
Session 4

Mohamed Emary

March 10, 2024

1 Review & Questions

In the first part of the session, eng. Shima reviewed the previous sessions and asked some questions.

2 Cont. CSS

2.1 Float & Clear

The `float` and `clear` section is from W3Schools So you better read it [from there](#).

2.1.1 Float

As we know from the exercise of the previous session, one of the problems with inline-block is the extra space between the elements, and to solve this issue, a possible solution is to use `float`.

`float` is a CSS property that allows an element to be taken out of the normal flow and placed along the left or right side of its container.

The `float` property is used for positioning and formatting content e.g. let an image float left to the text in a container.

The `float` property can have one of the following values:

- `left` or `right` - The element floats to the left or right of its container
- `none` - The element does not float (will be displayed just where it occurs in the text). This is default
- `inherit` - The element inherits the float value of its parent

`float` has two problems:

1. Floating elements are removed from the normal flow of the document, One of the obvious downsides of this is that the parent element no longer contains the floated element. For example if the container has a background color, it will not expand to contain the floating element.

2. The last floating element have to be cleared, otherwise it will affect the layout of the next element.

2.1.2 Clear

The `clear` property specifies on which sides of the cleared element no elements are allowed to float, it specifies what should happen with the element that is next to a floating element.

The `clear` property can have one of the following values:

- `none` (*default*) - The element is not pushed below left or right floated elements.
- `left` or `right` - The element is pushed below left or right floated elements
- `both` - The element is pushed below both left and right floated elements
- `inherit` - The element inherits the clear value from its parent

When clearing floats, you should match the clear to the float: If an element is floated to the left, then you should clear to the left.

2.2 Margin & Padding

`margin` is the space **outside** the border of an element. It is used to create space between the **element and the surrounding elements**.

`padding` is the space **inside** the border of an element. It is used to create space between the **element's border and the content**.

To give margin to an element use `margin` property in the form `margin: top right bottom left;` in clockwise order.

You can also use `margin-top`, `margin-right`, `margin-bottom`, and `margin-left` properties to specify the margin for each side separately.

In the shorthand `margin` property, if you specify one value, it will be applied to all sides, if you specify two values, the first value will be applied to the `top` and `bottom`, and the second value will be applied to the `right` and `left`.

If you ignore `left` for example it will be the same as `right`, and if you ignore `bottom` it will be the same as `top`.

All the above also apply to the `padding` property.

Sometimes we use `margin: auto;` to **center an element horizontally** (horizontally only vertical direction will not be affected).

Using both `margin-right: auto;` and `margin-left: auto;` will also center the element.

Using `margin-right: auto;` only will push the element to the left, and using `margin-left: auto` only will push it to the right.

To center items inside a div use `text-align: center;`.

The HTML page body has a default `margin` of 8px, but it can vary between browsers.

2.2.1 Margin Collapse

The top and bottom margins of blocks are sometimes combined (collapsed) into a single margin whose size is the largest of the individual margins (or just one of them, if they are equal), a

behavior known as margin collapsing.

It happens due to an empty div inside another div, and the margin of the empty div will collapse with the margin of the parent div, so you can use padding instead of margin to solve this problem, or use `overflow: auto;` in the parent div.

When we use a **percentage** in the margin or padding it will be a percentage of the **width of the parent** element. For example if you have a parent with width 500px and you have a child with `margin-top: 50%;` the margin will be 250px, however it's not recommended to use percentage with margin.

Box model in dev tools allows you to see the margin, padding, and borders of each element.

2.3 Dealing with Fonts in CSS

`font-size` property is used to specify the size of the font.

The default font size for paragraphs is 16px.

`font-size` can be specified in px, em, rem, vw, vh, vmin, vmax, or %.

When we use a percentage % with `font-size` it's a percentage of the default font size which is 16px for example 50% will be 8px.

`font-weight` ranges from 100 to 900 and default is 400. It's unitless. Some fonts don't have all the weights.

Some weights have names like `normal` which is 400, `bold` which is 700, `bolder` which is 900, and `lighter` which is 100.

`color` is used to change the color of the text. It can be a color name, a hex value, an RGB value, or an HSL value.

`font-style` specifies the style of the font. It can be `normal` (default), or `italic`.

`font-family` is used to change the shape of the font. One of the problems with this property is that if the font is not available on the user's device it will use the default font (fallback), so you can use font stack to solve this problem. You can also use fonts from google fonts or add your own font to the project files.

How to embed a font from [google fonts](#)?

1. Go to google fonts and choose the font you want.
2. Click on the + icon to add the font to the selection.
3. Click on the embed tab and copy the link tag and paste it in the head of your html file.
4. Copy the font-family and paste it in the css file.

If you choose more than one font, google fonts will give you a link tag with the two fonts in it, so you don't have to add a separate link tag for each font.

You can know that font used in a web page using the dev tools, or using a browser extension like [WhatFont](#).

2.4 Background

It can be a color or an image. A color using `background-color` and an image using `background-image: url('path to the image');`

If the image is not important for SEO add it using CSS `background-image`, as SEO don't see CSS code but if it affects the SEO use it with the HTML `` tag.

User can't save the image if it's added using CSS, but if it was added using HTML, he can save it.

`background-repeat` can be:

- `no-repeat` - The background-image will not be repeated.
- `repeat-x` - The background-image will be repeated horizontally only.
- `repeat-y` - The background-image will be repeated vertically only.
- `repeat` - The background-image will be repeated both horizontally and vertically.
- `space` - The background-image will be repeated as much as possible without clipping. The first and last images will be pinned to either side of the element, and whitespace will be distributed evenly between the images.

`background-position` can be `top`, `bottom`, `left`, `right`, `center`, or you can use the x and y coordinates.

- `background-position: center center;` this will center the image in both x and y axis.
- `background-position: top right;` this will put the image in the top right corner.
- `background-position: 50% 50%;` this will center the image in both x and y axis.
- `background-position: 50% 100%;` this will center the image in the x axis and put it in the bottom of the y axis.
- `background-position: 50px 100px;` this will put the image 50px from the left and 100px from the top.
- `background-position: 100%` the default value for the y axis is 50%.

`background-size` can be specified in pixels, percentage, or using the keywords `cover`, or `contain`.

Examples:

- `background-size: 100px 100px;` this will make the image 100px by 100px, and the image original aspect ratio will not be preserved.
- `background-size: 100% 100%;` this will make the image fit the container, and the image original aspect ratio will not be preserved.

What is the difference between cover and contain?

Both `contain` and `cover` will preserve the image original aspect ratio, however:

1. `background-size: cover`; this will make the image cover the whole container, and it will not be repeated.
2. `background-size: contain`; this will make the image fit inside the container, and it can be repeated.

`background-attachment` can be `scroll` (*default*) or `fixed`.

`vh` means viewport height, and `vw` means viewport width. These units are used to make the element take a percentage of the visible area of the screen. Each `1vw` or `1vh` is equal to 1/100 of the viewport width or height.

3 Summary

Float & Clear

- `float` is used to position an element along the left or right side of its container.
- `clear` is used to specify on which sides of an element no elements are allowed to float.

Margin & Padding

- `margin` is the space outside the border of an element
- `padding` is the space inside the border of an element.
- `margin: auto;` can be used to center an element horizontally.

Margin Collapse

Margin collapse happens when the top and bottom margins of blocks are combined into a single margin.

Dealing with Fonts in CSS

Some important font properties in CSS include:

- `font-size`
- `font-weight`
- `color`
- `font-style`
- `font-family`

How to embed a font from Google Fonts?

- Go to google fonts and choose the font you want.
- Click on the + icon to add the font to the selection.
- Click on the embed tab and copy the link tag and paste it in the head of your html file.
- Copy the font-family and paste it in the css file.

Background

- `background-color` is used to set the background color of an element.
- `background-image` is used to set the background image of an element, and it has some important properties like:
 - `background-repeat`
 - `background-position`
 - `background-size`
 - `background-attachment`.

`vh` means viewport height, and `vw` means viewport width.