1. Google fonts you would have to import for every file, there is a way though to import it to all the pages at once
2. Reason to use google fonts because only about 20 fonts are for sure downloaded on every browser and if you want to use different font it might not work on other computer, by using google fonts we overcome this issue since it get the font every time from google rather the user have the fonts on his computer or not
3. You can add video to html and you can add to the video tag autoplay, control, loop and even change the poster of the video so instead of showing the first it will show a custom image
4. Use gradient as background by googling css gradient generator, generate the gradient and copy the code onto your css file, you can change the type of ingredient like liner, circle etc.
5. Add icon in the head section <link relation=icon src=”…”> and just use and image for the icon, or you can use generator (no idea what it is, you can google it if you want)
6. In the node we can create templates instead of creating every page individually
7. Google normalize from cdn (libraries for css and js) and download the css file and use it in your project file right before the link of the regular CSS you wrote
8. You fixed position that the item will always stay in a certain position position: fixed; and you can set his left, right, top, bottom location e.i. top: 0; will be in the top of the page
9. It’s kink of pain in the neck to work with position: fixed and to place it where a places you want it to be you have to play with margin padding left right top bottom width length box-sizeing etc. to get it to be where you want it be and also you can’t move it around the page
10. You can also use position absolute it’s just like fixed that it’s totally out of the layout but instead of being fixed to the browser window he is fixed to the place in the html
11. In the case of fixed the top and bottom are relative to the window of the browser and in the case of absolute it will be place in relative to his closet parent where the position is not the default (static) if it doesn’t find any non static parent it will use the position based on the body
12. A little massage to display on the browser without JS 1. Use a div with p, with the massage “Reservation confirmed” 2. Give him a border 1px solid black Width: 20em; Height: 10em; background-color: lightblue; color: red; font-weight: bold; font-size: 1.5em text-align: center; paddint-top: 2em; box-sizing: border-box; and so on 3. Position: fixed; top: 0 left:0 and it will be in the top left corner, change the number to be in the center
13. In order to center a fixed item css is to set right: 0 left: 0; margin: auto it will be centered! (now we have nice centered confirmation box in the center of the page
14. Now let’s have a nice x bottom in the top right of the massage, 1. Download an icon 2. Change the size of the icon in pain.net to 64 by 64, add it the div of the class messageBox over the text of the message give the img class of close 3. (we can’t use Position: fixed; because it will change the position based on the size of the page) we should use position: absolute which will be position according to his first non statitc parent which is the message box
15. Image vs background image is that image is html stuff which we can’t change it in css i.e. :hover and stuff only in background image you could which is css stuff
16. Position relative is similar to position absolute but instead of being position based on it’s parent it will use the positon based on himself, and beside now he is not position; static which mean his childrent will be able to be position based on his own position because you can set the child for position absolute and will be position based on the טעטע which is not static
17. You can hide and show an element by using visibility: visible/hidden; another way of doing it is by using display: none; the different is that vissibleaity does not take the element of the layout just make it not visible but display non takes out the element from the layout, and the structure of the page changes accordingly, in other words display: noe makes the element as if it doesn’t exists and visibility: hidden doesn’t change anything in the layout and only make the element not visible, both are not relevant until we will learn JS
18. To change the css based on display size for example change the size of the video we were using class ago so when the screen get small enough instead of being 3 videos per row it will be 1, so in the CSS you can tag specific css to only be apply at a certain condition (when size of window is less than 800px for example by writing @media screen and (max-width: 799px) video {width: 100%;} (instead of 33.3% like it was before)
19. A good way to code is to code for mobile first so first css will be for phone and to write that in the case of bigger screen it will be different, so in our case the video will be 100% size, meaning 1 video per row and in the case of screen bigger than 800px the width of the video will be 33%, meaning 3 video per row, since most people use website on the phone (google why if you really board, yes shochet I am talking to you)
20. Using print in the browser will default that when ever you print the page, it will use the default CSS but you could use @media printer to style the page according to how it fit most with printing pages you can search google if you want, so if your website is heavily used to print stuff that it might be worth while to work on style to page differently when it’s printed, for example we can set video {display: non} to make it that when you print it wouldn’t print the unnecessary video
21. When we will we use bootstrap they are realy focus on making the site look beautiful based on the device you are using
22. We can say to only load the css file at a certain condition (width > 800px) again google it
23. 1. \* --> universal selector 2. Type selector i.e. header {font-size: 10px} 3. Id selector i.e. id=”hi” #hi {color: white} 4. Class = “special”, . special {text-align: center} 5. Combine few stuff together I.e. p, #hi, . special {text-align: center}, 6. Section that has class special section.special 7. Descented selector p inside a section -- > selection p {text-align: center} 8. Section \* will mean every thing inside section, (I am so tired) (oh the song of motty shteiments is stuck inside my head (the styele will not effect the selection itself only his קינדרלעך and even grandchildren, and if you want to only apply css the child only and not grandchildren use section>p 9. Sibling selection selection+p will only apply the css to the p that a immediately follow a selection 10. Section~p will apply to all p which follow the section in the same parent even if not immediately after 11. Pseudo elements to target .special::first-letter to style the first letter of section, you can also target first-line, there is no reason to write down all the selector so I am going to stop right here, in any case I am latterly falling asleep now