li class="main-nav\_\_item">

<a href="customers/index.html">Customers</a>

</li>

<li class="main-nav\_\_item">

<a href="start-hosting/index.html">Start Hosting</a>

</li>

</ul>

</nav>

</header>

Other way to handle this is, we can account for extra width in our calculations. Like this-

.main-nav {

display: inline-block;

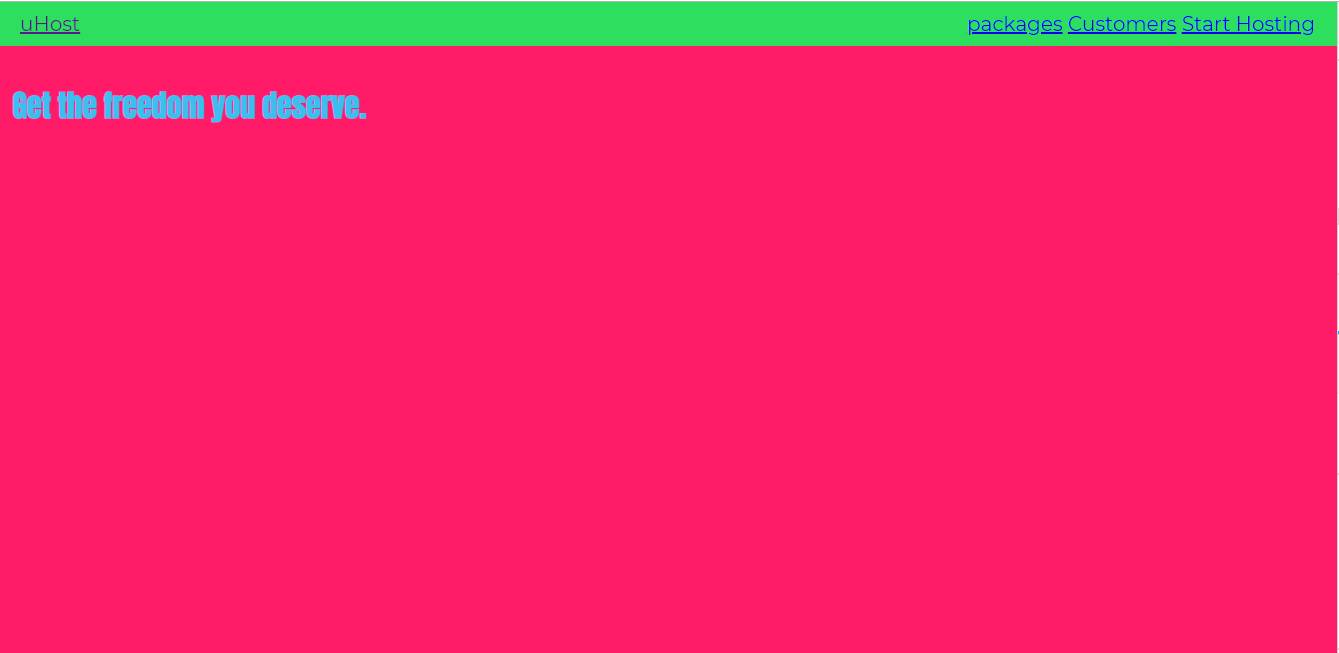
text-align: right;

width: calc(100% - 54px);

}

Now this is a hack. Later in the course we will see a proper approach to handle it.

Here we will go with second approach-



39)Working with “text-decoration” & “vertical-align”

Here we changed the styling of brand link. We do this-

.main-header\_\_brand {

color: #0e4f1f;

text-decoration: none;

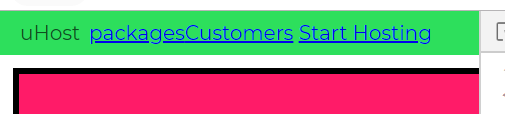
font-weight: bold;

font-size: 22px;

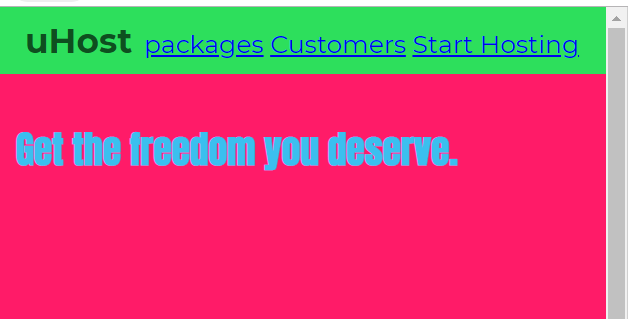
}

Note here font-weight bold works because in imports in index.htm we are also importing bold version of font.

Before-



After-



Now we can see that right navigation links are not aligned in middle anymore(middle of green).this is because font of logo is bigger. And height of header is set by height of logo because it is bigger than rest of links. Rest of links are allinged to bottom, but logo is at middle.

To fix this, we use vertical-align property. We set this property on nav(inline block element), the element which sits next to div holding our brand. Now we need to set this property on both elements which needs to be aligned though. So we need to set it up on div also.

.main-header > div {

display: inline-block;

vertical-align: middle;

}

.main-nav {

display: inline-block;

text-align: right;

width: calc(100% - 74px);

vertical-align: middle;

}



Now div and nav are aligned to each other in their middles.

Here is article that explains the verticle align-

<https://css-tricks.com/what-is-vertical-align/>

also see this link-

<https://christopheraue.net/design/vertical-align>

question-

1)nav also moves when I set vertical-align for div

<https://www.udemy.com/css-the-complete-guide-incl-flexbox-grid-sass/learn/v4/questions/6343944>

2)here is question, on setting height 100% of div, moves the navigation bar-

<https://www.udemy.com/css-the-complete-guide-incl-flexbox-grid-sass/learn/v4/questions/5966378>

40)Styling the anchor tags

Html-

<nav class="main-nav">

<ul class="main-nav\_\_items">

<li class="main-nav\_\_item">

<a href="packages/index.html">uHost</a>

</li>

<li class="main-nav\_\_item">

<a href="customers/index.html">Customers</a>

</li>

<li class="main-nav\_\_item">

<a href="start-hosting/index.html">Start Hosting</a>

</li>

</ul>

</nav>

Here we want to style the anchor tags on the right. First we need to have margin, so we add it in css-

.main-nav\_\_item {

display: inline-block;

margin: 0 16px;

}

Now we dnt want to have underline in a tags. But we set decoration to none in above selector it wnt work. This is because underline is set in browser defaults for a tag. So if we set text decoration to none in above selector and expect a tag to inherit it(which it does not do here), even then browser default will overidr this, because browser defaults are directly selecting the a tag, so it has higher specificity.

Now we have 2 ways of removing inderline. First we can give class to each a tag in header and then use that class a sselector, second way is we can use combinator. now which approach is better. From performance perspective, assigning a class should be little bit faster, though I will say unless you are not building the next facebook atleast, you will probably not notice it. And especially ofr this special cases, where you have one anchor tag directly , so as a direct child of item with our class. This should be high peformant selector too.

.main-nav\_\_item a {

text-decoration: none;

color: #0e4f1f;

}

So performance isn’t even that much worse here. so we will use it. We will not create extra class here.

Question-

1)top and margin works on inline-block element if we have added vertical align to that element or any of its parents-

<https://www.udemy.com/css-the-complete-guide-incl-flexbox-grid-sass/learn/v4/questions/4046282>

2)counter question to question1-

<https://www.udemy.com/css-the-complete-guide-incl-flexbox-grid-sass/learn/v4/questions/6374714>

3)text-decoration property not inherited-

<https://www.udemy.com/css-the-complete-guide-incl-flexbox-grid-sass/learn/v4/questions/5788656>

41)Addding psedo classes

Her we want to style anchor element when we hover over it and when we click and hold on it. We can do this is css with pseudo classes.

Here we first select a tag, then we apply pseudo class, we add psedo class by addng : after a tag selector(wihout space), space will mean another descendent cobminator, it would select select element over which we hover which are nested inside anchor tag. Hover is one of many pseudo classes that we can attach. Another one can be active .

.main-nav\_\_item a:hover {

color: white;

}

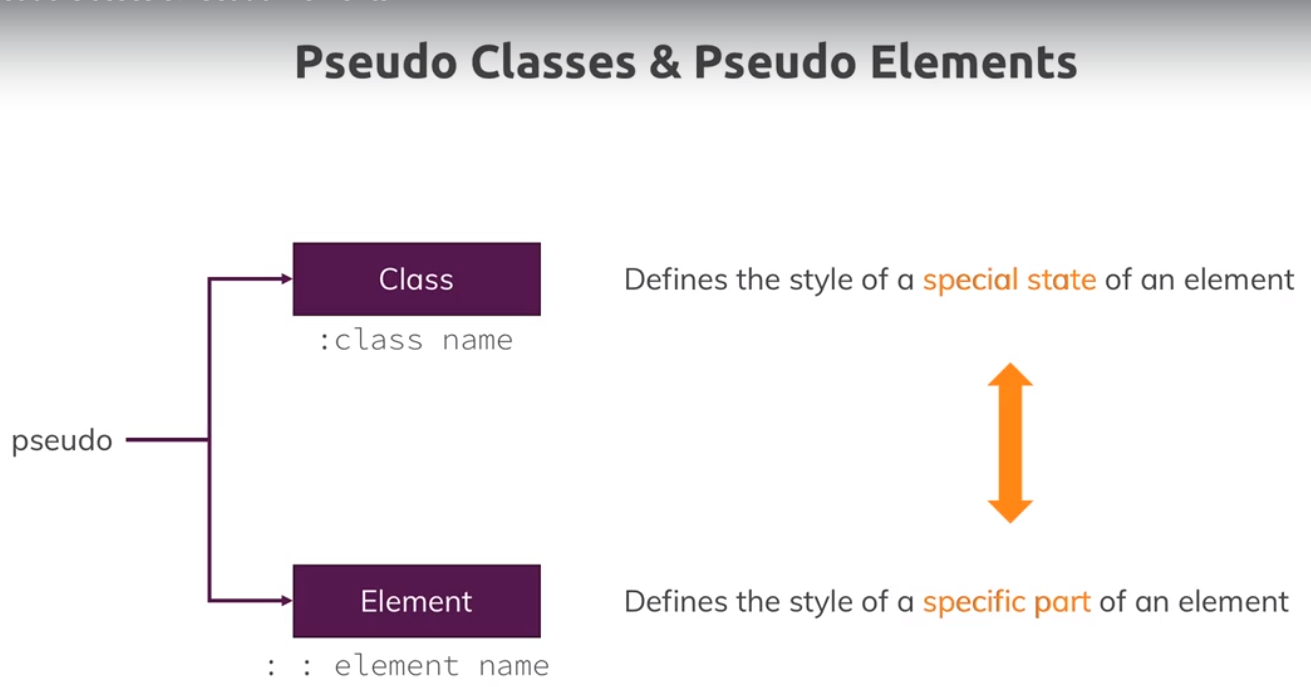
.main-nav\_\_item a:active {

color: white;

}

42)Theory time – pseudo classes and pseudo elements

We have to differentiate between pseudo classes and pseudo elements.



You can see list of all avalaible pseudo classes on mdn. There we have classes like :first-of-type, which help us to select like first p tag from list of pa tags. Feel free to dive into them, but honest we just use couple of them and that we will cover in this course. We have like :focus, which help us to style focused input. But feel free to have overview of thrm. Always keep in mind that they allows us to control different states or to be more precise about what we want to select like with first-child and first-of-type.

Then we have pseudo elements, list can be found on mdn. As explained on slide , they allow us to select parts of element. Example would be like, ::first-letter, it allows you to style first letter, like first letter of paragragh. Lets dive into them, we also have ::after, which is intersting one.

p::first-letter{

color: red;

font-size: 20px;

}

This styles the first letter.

Now interesting ones are ::after and ::before. They allow you to render content through css. And this should only be helpful for content which adds to designs not crucial page content. crucial page content should go to html, especially for accessibility reasons.

Lets use after. So we use ::after pseudo element. Now this allows us to set a special property which we can only set on after and before. This is content property. as a value it takes the string, now this is some content that will be added to DOM. then we can style that text, for example we can set color to red.

.main-nav\_\_item::after {

content: " (Link)";

color: red;

}



This can be used to render a little helper icon after text

See this link, to know more about ::after-

<https://developer.mozilla.org/en-US/docs/Web/CSS/::after>

this is also a great article about ::after and ::before, it shows we can style each property of newly added element-

<http://krasimirtsonev.com/blog/article/CSS-before-and-after-pseudo-elements-in-practice>

at the end of this aerticle, we have another links, go through them

so this is how we can use :before and : after, it can be nice to render a little helper icon after text, like this link will open in new tab, (we see it in our page).

question-

1)how to add icons with::before and ::after

<https://www.udemy.com/css-the-complete-guide-incl-flexbox-grid-sass/learn/v4/questions/6304556>

check this link-

<https://stackoverflow.com/questions/33207269/how-can-i-use-css-after-to-add-a-glyphicon-element-to-an-input/33208122#33208122>

2)Combining the pseudo slectors-

<https://www.udemy.com/css-the-complete-guide-incl-flexbox-grid-sass/learn/v4/questions/4095242>

43)grouping rules

We have these 2 selectors-

.main-nav\_\_item a:hover {

color: white;

}

.main-nav\_\_item a:active {

color: white;

}

They have same styles, so we can group them like this-

.main-nav\_\_item a:hover,

.main-nav\_\_item a:active {

color: white;

}

46)Adding & Styling CTA-button

Here we want to style a navigation link differently. Sow e add another class to it. Like this.

<nav class="main-nav">

<ul class="main-nav\_\_items">

<li class="main-nav\_\_item">

<a href="packages/index.html">uHost</a>

</li>

<li class="main-nav\_\_item">

<a href="customers/index.html">Customers</a>

</li>

<li class="main-nav\_\_item main-nav\_\_item--cta">

<a href="start-hosting/index.html">Start Hosting</a>

</li>

</ul>

</nav>

We added new class to start hosting link.now we have already defined some styles for a tags using previous class like this-

.main-nav\_\_item a {

text-decoration: none;

color: #0e4f1f;

font-weight: bold;

padding: 3px 0;

}

With new class we defined these styles-

.main-nav\_\_item--cta a {

color: white;

background: #ff1b68;

padding: 8px 16px;

border-radius: 8px;

}

Now our last a tag(see html) has both classes, new and old. So which style will be applied.Now both selectors have same selectvity, so which styles wins? Selector which is defined later in file wins.

Make sure that styles for new class selector are defiend later in file, otherwise they wnt override existing styles.

46)adding a background image

Here we added a background image to main section by using-

#product-overview {

background: url('freedom.jpg');

width: 100%;

height: 528px;

padding: 10px;

}

Here we used url helper method. It help us to refrence a image file, will be loaded as a background. As a value it takes string, which is path to image. It can be url or it can be image in local file system.

Before that we were just giving color to background like this-

background: red;

later we will see how we can position our image as we want in background and image section.

48)Wrap up

