Ilma selleta edasi ei lähe 👁

./ - sama kaust kus sa oled

../ - kaustast väljas

Kirj Java faili:

console.log(‘Tootab’) //JavaScripti testimiseks

Const name = ’Kevin’ //see on andmetüüp string e. tekst

Const name = 41 //num (alam. Kategooria int)

Const = muutuja tüüp

Const money = null //andmetüüp null, väärtus puudub hetkel aga seda saab hiljem panna

Const nice = false / const nice = true //boolean, andmetüüp on alati kas aniult väär või tõene

Const friends = [‘Gen’, ‘Karl’] //Array või Eesti keeles jada (pea meeles Array pikkust ja index-it.)

Kuidas töötab loop?

Loop teeb mingit funktsiooni kuni loop ära lõppeb (while Loop)

Võtab Array seest elemendi, teeb loogika sellega läbi ja võtab järgmise.

Console.log(friends)

Const favouriteCar = {

Id: 1;

Type: ‘manual’,

Mark: ‘proche’;

} //Object

Const sum = 20 + 10 //0.3

Console.log(sum) // kontrollib kas tehte sum on tõesti 0.3

Läbi console’i saab vaadata ka erinevaid evente (nt. On drag jms) selle abiga on lihtsam neid muuta ig.

Array’d saab killustada või üle loop’ida nt; While Loop’iga, For Loop’iga, forEach’iga.

Hea research link teema kohta; <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array>

Nii nagu ka Pythoni puhul, hakkab ka JavaScript’is Array loendamine 0-ist.

Callback – nimetad ühe elemendi ära.

Const.persons = [‘ ’, ‘ ’]

Persons.forEach(person => { …?})

Refer to an element: item //või component => {})

Nt siin:

const components = ['cpu', 'gpu', 'motherboard', 'ram', 'power supply']

components.forEach(component => {})

//component muutub cpu’ks).

Console.log(component) //Iga elemendi kohta console log component

//Sama loogika:

Components.map(component => console.log(component))

Component.filter (component => console.log(component))

Element, annad talle (elemendile) nime, kasutad teda siin sees muutujana