**Visual Studio Code: Kujunda elektroonilise kella näide vastavalt maitsele või kindlale teemale, mahutades kella täisekraanile, et saaks kasutada lauakella või ekraanisäästja asemel. Selleks, et see sobiks paljudele ekraanidele, kasuta kujunduse loomisel protsendilisi väärtusi (nt width: 100%; ) või nt võimalda kella suurust kasutajal muuta.**

Selleks, et kujundada elektroonilist kella Visual Studio Code'is, tuleb kasutada veebirakenduste arendamiseks mõeldud tööriistu, näiteks HTML, CSS ja JavaScript. Siin on mõned sammud, mida võid järgida:

1. Loome HTML faili, kus kirjutame kella jaoks vajaliku struktuuri. Näiteks:

phpCopy code

<div class="clock"> <div class="hours"> </div> <div class="minutes"> </div> <div class="seconds"> </div> </div>

1. Kujundame kella CSS abil, kasutades protsendilisi väärtusi, et kohandada kella suurust vastavalt ekraani suurusele. Näiteks:

cssCopy code

.clock { width: 100%; height: 100%; display: flex; justify-content: center; align-items: center; font-size: 5em; }

1. Kirjutame JavaScripti koodi, mis võimaldab kella käivitada ja töötab värskendades kella iga sekundit vastavalt kasutaja ajatsooni seadistusele. Näiteks:

javascriptCopy code

function updateClock() { const currentTime = new Date(); let currentHours = currentTime.getHours(); let currentMinutes = currentTime.getMinutes(); let currentSeconds = currentTime.getSeconds(); currentHours = (currentHours < 10 ? "0" : "") + currentHours; currentMinutes = (currentMinutes < 10 ? "0" : "") + currentMinutes; currentSeconds = (currentSeconds < 10 ? "0" : "") + currentSeconds; const timeOfDay = (currentHours < 12) ? "AM" : "PM"; document.querySelector(".hours").innerHTML = currentHours; document.querySelector(".minutes").innerHTML = currentMinutes; document.querySelector(".seconds").innerHTML = currentSeconds; } setInterval(updateClock, 1000);

1. Lisame HTML-le JavaScripti faili ja alustame kella käivitamist. Näiteks:

phpCopy code

<script src="clock.js"></script> <script> updateClock(); </script>

Lõpuks saame salvestada failid ja avada HTML-faili Visual Studio Code'is, et näha, kuidas kella välja näeb. Kui soovite, võite lisada kasutajaliidese elemendid, mis võimaldavad kasutajal kella suurust muuta või valida erinevaid teemasid.

**Kas html kood on õige: <!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta http-equiv="X-UA-Compatible" content="IE=edge"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <link rel="stylesheet" href="kodukell.css"> <script src="kodukell.js" defer></script> <title>KODUKELL</title> </head> <body> <div class="clock"> <div class = "hours"></div> <div class = "minutes"></div> <div class = "seconds"></div> </div> <script> updateClock(); </script> </body> </html>**

Jah, see HTML-kood on õige. See loob veebilehe, millel on metaandmed, HTML-lehtede põhiline struktuur, lingid stiililehtede ja JavaScripti-failide jaoks ning veebilehe sisu. Veebilehel on ka üks **div**-element, mis sisaldab kella loomiseks kolme alam-elementi **hours**, **minutes** ja **seconds**. Viimane rida kutsub JavaScripti funktsiooni **updateClock()** välja, mis võimaldab kella pidevalt värskendada.

**Kas javascripti kood on õige: function updateClock(){ const currentTime = newDate(); let currentHours = currentTime.getHours(); let currentMinutes = currentTime.getMinutes(); let currentSeconds = currentTime.getSeconds(); currentHours = (currentHours < 10 ? "0" : "") + currentHours; currentMinutes = (currentMinutes < 10 ? "0" : "") + currentMinutes; currentSeconds = (currentSeconds < 10 ? "0" : "") + currentSeconds; const timeOfDay = (currentHours < 12) ? "AM" : "PM"; document.querySelector(".hours").innerHTML = currentHours; document.querySelector(".minutes").innerHTML = currentMinutes; document.querySelector(".seconds").innerHTML = currentSeconds; } setInterval(updateClock, 1000);**

Peaaegu õige. Funktsiooni esimeses reas tuleks aga asendada **newDate()** **new Date()**-ga, et uus kuupäevaobjekt korrektselt luuakse.

Seega õige kood peaks olema:

javascriptCopy code

function updateClock(){ const currentTime = new Date(); let currentHours = currentTime.getHours(); let currentMinutes = currentTime.getMinutes(); let currentSeconds = currentTime.getSeconds(); currentHours = (currentHours < 10 ? "0" : "") + currentHours; currentMinutes = (currentMinutes < 10 ? "0" : "") + currentMinutes; currentSeconds = (currentSeconds < 10 ? "0" : "") + currentSeconds; const timeOfDay = (currentHours < 12) ? "AM" : "PM"; document.querySelector(".hours").innerHTML = currentHours; document.querySelector(".minutes").innerHTML = currentMinutes; document.querySelector(".seconds").innerHTML = currentSeconds; } setInterval(updateClock, 1000);

See funktsioon loob uue kuupäevaobjekti, võtab sellest hetkeaja tundi, minuteid ja sekundeid, lisab need sobivalt 0-ga, kui need on kahekohalised ja määrab päeva- või ööaja. Seejärel värskendab see kella HTML-i kaudu, muutes vastavate elementide sisu. Lõpuks käivitatakse funktsioon **updateClock** iga sekundi järel, kasutades funktsiooni **setInterval**.

**Visual Studio Code: Kujunda elektroonilise kella näide vastavalt maitsele või kindlale teemale, mahutades kella täisekraanile, et saaks kasutada lauakella või ekraanisäästja asemel. Selleks, et see sobiks paljudele ekraanidele, kasuta kujunduse loomisel protsendilisi väärtusi (nt width: 100%; ) või nt võimalda kella suurust kasutajal muuta. Kasuta eventListener-e'**

Siin on näide, kuidas elektroonilist kella saab kohandada vastavalt kasutaja eelistustele ja kasutada täisekraanil lauakellana või ekraanisäästjana. Kujunduse loomisel kasutatakse protsendilisi väärtusi ja kasutajal on võimalik kella suurust muuta nuppu klõpsates.

HTML:

phpCopy code

<!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta http-equiv="X-UA-Compatible" content="IE=edge"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <link rel="stylesheet" href="style.css"> <title>Clock</title> </head> <body> <div class="clock-container"> <div class="clock"> <div class="hours"></div> <div class="minutes"></div> <div class="seconds"></div> </div> <button id="resize-button">Resize Clock</button> </div> <script src="script.js"></script> </body> </html>

CSS:

cssCopy code

body { margin: 0; padding: 0; background-color: black; } .clock-container { display: flex; justify-content: center; align-items: center; height: 100vh; } .clock { display: flex; flex-direction: column; justify-content: center; align-items: center; width: 100%; height: 100%; font-size: 10vw; color: white; } button { position: absolute; top: 20px; right: 20px; padding: 10px 20px; font-size: 1.2rem; border: none; background-color: white; color: black; }

JavaScript:

javascriptCopy code

const clock = document.querySelector('.clock'); const resizeButton = document.querySelector('#resize-button'); function updateClock() { const currentTime = new Date(); let currentHours = currentTime.getHours(); let currentMinutes = currentTime.getMinutes(); let currentSeconds = currentTime.getSeconds(); currentHours = (currentHours < 10 ? "0" : "") + currentHours; currentMinutes = (currentMinutes < 10 ? "0" : "") + currentMinutes; currentSeconds = (currentSeconds < 10 ? "0" : "") + currentSeconds; const timeOfDay = (currentHours < 12) ? "AM" : "PM"; document.querySelector(".hours").innerHTML = currentHours; document.querySelector(".minutes").innerHTML = currentMinutes; document.querySelector(".seconds").innerHTML = currentSeconds; } setInterval(updateClock, 1000); resizeButton.addEventListener('click', function() { clock.classList.toggle('large'); });

Kasutatakse kahte eventListenerit - **setInterval** funktsiooni, mis uuendab kella iga sekundi järel, ja nupuvajutuse eventListenerit, mis käivitab funktsiooni **toggle**, mis lisab või eemaldab klassi **large** kellale vastavalt kasutaja soovile. Klass **large** on määratud CSS-is ja see muudab kella suurust.

CSS-is kasutatakse protsendilisi väärtusi, et muuta kella vastavalt erinevatele ekra

**Kuidas saada tundide, sekundite ja minutite vahele koolon? function updateClock(){ const currentTime = new Date(); let currentHours = currentTime.getHours(); let currentMinutes = currentTime.getMinutes(); let currentSeconds = currentTime.getSeconds(); currentHours = (currentHours < 10 ? "0" : "") + currentHours; currentMinutes = (currentMinutes < 10 ? "0" : "") + currentMinutes; currentSeconds = (currentSeconds < 10 ? "0" : "") + currentSeconds; const timeOfDay = (currentHours < 12) ? "AM" : "PM"; document.querySelector(".hours").innerHTML = currentHours + ":"; document.querySelector(".minutes").innerHTML = ":" + currentMinutes; document.querySelector(".seconds").innerHTML = ":" + currentSeconds; } setInterval(updateClock, 1000);**

Saate lisada kooloni tundide ja minutite vahele, asendades järgmise rea:

javascriptCopy code

document.querySelector(".hours").innerHTML = currentHours + ":"; document.querySelector(".minutes").innerHTML = ":" + currentMinutes;

järgmise reaga:

javascriptCopy code

document.querySelector(".hours").innerHTML = currentHours + ":"; document.querySelector(".minutes").innerHTML = ":" + currentMinutes + ":";

See lisab kooloni minutite ja sekundite vahele. Lõplik kood koos kooloniga oleks järgmine:

javascriptCopy code

function updateClock(){ const currentTime = new Date(); let currentHours = currentTime.getHours(); let currentMinutes = currentTime.getMinutes(); let currentSeconds = currentTime.getSeconds(); currentHours = (currentHours < 10 ? "0" : "") + currentHours; currentMinutes = (currentMinutes < 10 ? "0" : "") + currentMinutes; currentSeconds = (currentSeconds < 10 ? "0" : "") + currentSeconds; const timeOfDay = (currentHours < 12) ? "AM" : "PM"; document.querySelector(".hours").innerHTML = currentHours + ":"; document.querySelector(".minutes").innerHTML = ":" + currentMinutes + ":"; document.querySelector(".seconds").innerHTML = ":" + currentSeconds; } setInterval(updateClock, 1000);

**Kuidas kuvada kuupäeva? function updateClock(){ const currentTime = new Date(); let currentHours = currentTime.getHours(); let currentMinutes = currentTime.getMinutes(); let currentSeconds = currentTime.getSeconds(); currentHours = (currentHours < 10 ? "0" : "") + currentHours + ":"; currentMinutes = (currentMinutes < 10 ? "0" : "") + currentMinutes + ":"; currentSeconds = (currentSeconds < 10 ? "0" : "") + currentSeconds; const timeOfDay = (currentHours < 12) ? "AM" : "PM"; document.querySelector(".hours").innerHTML = currentHours + ":"; document.querySelector(".minutes").innerHTML = ":" + currentMinutes + ":"; document.querySelector(".seconds").innerHTML = ":" + currentSeconds; } function upDateDate(){ d = new Date(); let date = d.getDate(); let month = d.getMonth(); let year = d.getFullYear(); let day = d.getDay(); if(date < 10){ date = "0" + date; } console.log(date, month, year); dayContainer.innerHTML = days[day]; dateContainer.innerHTML = date + ". " + months[month] + " " + year; } updateClock() upDateDate(); setInterval(updateClock, 1000);**

Teie praegune kood sisaldab juba funktsiooni **upDateDate**, mis võimaldab kuupäeva kuvada. Funktsioonis kasutatakse **new Date()** funktsiooni, et saada hetke kuupäev ja seejärel vormindatakse kuupäeva tervikuna stringiks, kasutades sobivaid kuupäevafunktsioone.

Siiski tuleb teil HTML-is luua kaks uut elemendikonteinerit, üks kuupäeva jaoks ja teine päeva jaoks, et kuvada kuupäev ja päev vastavalt teie soovile. Näiteks võiksite lisada järgmise HTML-koodi:

phpCopy code

<div class="date-container"></div> <div class="day-container"></div>

Siis muutke funktsiooni **upDateDate** nii, et see kasutaks **document.querySelector()** meetodit uute elemendikonteinerite leidmiseks, ja seadistaks vastava sisu teksti:

vbnetCopy code

function upDateDate(){ const days = ["Pühapäev", "Esmaspäev", "Teisipäev", "Kolmapäev", "Neljapäev", "Reede", "Laupäev"]; const months = ["jaanuar", "veebruar", "märts", "aprill", "mai", "juuni", "juuli", "august", "september", "oktoober", "november", "detsember"]; const d = new Date(); const dateContainer = document.querySelector(".date-container"); const dayContainer = document.querySelector(".day-container"); let date = d.getDate(); let month = d.getMonth(); let year = d.getFullYear(); let day = d.getDay(); if(date < 10){ date = "0" + date; } dayContainer.innerHTML = days[day]; dateContainer.innerHTML = date + ". " + months[month] + " " + year; }

Lõpuks kutsuge funktsioon **upDateDate()** välja koos funktsiooniga **updateClock()** järgmise koodiga:

scssCopy code

updateClock() upDateDate(); setInterval(updateClock, 1000);

**Kuidas saab CSS koodiga HTML lehe taustavärvi muuta?**

HTML lehe taustavärvi muutmiseks võite kasutada CSS-i **background-color** omadust. See võimaldab teil määrata mis tahes värviga taustavärvi. Siin on mõned näited:

Ühtse värviga taust:

cssCopy code

body { background-color: red; }

Gradient taustavärv:

cssCopy code

body { background: linear-gradient(to bottom right, #ff0066, #6600ff); }

Pildi taustavärv:

cssCopy code

body { background-image: url("pilt.jpg"); }

Lisaks saate kasutada ka teisi CSS-i omadusi, nagu **background-repeat**, **background-size**, **background-position** jne, et täpsemalt kohandada taustavärvi kuvamist vastavalt oma vajadustele.

**Kas CSS-i "background-color " peab välja tooma ka HTML-is või JavaScriptis?**

CSS-i **background-color** omadust ei ole vaja HTML-is ega JavaScriptis välja tuua. Seda saab kasutada otse CSS-i stiililehe või stiilibloki sees, et määrata HTML elemendi taustavärv.

Siin on näide, kuidas CSS-i **background-color** omadust saab kasutada HTML elemendi taustavärvi määramiseks:

htmlCopy code

<!DOCTYPE html> <html> <head> <title>Minu HTML lehekülg</title> <style> body { background-color: #f2f2f2; } </style> </head> <body> <h1>Tere tulemast minu lehele</h1> <p>Siin on minu lehe sisu</p> </body> </html>

Sel juhul on HTML lehe taustavärv määratud CSS-i stiililehe sees, kasutades **body** selektorit ja **background-color** omadust. HTML-is pole selleks vaja midagi eraldi kirjutada.

**Kuidas muuta teksti värvi: .clock{ width: 100%; height: 100%; display: flex; justify-content: space-around; align-items: center; font-size: 5em; background-color: lightgray; text-decoration-color: crimson; }**

Teksti värvi muutmiseks saate kasutada CSS-i omadust **color**. See omadus määrab HTML elemendi teksti värvi. Näiteks, kui soovite muuta **.clock** elemendi teksti värvi punaseks, võite selle CSS-i reegli lisada:

cssCopy code

.clock{ width: 100%; height: 100%; display: flex; justify-content: space-around; align-items: center; font-size: 5em; background-color: lightgray; color: crimson; }

Siin on lisatud uus rida **color: crimson;**, mis määrab teksti värvi punaseks. Saate kasutada mistahes CSS-i toetatud värviväärtust, nagu näiteks nime, HEX-koodi või RGB-väärtust, et määrata teksti värv.

**Stylesheet CSS ei toimi. Kas kõik õige?**

**HTML:**

**<!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta http-equiv="X-UA-Compatible" content="IE=edge"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <link rel="stylesheet" href="kodukell.css"> <script src="kodukell.js" defer></script> <title>KODUKELL</title> </head> <body> <div class="clock"> <div class = "hours">Tunnid</div> <div class = "minutes">Minutid</div> <div class = "seconds">Sekundid</div> </div> <div class="date-container"> Kuupäev </div> <div class="day-container"> Päev </div> <script> updateClock(); </script> <style> body{ background-color: darksalmon; } </style> </body> </html>**

**CSS:**

**.clock{ width: 100%; height: 100%; display: flex; justify-content: space-around; align-items: center; font-size: 10em; background-color: lightgray; color: crimson; }**

**JavaScript:**

**function updateClock(){ const currentTime = new Date(); let currentHours = currentTime.getHours(); let currentMinutes = currentTime.getMinutes(); let currentSeconds = currentTime.getSeconds(); currentHours = (currentHours < 10 ? "0" : "") + currentHours + ":"; currentMinutes = (currentMinutes < 10 ? "0" : "") + currentMinutes + ":"; currentSeconds = (currentSeconds < 10 ? "0" : "") + currentSeconds; const timeOfDay = (currentHours < 12) ? "AM" : "PM"; document.querySelector(".hours").innerHTML = currentHours ; document.querySelector(".minutes").innerHTML = currentMinutes; document.querySelector(".seconds").innerHTML = currentSeconds; } function upDateDate(){ const months = ["Jaanuar", "Veebruar", "Märts", "Aprill", "Mai", "Juuni", "Juuli"] const days = ["Pühapäev", "Esmaspäev", "Teisipäev", "Kolmapäev", "Neljapäev", "Reede", "Laupäev"] const d = new Date(); const dateContainer = document.querySelector(".date-container"); const dayContainer = document.querySelector(".day-container"); let date = d.getDate(); let month = d.getMonth(); let year = d.getFullYear(); let day = d.getDay(); if(date < 10){ date = "0" + date; } dayContainer.innerHTML = days[day]; dateContainer.innerHTML = date + ". " + months[month] + " " + year; } updateClock() upDateDate(); setInterval(updateClock, 1000);**

CSS-stiilid tunduvad olevat õigesti rakendatud. Kuid, kuna kella ja kuupäeva kuvamise elemendid on määratletud klassidega **.clock**, **.date-container** ja **.day-container**, siis peaks CSS-i reeglid olema rakendatud nende klasside järgi, mitte üldisele **body** elemendile.

Seetõttu peaks CSS-i reeglid olema lisatud eraldi faili (nagu antud juhul on **kodukell.css**) või **<style>** elemendi sisse (aga mitte **body** elemendi sisse). Näiteks, kui soovite kella taustavärvi muuta, peaks CSS välja nägema järgmiselt:

cssCopy code

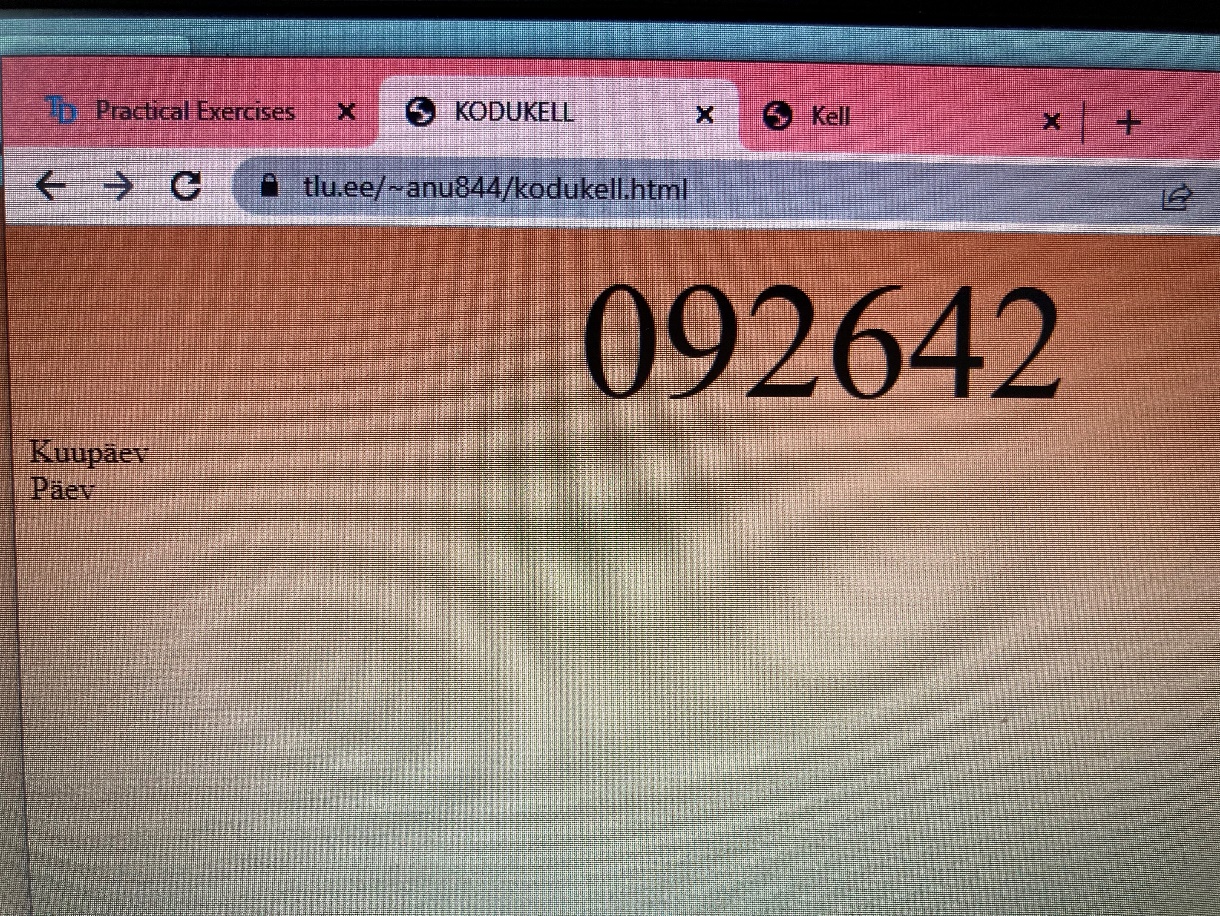
.clock { width: 100%; height: 100%; display: flex; justify-content: space-around; align-items: center; font-size: 10em; background-color: darksalmon; color: crimson; }

Sellisel juhul muudetakse kella taustavärvi "lightsalmon" asemel "darksalmon"-iks.

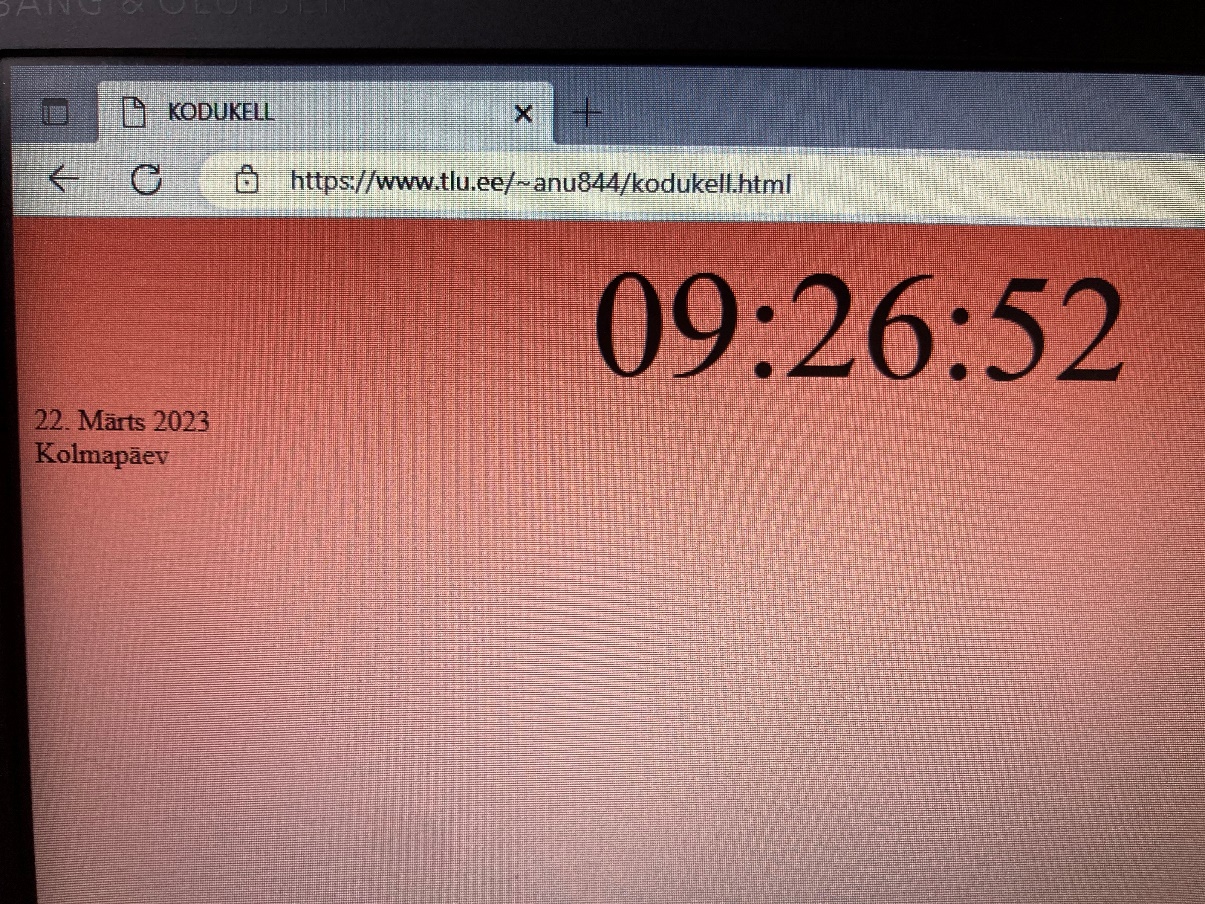
**SIIN SAIN ARU, ET MINU VEEBIBRAUSERID EI TEE KOOSTÖÖD EGA KUVA MULLE TEHTUD MUUDATUSI. EDASI VAATASIN OMA MUUDATUSI TELEFONIST, KUS KÕIK KOHE TOIMIS. NÜÜD AGA VÄIKE TAGASILÖÖK, KUIDAS LISADA EVENTLISTENERE, KUI AINULT TELEFONIS LEHT TÖÖTAB… TULEB NATUKE MÕELDA JA VÕIB-OLLA PAUS TEHA.**

LISAN KA PILDID HETKEL NÄHTAVATEST VARIANTIDEST:

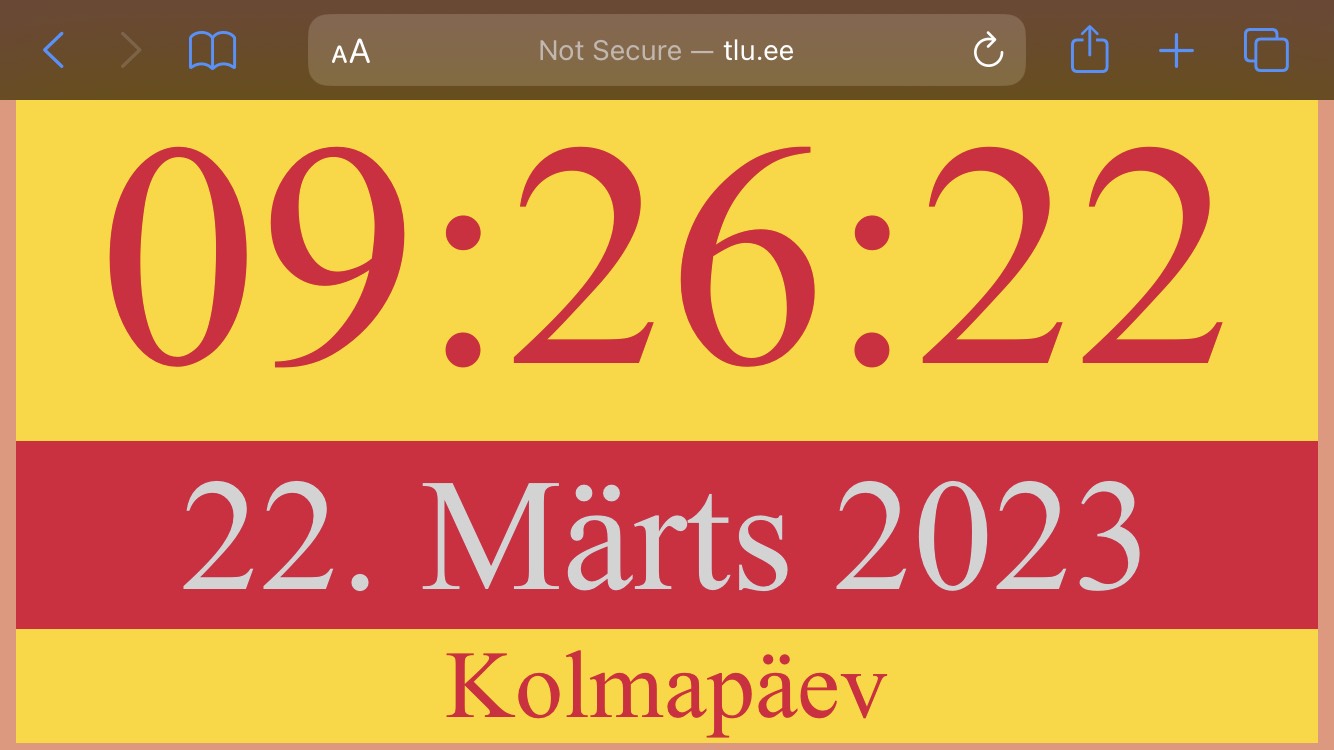
CHROME:



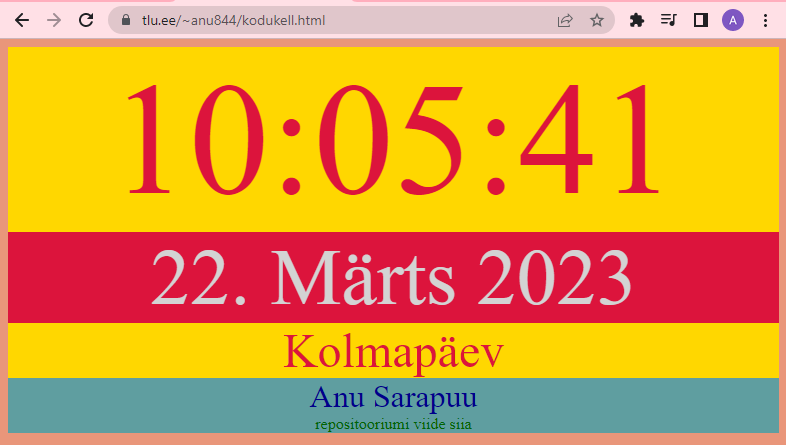
MICROSOFT EDGE:



IPHONE:



**UPDATE:** Chrome otsustas koostööd tegema hakata!



**EDASI UUED CHATGPT VESTLUSED:**

**"Button" ei muuda clock suurust veebilehel. Mis on valesti? HTML: <!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta http-equiv="X-UA-Compatible" content="IE=edge"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <link rel="stylesheet" href="kodukell.css"> <script src="kodukell.js" defer></script> <title>KODUKELL</title> </head> <body> <div class="clock"> <div class = "hours">Tunnid</div> <div class = "minutes">Minutid</div> <div class = "seconds">Sekundid</div> </div> <div class="date-container"> Kuupäev </div> <div class="day-container"> Päev </div> <div class="author"> Anu Sarapuu </div> <div class="repo"> repositooriumi viide siia </div> <button id="resize-button">Muuda kella suurust</button> <script> updateClock(); </script> <style> body{ background-color: darksalmon; } </style> </body> </html> CSS: .clock{ width: 100%; height: 100%; display: flex; justify-content: center; align-items: center; font-size: 1em; background-color: gold; color: crimson; } .date-container{ display: flex; justify-content: center; align-items: center; font-size: 5em; background-color: crimson; color: lightgrey; } .day-container{ display: flex; justify-content: center; align-items: center; font-size: 3em; background-color: gold; color: crimson; } .author{ display: flex; justify-content: center; align-items: center; font-size: 2em; background-color: cadetblue; color: darkblue; } .repo{ display: flex; justify-content: center; align-items: center; font-size: 1em; background-color: cadetblue; color: darkgreen; } button{ display: flex; justify-content: center; align-items: center; top: 20px; right: 20px; padding: 10px 20px; font-size: 1em; border: none; background-color: lightyellow; color: black; } JAVASCRIPT: const clock = document.querySelector('.clock'); const resizeButton = document.querySelector('#resize-button'); function updateClock(){ const currentTime = new Date(); let currentHours = currentTime.getHours(); let currentMinutes = currentTime.getMinutes(); let currentSeconds = currentTime.getSeconds(); currentHours = (currentHours < 10 ? "0" : "") + currentHours + ":"; currentMinutes = (currentMinutes < 10 ? "0" : "") + currentMinutes + ":"; currentSeconds = (currentSeconds < 10 ? "0" : "") + currentSeconds; const timeOfDay = (currentHours < 12) ? "AM" : "PM"; document.querySelector(".hours").innerHTML = currentHours ; document.querySelector(".minutes").innerHTML = currentMinutes; document.querySelector(".seconds").innerHTML = currentSeconds; } function upDateDate(){ const months = ["Jaanuar", "Veebruar", "Märts", "Aprill", "Mai", "Juuni", "Juuli"] const days = ["Pühapäev", "Esmaspäev", "Teisipäev", "Kolmapäev", "Neljapäev", "Reede", "Laupäev"] const d = new Date(); const dateContainer = document.querySelector(".date-container"); const dayContainer = document.querySelector(".day-container"); let date = d.getDate(); let month = d.getMonth(); let year = d.getFullYear(); let day = d.getDay(); if(date < 10){ date = "0" + date; } dayContainer.innerHTML = days[day]; dateContainer.innerHTML = date + ". " + months[month] + " " + year; } updateClock() upDateDate(); setInterval(updateClock, 1000); resizeButton.addEventListener('click', function(){ clock.classList.toggle('large'); });**

Based on the provided code, it seems that the issue is related to the CSS code for the clock element. You have defined the clock class with a height and width of 100%, which makes it take up the entire available space on the webpage. When you add the "large" class by clicking the "Muuda kella suurust" button, it does not change the size of the clock because the height and width are already set to 100%.

To fix this issue, you can add some new CSS rules to the ".large" class to change the size of the clock when the button is clicked. For example, you can change the width and height properties to a fixed size, like this:

cssCopy code

.large { width: 500px; height: 500px; }

Of course, you can adjust the size to whatever value you want. After making this change, the clock element should resize when the button is clicked.

**<button id="resize-button"> Muuda kella suurust väiksemaks/suuremaks </button> Kuidas saab "väiksemaks/suuremaks" veebilehel järgmisele reale?**

Selleks saab kasutada HTML-i **<br>** elemendi, mis lisab reavahetuse. Muutes nupu koodi järgnevalt:

bashCopy code

<button id="resize-button"> Muuda kella suurust<br>väiksemaks/suuremaks </button>

Sellisel juhul kuvatakse "väiksemaks/suuremaks" uuel real.