1. Yes

2. infection

3.

4. A wave

5. We are no longer conscious

6. yes

7. because of tectonic plates

8. It moves because it is pumped by the heart

9. No

10. Because of the big bang

//BACKGROUND:

/\*

const container = document.getElementById('dottedBackground');

  for (let i = 0; i < 6; i++) {

    const item = document.createElement('svg');

    item.outerHTML = `<svg xmlns="http://www.w3.org/2000/svg"> <path d="M14.554,0C6.561,0,0,6.562,0,14.552c0,7.996,6.561,14.555,14.554,14.555c7.996,0,14.553-6.559,14.553-14.555     C29.106,6.562,22.55,0,14.554,0z"/> </svg>`;

    container.appendChild(item);

  }

\*/

//TIMER:

let [milliseconds, seconds, minutes, hours] = [0, 0, 0, 0];

let timerReference = document.querySelector('.timer');

let timerID = null;

let playing = false;

let playButtonSVG = document.getElementById("playButtonSVG");

function resetTimer () {

  clearInterval(timerID);

  timerID = null;

  playButtonSVG.outerHTML = `<svg id="playButtonSVG" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"><path d="m16.53 11.152-8-5A1 1 0 0 0 7 7v10a1 1 0 0 0 1.53.848l8-5a1 1 0 0 0 0-1.7zM9 15.2V8.805L14.113 12z" style="fill:#1c1b1e"/>

        </svg> `

  playButtonSVG = document.getElementById("playButtonSVG");

  playing = false;

  [milliseconds, seconds, minutes, hours] = [0, 0, 0, 0];

  timerReference.setAttribute("placeholder", "00:00:00:00") = '00:00:00:00';

}

function playTimer() {

  timerID = setInterval(incrementTimer, 10);

  //Setting the correct height for the pause button

  playButtonSVG.outerHTML = `<svg id="playButtonSVG" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 32 32"><path d="M11.12 2H7.88A2.88 2.88 0 0 0 5 4.88v22.24A2.88 2.88 0 0 0 7.88 30h3.24A2.88 2.88 0 0 0 14 27.12V4.88A2.88 2.88 0 0 0 11.12 2zM12 27.12a.89.89 0 0 1-.88.88H7.88a.89.89 0 0 1-.88-.88V4.88A.89.89 0 0 1 7.88 4h3.24a.89.89 0 0 1 .88.88zM24.12 2h-3.24A2.88 2.88 0 0 0 18 4.88v22.24A2.88 2.88 0 0 0 20.88 30h3.24A2.88 2.88 0 0 0 27 27.12V4.88A2.88 2.88 0 0 0 24.12 2zM25 27.12a.89.89 0 0 1-.88.88h-3.24a.89.89 0 0 1-.88-.88V4.88a.89.89 0 0 1 .88-.88h3.24a.89.89 0 0 1 .88.88z"/></svg>`

  playButtonSVG = document.getElementById("playButtonSVG");

  playButtonSVG.style.height = "200px";

}

function pauseTimer() {

  clearInterval(timerID);

  playButtonSVG.outerHTML = `<svg id="playButtonSVG" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"><path d="m16.53 11.152-8-5A1 1 0 0 0 7 7v10a1 1 0 0 0 1.53.848l8-5a1 1 0 0 0 0-1.7zM9 15.2V8.805L14.113 12z" style="fill:#1c1b1e"/>

        </svg> `

  playButtonSVG = document.getElementById("playButtonSVG");

}

function onMainButtonClick() {

  console.log("onMainButtonClick() -- Check");

  if (playing == false) {

    playTimer();

    playing = true;

  } else {

    pauseTimer();

    playing = false;

  }

}

document.getElementById('startButton').addEventListener('click', (event) => onMainButtonClick(event));

document.getElementById('resetButton').addEventListener('click', resetTimer)

function changeValues\_arr\_miltohrs() {

  //milliseconds, seconds, minutes, hours = milliseconds, seconds, minutes, hours;

  milliseconds = milliseconds + 1;

  if (milliseconds == 100) {

    milliseconds = 0;

    seconds++;

    if (seconds == 60) {

      seconds = 0;

      minutes++;

      if (minutes == 60) {

        minutes = 0;

        hours++;

      }

    }

  }

}

function decreaseTimer() {

  milliseconds = milliseconds - 1;

  if (milliseconds == 0) {

    milliseconds = 100;

    seconds--;

    if (seconds == 0) {

      seconds = 60;

      minutes--;

      if (minutes == 0) {

        minutes = 60;

        hours--;

        if (hours == 0) {

          console.log('Finished!')

          window.alert("finished");

        }

      }

    }

  }

}

}

function incrementTimer() {

  changeValues\_arr\_miltohrs();

  let msLength = milliseconds.toString().length;

  let h = hours < 10 ? "0" + hours : hours;

  let m = minutes < 10 ? "0" + minutes : minutes;

  let s = seconds < 10 ? "0" + seconds : seconds;

  //let twoLongMS = Number(String(milliseconds).substring(0, (msLength - 1)));

  //let ms = twoLongMS < 10 ? "0" + twoLongMS : twoLongMS;

  let ms = milliseconds < 10 ? "0" + milliseconds : milliseconds;

  //let ms = milliseconds;

  timerReference.setAttribute("placeholder", ` ${h}:${m}:${s}:${ms}`)

}

//  <svg version="1.1" id="Layer\_1" xmlns="http://www.w3.org/2000/svg" x="0" y="0" viewBox="0 0 32 32" style="enable-background:new 0 0 32 32" xml:space="preserve"><style></style><path d="M13 28H7a1 1 0 0 1-1-1V5a1 1 0 0 1 1-1h6a1 1 0 0 1 1 1v22a1 1 0 0 1-1 1zm-5-2h4V6H8v20zM25 28h-6a1 1 0 0 1-1-1V5a1 1 0 0 1 1-1h6a1 1 0 0 1 1 1v22a1 1 0 0 1-1 1zm-5-2h4V6h-4v20z"/></svg>

//PLACEHOLDER KEYDOWN:

console.log("done")

timerReference = document.getElementById('timer');

let timerStringLong = String(timerReference.getAttribute("placeholder"));

let timerStringShort = longToShort(timerStringLong);

function redefine() {

  timerReference = document.getElementById('timer');

  timerStringLong = String(timerReference.getAttribute("placeholder"));

  timerStringShort = longToShort(timerStringLong);

  return timerReference, timerStringLong, timerStringShort;

}

//Function for making a number without a bunch of zeros before it

function longToShort(str) {

  //

  let length = str.length;

  let i = 0;

  while (i < 11) {

    firstLetter = str.substring(0, 1);

    if (firstLetter == ":") {

      str = str.substring(1, length);

    }

    else if (firstLetter == "0") {

      str = str.substring(1, length);

    }

    else if (firstLetter !== "0") {

      break

    }

    i++

  }

  if (str == "") {

    str = "0";

  }

  return str;

}

function shortToLong (str) {

  str = str;

  let pureStr = str.replaceAll(":", "");

  let length = pureStr.length;

  let missingZeros = 8 - length;

  let i = 0;

  while (i < missingZeros) {

    pureStr = "0" + pureStr;

    i++

  }

  let listValues = [];

  i = 0;

  let newStr = ""

  while (i < 4) {

    newVal = pureStr.substring(0, 2);

    newStr = newVal + ":" + newStr

    newVal = null;

    i++

  }

  str = newStr;

  return str

}

//[hours, minutes, seconds, milliseconds] = timerStringLong.split(':');

//Checking if they've entered a value and changing the placeholder to that

timerReference.addEventListener('keydown', keydownFunc);

function keydownFunc(event) {

  // Check if the Enter key is pressed

  event.preventDefault();

  if (event.key === 'Enter') {

      // Get the value of the input field

      let inputValue = this.placeholder;

      // Do something with the input value (e.g., display an alert)

      console.log("you entered", inputValue);

      timerReference.setAttribute("placeholder", inputValue);

      //Make it so that the text is align is center again

      timerReference.style.textAlign = "center";

      timerReference.style.cursor = "auto";

      // Clear the input field

      //this.value = '';

  }

};

//Changing the value to the placeholder if they've clicked on it

let timerClicked = false;

timerReference.addEventListener("click", onTimerClick);

function onTimerClick(event) {

  //console.log("worked");

  timerClicked = true;

  //console.log("Clicked the timer");

  placeholder = timerReference.getAttribute("placeholder");

  //Setting the text align right for aesthetics purposes.

  timerReference.style.textAlign = "right";

  this.placeholder = timerReference.getAttribute("placeholder");

  console.log("CLICKED: the current placeholder is ", this.placeholder);

  //timerReference.setAttribute("value", this.value);

}

document.addEventListener('click', onDocumentClick);

function onDocumentClick(event) {

  if (timerClicked) {

    timerClicked = false;

    return;

  }

  if (!timerReference.contains(event.target)) {

    console.log("clicked outside of the box");

    document.getElementById('timer').style.textAlign = 'center';

  }

}

timerReference.addEventListener("keydown", backspace);

function backspace(event) {

  let newStr = "";

  if (event.key === 'Backspace') {

    console.log("Key: Backspace");

    redefine();

    let TSLPure = timerStringLong.replaceAll(":", "");

    newInputValue = TSLPure.substring(0, (TSLPure.length - 1));

    newInputValue = "0" + newInputValue;

    console.log("newVal: ", newInputValue);

    i = 0;

    pureStr = newInputValue;

    newStr = "";

    while (i < 4) {

      //Make new values and new strings

      newVal = pureStr.substring(0, 2);

      newStr = newStr + ":" + newVal;

      //Delete from pureStr to make it work

      pureStr = pureStr.substring(2, pureStr.length)

      console.log("backspace step ", i, newVal, newStr);

      newVal = null;

      i++

    }

    //This is to remove the innevitable extra colon at the beginning

    newStr = newStr.substring(1, (newStr.length));

    console.log("newStr: ", newStr);

    redefine()

    //timerReference.setAttribute("value", newStr);

    timerReference.setAttribute("placeholder", newStr);

    redefine()

    //console.log("BOOOOM:", timerReference.getAttribute("value"));

  //if (event.key === 'Enter') {

  //  console.log("Key: Enter");

  //}

  // Set the new value to the input field

  //timerReference.value = newStr;

  timerReference.setAttribute("placeholder", newStr);

  // Prevent the default backspace action

  event.preventDefault();

  }

}

//redefine()

//timerReference.addEventListener("keydown", enteredInput);

milliseconds, seconds, minutes, hours = strToVars(String(document.getElementById('timer').placeholder));

timeValues = [milliseconds, seconds, minutes, hours];

document.getElementById('timer').addEventListener("keydown", (event) => enteredInput(event));

let timeValsPerm = {

  milliseconds: 0,

  seconds: 0,

  minutes: 0,

  hours: 0

}

function enteredInput(event) {

  event.preventDefault();

  let key = event.key;

  let nums = ["0", "1", "2", "3", "4", "5", "6", "7", "8", "9"]

  console.log("-----------------------------------------");

  console.log("baa: ", document.getElementById('timer').placeholder,   typeof("baa: ", document.getElementById('timer').placeholder));

  let timeValues = strToVars(document.getElementById('timer').placeholder);

  let [milliseconds, seconds, minutes, hours] = timeValues;

  console.log(milliseconds, seconds, minutes, hours);

  //timeValues = [milliseconds, seconds, minutes, hours];

  let timerReference, TSL, TSS = redefine();

  //TSL = varsToStr([milliseconds, seconds, minutes, hours]);

  TSL = varsToStr([hours, minutes, seconds, milliseconds]);

  console.log("this is timeValues: ", timeValues);

  TSL = expandSections\_str(TSL);

  console.log("TSL      :", TSL);

  console.log("pull up");

  let newInputPlaceholder;

  if (nums.includes(key)) {

    if (TSL == undefined) TSL = "00:00:00:00";

    let TSLPure = TSL.replaceAll(":", "");

    let TSLPureLength = TSLPure.length;

    //milliseconds, seconds, minutes, hours = strToVars(TSL);

    console.log("num: ", key);

    console.log();

    console.log("Before   ", hours, minutes, seconds, milliseconds);

    console.log();

    TSLPure = TSLPure.substring(1, TSLPureLength);

    TSLPure = TSLPure + String(event.key);

    TSL = String(String(TSLPure.slice(0, 2)) + ":" +

     String(TSLPure.slice(2, 4)) + ":" +

     String(TSLPure.slice(4, 6)) + ":" +

     String(TSLPure.slice(6, 8)));

/\* String(TSLPure.slice(6, 8)) + ":" +

     String(TSLPure.slice(4, 6)) + ":"

     String(TSLPure.slice(2, 4)) + ":"

     String(TSLPure.slice(0, 2));

     \*/

    console.log("TSLPure: ", TSLPure);

    console.log("TSL: ", TSL);

    //[milliseconds, seconds, minutes, hours] = [TSLPure.slice(6, 8), TSLPure.slice(4, 6), TSLPure.slice(2, 4), TSLPure.slice(0, 2)];

    console.log("Begin");

    console.log("After ", hours, minutes, seconds, milliseconds);

    [hours, minutes, seconds, milliseconds] = strToVars(TSL);

    console.log("After ", hours, minutes, seconds, milliseconds);

    console.log("End");

    console.log();

    console.log("After ", hours, minutes, seconds, milliseconds);

    console.log("AFter :", hours, milliseconds);

    console.log();

    newInputPlaceholder = varsToStr([milliseconds, seconds, minutes, hours]);

    //document.getElementById('timer').value = newInputValue;

    //updateInputsValuePlaceholder\_2FalseOrStr(newInputValue, false)

  }

  newInputPlaceholder = expandSections\_str(newInputPlaceholder);

  console.log("new input placeholder: ", newInputPlaceholder);

  console.log(" (Mil -> Hour) Time values:", milliseconds, seconds, minutes, hours);

  timeValsPerm.milliseconds = milliseconds;

  timeValsPerm.seconds = seconds;

  timeValsPerm.minutes = minutes;

  timeValsPerm.hours = hours;

  document.getElementById('timer').placeholder = newInputPlaceholder;

}

function strToVars(str) {

  str = str;

  let pureStr = str.replaceAll(":", "");

  let newStr = "";

  let i = 0;

  console.log("STR TO VARS");

  console.log("passed str: ", str)

  let values = [];

    while (i < 4) {

      //Make new values and new strings

      newVal = pureStr.substring(0, 2);

      values.unshift(Number(newVal));

      //Delete from pureStr to make it work

      pureStr = pureStr.substring(2, pureStr.length)

      console.log("strToVars step ", i, newVal, values);

      newVal = null;

      i++

    }

    console.log(values);

    hours = Number(values[3]);

    //console.log(values[3]);

    minutes = Number(values[2]);

    seconds = Number(values[1]);

    milliseconds = Number(values[0]);

    return [hours, minutes, seconds, milliseconds];

}

function varsToStr(arrayWithVariables) {

  let vars = arrayWithVariables;

  let i = 0;

  let varsLength = vars.length;

  let newStr = "";

  while (i < varsLength) {

    newStr = vars[i] + ":" + newStr;

    //console.log("varsToStr step ", i, vars[i], "  ", newStr);

    i++

  }

  newStr = newStr.substring(0, (newStr.length - 1));

  return newStr;

}

function expandSections\_str(str) {

  str = str;

  let i = 0;

  sectionsList = str.split(":");

  while (i < 4) {

    if (sectionsList[i].length < 2) {

      sectionsList[i] = "0" + sectionsList[i];

    }

    i++

  }

  newStr = sectionsList.join(":");

  return newStr;

}

function expandSections\_arr(arr) {

  let i = 0;

  arr = arr;

  let lengthArr = arr.length;

  while (i < lengthArr) {

    arr[i] = String(arr[i]);

    if (arr[i].length < 2) {

      arr[i] = "0" + arr[i];

      arr[i] = Number(arr[i]);

    }

    i++

  }

  return arr;

}

function updateInputsValuePlaceholder\_2FalseOrStr(updateValue, updatePlaceholder) {

  let timerReference, TSL, TSS = redefine()

  timerReference = document.getElementById('timer');

  if (updateValue !== false) {

    timerReference.value = updateValue;

  }

  if (updatePlaceholder !== false) {

    timerReference.placeholder = updatePlaceholder

  }

}

/\*

timerReference.addEventListener("keydown", backspace);

function backspace(event) {

    if (event.key === 'Backspace') {

        console.log("Key: Backspace");

        // Get the current value and remove colons

        let TSLPure = timerReference.value.replaceAll(":", "");

        // Remove the last digit, prepend a '0'

        let newInputValue = "0" + TSLPure.slice(0, -1);

        // Reinsert colons every 2 digits

        let newStr = `${newInputValue.slice(0, 2)}:${newInputValue.slice(2, 4)}:${newInputValue.slice(4)}`;

        console.log("newStr: ", newStr);

        // Set the new value to the input field

        timerReference.value = newStr;

        timerReference.setAttribute("placeholder", newStr);

        // Prevent the default backspace action

        event.preventDefault();

    }

}

\*/

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

//BACKGROUND:

/\*

const container = document.getElementById('dottedBackground');

  for (let i = 0; i < 6; i++) {

    const item = document.createElement('svg');

    item.outerHTML = `<svg xmlns="http://www.w3.org/2000/svg"> <path d="M14.554,0C6.561,0,0,6.562,0,14.552c0,7.996,6.561,14.555,14.554,14.555c7.996,0,14.553-6.559,14.553-14.555     C29.106,6.562,22.55,0,14.554,0z"/> </svg>`;

    container.appendChild(item);

  }

\*/

//TIMER:

let [milliseconds, seconds, minutes, hours] = [0, 0, 0, 0];

let timerReference = document.querySelector('.timer');

let timerID = null;

let playing = false;

let playButtonSVG = document.getElementById("playButtonSVG");

function resetTimer () {

  clearInterval(timerID);

  timerID = null;

  playButtonSVG.outerHTML = `<svg id="playButtonSVG" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"><path d="m16.53 11.152-8-5A1 1 0 0 0 7 7v10a1 1 0 0 0 1.53.848l8-5a1 1 0 0 0 0-1.7zM9 15.2V8.805L14.113 12z" style="fill:#1c1b1e"/>

        </svg> `

  playButtonSVG = document.getElementById("playButtonSVG");

  playing = false;

  [milliseconds, seconds, minutes, hours] = [0, 0, 0, 0];

  timerReference.setAttribute("placeholder", "00:00:00:00") = '00:00:00:00';

}

function playTimer () {

  if (playing == false) {

    timerID = setInterval(incrementTimer, 10);

  //Setting the correct height for the pause button

    playButtonSVG.outerHTML = `<svg id="playButtonSVG" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 32 32"><path d="M11.12 2H7.88A2.88 2.88 0 0 0 5 4.88v22.24A2.88 2.88 0 0 0 7.88 30h3.24A2.88 2.88 0 0 0 14 27.12V4.88A2.88 2.88 0 0 0 11.12 2zM12 27.12a.89.89 0 0 1-.88.88H7.88a.89.89 0 0 1-.88-.88V4.88A.89.89 0 0 1 7.88 4h3.24a.89.89 0 0 1 .88.88zM24.12 2h-3.24A2.88 2.88 0 0 0 18 4.88v22.24A2.88 2.88 0 0 0 20.88 30h3.24A2.88 2.88 0 0 0 27 27.12V4.88A2.88 2.88 0 0 0 24.12 2zM25 27.12a.89.89 0 0 1-.88.88h-3.24a.89.89 0 0 1-.88-.88V4.88a.89.89 0 0 1 .88-.88h3.24a.89.89 0 0 1 .88.88z"/></svg>`

    playButtonSVG = document.getElementById("playButtonSVG");

    playButtonSVG.style.height = "200px";

    playing = true;

  } else {

    clearInterval(timerID);

    playButtonSVG.outerHTML = `<svg id="playButtonSVG" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"><path d="m16.53 11.152-8-5A1 1 0 0 0 7 7v10a1 1 0 0 0 1.53.848l8-5a1 1 0 0 0 0-1.7zM9 15.2V8.805L14.113 12z" style="fill:#1c1b1e"/>

          </svg> `

    playButtonSVG = document.getElementById("playButtonSVG");

    playing = false;

  }

}

document.getElementById('startButton').addEventListener('click', (event) => playTimer(event));

document.getElementById('resetButton').addEventListener('click', resetTimer)

function changeValues\_arr\_miltohrs() {

  //milliseconds, seconds, minutes, hours = milliseconds, seconds, minutes, hours;

  milliseconds = milliseconds + 1;

  if (milliseconds == 100) {

    milliseconds = 0;

    seconds++;

    if (seconds == 60) {

      seconds = 0;

      minutes++;

      if (minutes == 60) {

        minutes = 0;

        hours++;

      }

    }

  }

}

function decreaseTimer() {

  milliseconds = milliseconds - 1;

  if (milliseconds == 0) {

    milliseconds = 100;

    seconds--;

    if (seconds == 0) {

      seconds = 60;

      minutes--;

      if (minutes == 0) {

        minutes = 60;

        hours--;

        if (hours == 0) {

          console.log('Finished!')

          window.alert("finished")

        }

      }

    }

  }

}

}

function incrementTimer() {

  changeValues\_arr\_miltohrs();

  let msLength = milliseconds.toString().length;

  let h = hours < 10 ? "0" + hours : hours;

  let m = minutes < 10 ? "0" + minutes : minutes;

  let s = seconds < 10 ? "0" + seconds : seconds;

  //let twoLongMS = Number(String(milliseconds).substring(0, (msLength - 1)));

  //let ms = twoLongMS < 10 ? "0" + twoLongMS : twoLongMS;

  let ms = milliseconds < 10 ? "0" + milliseconds : milliseconds;

  //let ms = milliseconds;

  timerReference.setAttribute("placeholder", ` ${h}:${m}:${s}:${ms}`)

}

//  <svg version="1.1" id="Layer\_1" xmlns="http://www.w3.org/2000/svg" x="0" y="0" viewBox="0 0 32 32" style="enable-background:new 0 0 32 32" xml:space="preserve"><style></style><path d="M13 28H7a1 1 0 0 1-1-1V5a1 1 0 0 1 1-1h6a1 1 0 0 1 1 1v22a1 1 0 0 1-1 1zm-5-2h4V6H8v20zM25 28h-6a1 1 0 0 1-1-1V5a1 1 0 0 1 1-1h6a1 1 0 0 1 1 1v22a1 1 0 0 1-1 1zm-5-2h4V6h-4v20z"/></svg>

//PLACEHOLDER KEYDOWN:

console.log("done")

timerReference = document.getElementById('timer');

let timerStringLong = String(timerReference.getAttribute("placeholder"));

let timerStringShort = longToShort(timerStringLong);

function redefine() {

  timerReference = document.getElementById('timer');

  timerStringLong = String(timerReference.getAttribute("placeholder"));

  timerStringShort = longToShort(timerStringLong);

  return timerReference, timerStringLong, timerStringShort;

}

//Function for making a number without a bunch of zeros before it

function longToShort(str) {

  //

  let length = str.length;

  let i = 0;

  while (i < 11) {

    firstLetter = str.substring(0, 1);

    if (firstLetter == ":") {

      str = str.substring(1, length);

    }

    else if (firstLetter == "0") {

      str = str.substring(1, length);

    }

    else if (firstLetter !== "0") {

      break

    }

    i++

  }

  if (str == "") {

    str = "0";

  }

  return str;

}

function shortToLong (str) {

  str = str;

  let pureStr = str.replaceAll(":", "");

  let length = pureStr.length;

  let missingZeros = 8 - length;

  let i = 0;

  while (i < missingZeros) {

    pureStr = "0" + pureStr;

    i++

  }

  let listValues = [];

  i = 0;

  let newStr = ""

  while (i < 4) {

    newVal = pureStr.substring(0, 2);

    newStr = newVal + ":" + newStr

    newVal = null;

    i++

  }

  str = newStr;

  return str

}

//[hours, minutes, seconds, milliseconds] = timerStringLong.split(':');

//Checking if they've entered a value and changing the placeholder to that

timerReference.addEventListener('keydown', keydownFunc);

function keydownFunc(event) {

  // Check if the Enter key is pressed

  event.preventDefault();

  if (event.key === 'Enter') {

      // Get the value of the input field

      let inputValue = this.placeholder;

      // Do something with the input value (e.g., display an alert)

      console.log("you entered", inputValue);

      timerReference.setAttribute("placeholder", inputValue);

      //Make it so that the text is align is center again

      timerReference.style.textAlign = "center";

      timerReference.style.cursor = "auto";

      // Clear the input field

      //this.value = '';

  }

};

//Changing the value to the placeholder if they've clicked on it

let timerClicked = false;

timerReference.addEventListener("click", onTimerClick);

function onTimerClick(event) {

  //console.log("worked");

  timerClicked = true;

  //console.log("Clicked the timer");

  placeholder = timerReference.getAttribute("placeholder");

  //Setting the text align right for aesthetics purposes.

  timerReference.style.textAlign = "right";

  this.placeholder = timerReference.getAttribute("placeholder");

  console.log("CLICKED: the current placeholder is ", this.placeholder);

  //timerReference.setAttribute("value", this.value);

}

document.addEventListener('click', onDocumentClick);

function onDocumentClick(event) {

  if (timerClicked) {

    timerClicked = false;

    return;

  }

  if (!timerReference.contains(event.target)) {

    console.log("clicked outside of the box");

    document.getElementById('timer').style.textAlign = 'center';

  }

}

timerReference.addEventListener("keydown", backspace);

function backspace(event) {

  let newStr = "";

  if (event.key === 'Backspace') {

    console.log("Key: Backspace");

    redefine();

    let TSLPure = timerStringLong.replaceAll(":", "");

    newInputValue = TSLPure.substring(0, (TSLPure.length - 1));

    newInputValue = "0" + newInputValue;

    console.log("newVal: ", newInputValue);

    i = 0;

    pureStr = newInputValue;

    newStr = "";

    while (i < 4) {

      //Make new values and new strings

      newVal = pureStr.substring(0, 2);

      newStr = newStr + ":" + newVal;

      //Delete from pureStr to make it work

      pureStr = pureStr.substring(2, pureStr.length)

      console.log("backspace step ", i, newVal, newStr);

      newVal = null;

      i++

    }

    //This is to remove the innevitable extra colon at the beginning

    newStr = newStr.substring(1, (newStr.length));

    console.log("newStr: ", newStr);

    redefine()

    //timerReference.setAttribute("value", newStr);

    timerReference.setAttribute("placeholder", newStr);

    redefine()

    //console.log("BOOOOM:", timerReference.getAttribute("value"));

  //if (event.key === 'Enter') {

  //  console.log("Key: Enter");

  //}

  // Set the new value to the input field

  //timerReference.value = newStr;

  timerReference.setAttribute("placeholder", newStr);

  // Prevent the default backspace action

  event.preventDefault();

  }

}

//redefine()

//timerReference.addEventListener("keydown", enteredInput);

milliseconds, seconds, minutes, hours = strToVars(String(document.getElementById('timer').placeholder));

timeValues = [milliseconds, seconds, minutes, hours];

document.getElementById('timer').addEventListener("keydown", (event) => enteredInput(event));

let timeValsPerm = {

  milliseconds: 0,

  seconds: 0,

  minutes: 0,

  hours: 0

}

function enteredInput(event) {

  event.preventDefault();

  let key = event.key;

  let nums = ["0", "1", "2", "3", "4", "5", "6", "7", "8", "9"]

  console.log("-----------------------------------------");

  console.log("baa: ", document.getElementById('timer').placeholder,   typeof("baa: ", document.getElementById('timer').placeholder));

  let timeValues = strToVars(document.getElementById('timer').placeholder);

  let [milliseconds, seconds, minutes, hours] = timeValues;

  console.log(milliseconds, seconds, minutes, hours);

  //timeValues = [milliseconds, seconds, minutes, hours];

  let timerReference, TSL, TSS = redefine();

  //TSL = varsToStr([milliseconds, seconds, minutes, hours]);

  TSL = varsToStr([hours, minutes, seconds, milliseconds]);

  console.log("this is timeValues: ", timeValues);

  TSL = expandSections\_str(TSL);

  console.log("TSL      :", TSL);

  console.log("pull up");

  let newInputPlaceholder;

  if (nums.includes(key)) {

    if (TSL == undefined) TSL = "00:00:00:00";

    let TSLPure = TSL.replaceAll(":", "");

    let TSLPureLength = TSLPure.length;

    //milliseconds, seconds, minutes, hours = strToVars(TSL);

    console.log("num: ", key);

    console.log();

    console.log("Before   ", hours, minutes, seconds, milliseconds);

    console.log();

    TSLPure = TSLPure.substring(1, TSLPureLength);

    TSLPure = TSLPure + String(event.key);

    TSL = String(String(TSLPure.slice(0, 2)) + ":" +

     String(TSLPure.slice(2, 4)) + ":" +

     String(TSLPure.slice(4, 6)) + ":" +

     String(TSLPure.slice(6, 8)));

/\* String(TSLPure.slice(6, 8)) + ":" +

     String(TSLPure.slice(4, 6)) + ":"

     String(TSLPure.slice(2, 4)) + ":"

     String(TSLPure.slice(0, 2));

     \*/

    console.log("TSLPure: ", TSLPure);

    console.log("TSL: ", TSL);

    //[milliseconds, seconds, minutes, hours] = [TSLPure.slice(6, 8), TSLPure.slice(4, 6), TSLPure.slice(2, 4), TSLPure.slice(0, 2)];

    console.log("Begin");

    console.log("After ", hours, minutes, seconds, milliseconds);

    [hours, minutes, seconds, milliseconds] = strToVars(TSL);

    console.log("After ", hours, minutes, seconds, milliseconds);

    console.log("End");

    console.log();

    console.log("After ", hours, minutes, seconds, milliseconds);

    console.log("AFter :", hours, milliseconds);

    console.log();

    newInputPlaceholder = varsToStr([milliseconds, seconds, minutes, hours]);

    //document.getElementById('timer').value = newInputValue;

    //updateInputsValuePlaceholder\_2FalseOrStr(newInputValue, false)

  }

  newInputPlaceholder = expandSections\_str(newInputPlaceholder);

  console.log("new input placeholder: ", newInputPlaceholder);

  console.log(" (Mil -> Hour) Time values:", milliseconds, seconds, minutes, hours);

  timeValsPerm.milliseconds = milliseconds;

  timeValsPerm.seconds = seconds;

  timeValsPerm.minutes = minutes;

  timeValsPerm.hours = hours;

  document.getElementById('timer').placeholder = newInputPlaceholder;

}

function strToVars(str) {

  str = str;

  let pureStr = str.replaceAll(":", "");

  let newStr = "";

  let i = 0;

  console.log("STR TO VARS");

  console.log("passed str: ", str)

  let values = [];

    while (i < 4) {

      //Make new values and new strings

      newVal = pureStr.substring(0, 2);

      values.unshift(Number(newVal));

      //Delete from pureStr to make it work

      pureStr = pureStr.substring(2, pureStr.length)

      console.log("strToVars step ", i, newVal, values);

      newVal = null;

      i++

    }

    console.log(values);

    hours = Number(values[3]);

    //console.log(values[3]);

    minutes = Number(values[2]);

    seconds = Number(values[1]);

    milliseconds = Number(values[0]);

    return [hours, minutes, seconds, milliseconds];

}

function varsToStr(arrayWithVariables) {

  let vars = arrayWithVariables;

  let i = 0;

  let varsLength = vars.length;

  let newStr = "";

  while (i < varsLength) {

    newStr = vars[i] + ":" + newStr;

    //console.log("varsToStr step ", i, vars[i], "  ", newStr);

    i++

  }

  newStr = newStr.substring(0, (newStr.length - 1));

  return newStr;

}

function expandSections\_str(str) {

  str = str;

  let i = 0;

  sectionsList = str.split(":");

  while (i < 4) {

    if (sectionsList[i].length < 2) {

      sectionsList[i] = "0" + sectionsList[i];

    }

    i++

  }

  newStr = sectionsList.join(":");

  return newStr;

}

function expandSections\_arr(arr) {

  let i = 0;

  arr = arr;

  let lengthArr = arr.length;

  while (i < lengthArr) {

    arr[i] = String(arr[i]);

    if (arr[i].length < 2) {

      arr[i] = "0" + arr[i];

      arr[i] = Number(arr[i]);

    }

    i++

  }

  return arr;

}

function updateInputsValuePlaceholder\_2FalseOrStr(updateValue, updatePlaceholder) {

  let timerReference, TSL, TSS = redefine()

  timerReference = document.getElementById('timer');

  if (updateValue !== false) {

    timerReference.value = updateValue;

  }

  if (updatePlaceholder !== false) {

    timerReference.placeholder = updatePlaceholder

  }

}

/\*

timerReference.addEventListener("keydown", backspace);

function backspace(event) {

    if (event.key === 'Backspace') {

        console.log("Key: Backspace");

        // Get the current value and remove colons

        let TSLPure = timerReference.value.replaceAll(":", "");

        // Remove the last digit, prepend a '0'

        let newInputValue = "0" + TSLPure.slice(0, -1);

        // Reinsert colons every 2 digits

        let newStr = `${newInputValue.slice(0, 2)}:${newInputValue.slice(2, 4)}:${newInputValue.slice(4)}`;

        console.log("newStr: ", newStr);

        // Set the new value to the input field

        timerReference.value = newStr;

        timerReference.setAttribute("placeholder", newStr);

        // Prevent the default backspace action

        event.preventDefault();

    }

}

\*/