Harmadiki óra - zh

const planets =

`1,Tatooine,arid,10465

2,Alderaan,temperate,12500

3,Yavin IV,temperate,10200

4,Hoth,frozen,7200

5,Dagobah,murky,8900

6,Bespin,temperate,118000

7,Endor,temperate,4900

8,Naboo,temperate,12120

9,Coruscant,temperate,12240

10,Kamino,temperate,19720

11,Geonosis,temperate,11370

12,Utapau,windy,12900

13,Mustafar,hot,4200

14,Kashyyyk,tropical,12765

15,Polis Massa,artificial temperate ,0

16,Mygeeto,frigid,10088

17,Felucia,hot humid,9100

18,Cato Neimoidia,temperate moist,0

19,Saleucami,hot,14920

20,Stewjon,temperate,0

21,Eriadu,polluted,13490

22,Corellia,temperate,11000

23,Rodia,hot,7549

24,Nal Hutta,temperate,12150

25,Dantooine,temperate,9830

26,Bestine IV,temperate,6400

27,Ord Mantell,temperate,14050

29,Trandosha,arid,0

30,Socorro,arid,0

31,Mon Cala,temperate,11030

32,Chandrila,temperate,13500

33,Sullust,superheated,12780

34,Toydaria,temperate,7900

35,Malastare,arid temperate tropical,18880

36,Dathomir,temperate,10480

37,Ryloth,temperate arid subartic,10600

39,Vulpter,temperate artic,14900

41,Tund,unknown,12190

42,Haruun Kal,temperate,10120

44,Glee Anselm,tropical temperate,15600

49,Dorin,temperate,13400

57,Muunilinst,temperate,13800

59,Kalee,arid temperate tropical,13850`;

function getListItem(planet) {

return `

<li>

<div><h4>#${planet.id} ${planet.name}</h4>Climate: ${planet.climate}</div>

<svg height="120" width="120">

<circle cx="60" cy="60" r="${planet.diameter / 2000}" />

</svg>

</li>`

}

//A getPlanetData függvény egy objektumtömbbel tér vissza, az objektumok property-i: id, name, climate, diameter

//a tömb a diameter property szerint csökkenő sorrendben rendezett

//[

//{id:6, name:"Bespin", climate:"temperate", diameter:"118000"},

//{id:10, name:"Kamino", climate:"temperate", diameter:"19720"},

//{id:35, name:"Malastare", climate: arid temperate tropical, diameter:"18880"}

// ...

//]

function getPlanetData(planetList){

datalist=planetList.split('\n');

var dataArray=[ ];

for(var i=0; i<datalist.length; i++){

tmp=datalist[i].split(',');

let planet = {

id: tmp[0],

name: tmp[1],

climate: tmp[2],

diameter: tmp[3]

}

dataArray.push(planet);

}

return dataArray;

}

function diameterSorting(planet1, planet2){

return planet1.diamater.length - planet2.diameter.length;

}

function getPLanetDataSort(){

return dataArray.sort(diameterSorting);

}

console.log(getPlanetData(planets));

//A getTotal függvény egy stringet ad vissza:

//Total: [planet objektumok száma] (temperate climate: [azon planet objektumok száma, melyek climate propertyjében szerepel a `temperate` szó])

//pl. Total: 10 (temperate climate: 6)

//function getTotal() {}

//A legnagyobb átmérőjű bolygó adatait tartalmazó stringgel tér vissza

//pl Largest planet: #10 Kamino (diameter: 19720)

//function getLargest() {}

//getPlanetData().map(planet => getListItem(planet)).forEach(li => $('#planets').append(li));

//$('#total').text(getTotal());

//$('#min').text(getLargest());

Streams2

const fs = require('fs');

//const str = 'Lórum ipse az egyik szűző írásbeli kölcsepeték. Ilyen csóna valamennyi belényben matott és tulajdonképpen már ma is harjárják ezeket a gerektásokat, csüggőség kübli, torgátozás nélkül.Lórum ipse az egyik szűző írásbeli kölcsepeték. Ilyen csóna valamennyi belényben matott és tulajdonképpen már ma is harjárják ezeket a gerektásokat, csüggőség kübli, torgátozás nélkül.Lórum ipse az egyik szűző írásbeli kölcsepeték. Ilyen csóna valamennyi belényben matott és tulajdonképpen már ma is harjárják ezeket a gerektásokat, csüggőség kübli, torgátozás nélkül.';

//process.stdout.write('hello');

//process.stdin.pipe(process.stdout);

/\*const writeStream = fs.createWriteStream('./bigfile.txt');

process.stdin.pipe(writeStream);

for(var i=0; i<1e6; i++){

writeStream.write(str);

}

writeStream.end(); //ha pipe-t használunk, akkor nem kell az end\*/

//fs.readFile('./bigfile.txt', (err, data)=>console.log('Done', data.length));

const str = fs.createReadStream('./bigfile.txt');

str.pipe(process.stdout);

const http = require('http');

const port = '8084';

const url = require('url');

const fs = require('fs');

const osData = require('.service/os-data.service.js');

const serrver = http.createServer(processRequest);

serrver.listen(port);

function processRequest(request, response) {

const parsedUrl = url.parse(request.url, true);

console.log(parsedUrl);

switch (parsedUrl.pathname) {

case '/hello:': {

response.writeHead(200);

response.write(sendHello(parsedUrl));

response.end();s

} break;

case '/home':

getPage('home.html', response);

break;

case '/dashboard':

getPage('dashboard.html', response);

break;

case 'api/os-data':

response.writeHead(200, { 'content-type': 'application/json' });

response.write(JSON.stringify(osData.getOSData()));//jsont cisnál az objektumból

response.end();

break;

}

}

function getPage(path, response) {

fs.readFile('./public/' + path, (err, data) => {

if (err) {

response.writeHead(404);

response.end();

} else {

response.writeHead(200, { 'content-type': 'text/html' });

response.write(data);

response.end();

}

})

}

function sendHello(parsedUrl) {

return 'hello:' + parsedUrl.query.user;

}

const fs = require('fs');

const stream = require('stream');

/\*

const writeStream = fs.createWriteStream('./random.txt');

function rand(){

return `${Math.random()\*1e20}\n`;

}

for(let i = 0; i<1e5; i++){

writeStream.write(rand());

}

writeStream.end();\*/

let numStr=' '; //az éppen aktuális sor tárolására alkamas

const transformStream = new stream.Transform({

transform(chunk, encoding, callback){

let str = chunk.toString();

for(let i=0; i < str.length; i++){

if(str[i]=='\n'){

this.push(numStr + '\n');

numStr = '';

}else{

numStr=str[i]+numStr;

}

}

callback();

}

});

const writeStream = fs.createWriteStream('./random-copy.txt');

const readStream = fs.createReadStream('./random.txt');

readStream.pipe(transformStream).pipe(writeStream);

Morseservice js a házi feladat