

Kujunda elektrooniline kell, kasutades html, css ja javascript keeli. Kujunda kell nii, et see sobiks igale ekraani suurusele. Näita ekraanil Tallinna kellaaega, selle all nädalapäeva ja kuupäeva.

Vabandust, kuid mina ei saa HTML, CSS ja JavaScripti kasutades visuaalseid elemente luua, kuna olen tekstipõhine tehisintellekt. Kuid ma saan teile pakkuda üldist struktuuri ja lähenemistele, mida saate kasutada kellaprojekti loomiseks.

HTML-i struktuur:

```
<!DOCTYPE html> <html>
  <head>
    <title>Minu kell</title>
  </head>
  <body>
    <div class="clock">
      <div class="time"></div>
      <div class="date"></div>
    </div>
    <script src="script.js"></script>
  </body>
</html>
```

CSS-i stiil:

```
html, body {
  height: 100%; }
.clock {display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  height: 100%; }
.time {font-size: 5em;
  font-weight: bold;
  margin-bottom: 10px; }
.date { font-size: 2em; }
```

JavaScript-i kood:

```
function updateTime() {
  let today = new Date();
  let time = today.toLocaleTimeString('et-EE');
  let date = today.toLocaleDateString('et-EE', { weekday: 'long', year: 'numeric', month: 'long', day: 'numeric' });
  document.querySelector('.time').textContent = time;
  document.querySelector('.date').textContent = date; }
setInterval(updateTime, 1000);
```

JavaScript-i koodi selgitus:

- Funktsioon updateTime loob uue Date objekti, mis esindab praegust aega.
- Seejärel kasutatakse kahte meetodit toLocaleTimeString ja toLocaleDateString, et saada kohalik aeg ja kuupäev vastavalt Eesti ajavööndile ja keelele.
- Lõpuks kasutatakse textContent meetodit, et uuendada HTML-i sisu vastavalt praegusele ajale ja kuupäevale.

- setInterval funktsiooni kasutatakse selleks, et funktsioon updateTime käivituks iga sekundi tagant, nii et kuvatav kell näitaks alati õiget aega.

Kui olete loonud HTML-, CSS- ja JavaScript-failid, võite need koos panna ja käivitada veebibrauseris, et näha, kuidas teie kell töötab. Kui soovite kohandada kelladisaini või lisada muid funktsioone, võite HTML-, CSS- ja JavaScript-koodi vastavalt oma soovidele muuta.

nüüd lisa koodile veel 2 erinevat ajatsooni - New York ja Tokyo

Siin on muudetud kood, mis lisab lisaks Tallinna ajale ka New Yorki ja Tokyo ajad: HTML-i struktuur ja CSS stiil jäävad samaks nagu eelmises vastuses:

```
<!DOCTYPE html>
<html>
<head>
<title>Minu kell</title>
</head>
<body>
<div class="clock">
<div class="time tallinn"></div>
<div class="date"></div>
<div class="time new-york"></div>
<div class="time tokyo"></div>
</div> <script src="script.js"></script>
</body>
</html>
```

```
html, body {height: 100%; }
.clock {display: flex;
flex-direction: column;
align-items: center;
justify-content: center;
height: 100%; }
.time { font-size: 5em;
font-weight: bold;
margin-bottom: 10px; }
.tallinn { color: blue; }
.new-york {color: green; }
.tokyo {color: red; }
.date { font-size: 2em; }
```

JavaScript-kood, mis lisab New Yorgi ja Tokyo ajad:

```
function updateTime() {
let today = new Date();
// Tallinn
let tallinnTime = today.toLocaleTimeString('et-EE');
document.querySelector('.tallinn').textContent = tallinnTime;
// New York
```

```

let newYorkTime = today.toLocaleTimeString('en-US', { timeZone:
'America/New_York' });
document.querySelector('.new-york').textContent = newYorkTime;
// Tokyo
let tokyoTime = today.toLocaleTimeString('ja-JP', { timeZone: 'Asia/Tokyo' });
document.querySelector('.tokyo').textContent = tokyoTime;
let date = today.toLocaleDateString('et-EE', { weekday: 'long', year: 'numeric', month:
'long', day: 'numeric' });
document.querySelector('.date').textContent = date; } setInterval(updateTime, 1000);

```

JavaScript-i koodi selgitus:

- Funktsioon updateTime loob uue Date objekti, mis esindab praegust aega.
- Kasutatakse kolme erinevat meetodit toLocaleTimeString, et saada kohalik aeg vastavalt Tallinnale, New Yorkile ja Tokyole.
- Iga ajatsooni jaoks määratakse klassi järgi vastav värv.
- Lõpuks kasutatakse textContent meetodit, et uuendada HTML-i sisu vastavalt praegustele aegadele ja kuupäevale.
- setInterval funktsiooni kasutatakse selleks, et funktsioon updateTime käivituks iga sekundi tagant, nii et kuvatav kell näitaks alati õiget aega.

how to add an eventlistener for when the user clicks on the screen for changing the background color

You can add an event listener to the document object to listen for the "click" event, and then change the background color of the body element. Here's an example:

```

// add event listener to document for "click" event
document.addEventListener("click", function() {
  // select the body element
  var body = document.querySelector("body");

  // change the background color to a random color
  body.style.backgroundColor = getRandomColor();
});

```

```

// function to generate a random color
function getRandomColor() {
  var letters = "0123456789ABCDEF";
  var color = "#";
  for (var i = 0; i < 6; i++) {
    color += letters[Math.floor(Math.random() * 16)];
  }
  return color;
}

```

In this example, the addEventListener method is used to listen for the "click" event on the document object. When the event is triggered, the getRandomColor function is called to generate a random color, and the background color of the body element is set to this color using the style.backgroundColor property. The getRandomColor function generates a random hexadecimal color code by selecting random characters from the string "0123456789ABCDEF".

i want it to have only 2 options: white background and black text OR black background and white text

You can achieve this by toggling between two CSS classes when the user clicks on the screen. One class can define a white background and black text, while the other class can define a black background and white text. Here's an example:

HTML:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Toggle Background Color</title>
  <style>
    /* Define styles for white background and black text */
    .white-background {
      background-color: white;
      color: black;
    }

    /* Define styles for black background and white text */
    .black-background {
      background-color: black;
      color: white;
    }
  </style>
</head>
<body class="white-background">
  <h1>Click anywhere to toggle background color</h1>
  <script src="script.js"></script>
</body>
</html>
```

JavaScript:

```
// Select the body element
var body = document.querySelector("body");

// Add event listener for "click" event
body.addEventListener("click", function() {
  // Toggle between "white-background" and "black-background" classes
  body.classList.toggle("white-background");
  body.classList.toggle("black-background");
});
```

In this example, the body element is selected using document.querySelector. The event listener is added to the body element for the "click" event. When the user clicks on the screen, the classList.toggle method is used to toggle between the white-background and black-background classes. These classes define the styles for the white and black

background, and black and white text, respectively. When the user clicks on the screen again, the classes are toggled back to their original state.

how to play music in the background of a webpage

To play music in the background of a webpage, you can use the HTML5 <audio> element. Here is an example of how to use it:

```
<audio src="your_music_file.mp3" autoplay loop></audio>
```

In this example, src is the path to your music file. autoplay will start playing the music automatically when the page loads, and loop will make the music repeat continuously. You can also add other attributes to the <audio> element, such as controls to show playback controls, or preload to specify when the browser should start loading the audio file.

Keep in mind that playing music automatically can be annoying for some users, so it's a good practice to provide a way to turn off or control the music playback, such as adding a mute or pause button.

how to add functions for the audio - change volume, pause audio

To change the volume of an audio element in JavaScript, you can access the volume property of the audio element and set it to a value between 0 and 1. For example, to set the volume to half, you can use:

```
let audio = document.querySelector('audio');  
audio.volume = 0.5; // sets volume to 50%
```

To pause an audio element, you can use the pause() method. For example:

```
let audio = document.querySelector('audio');  
audio.pause(); // pauses the audio
```

You can also check the paused property of the audio element to see if it is currently paused. For example:

```
let audio = document.querySelector('audio');  
if (audio.paused) {  
    // do something if audio is paused  
}
```