Node.js
Simple Website development

Εισαγωγή

- Στη συνέχεια θα αναπτύξουμε ένα δυναμικό site, όπου τα δεδομένα που παρουσιάζονται λαμβάνονται από ένα json αρχείο

- Στο συγκεκριμένο παράδειγμα θα εστιάσουμε σε ένα προσωπικό portfolio website

Δυναμικό site

- Ένα site του οποίου το περιεχόμενο μεταβάλλεται
- Αυτό θα δημιουργήσουμε και εδώ
- Έστω ότι έχουμε ένα json αρχείο το οποίο περιέχει όλες τις γλώσσες προγραμματισμού που γνωρίζουμε
- Θέλουμε αυτές να φορτώνονται στη σελίδα μας, χωρίς να μας ενδιαφέρει πόσες είναι αυτές. Αρχικά μπορεί να είναι μία ενώ στο μέλλον 10. Θέλουμε τροποποιώντας απλά ένα json αρχείο, χωρίς να πειράξουμε τον κώδικά μας, να βλέπουμε πάντα όλες τις εγγραφές.

Βήματα

- Το πρώτο βήμα είναι η δημιουργία της html/css δομής των σελίδων της ιστοσελίδας μας (Templates)

- Στην συνέχεια αναπτύσσουμε το index.js αρχείο που είναι υπεύθυνο για την
  - διαχείριση των «requests» που γίνονται στο site
  - για την ανάγνωση των δεδομένων από το json
  - και για την φόρτωση των δεδομένων στις html σελίδες

ARISTEA KONTOGIANNI

Ph.D. Candidate

	>HTML5	★★★★★	MORE...
	>CSS3	★★★★★	MORE...
	>Javascript	★★★★★	MORE...
	>PHP	★★★★★	MORE...
	>Python	★★★	MORE...

1ο βήμα

- Δημιουργία των templates…

- Τα αποθηκεύουμε σε έναν folder-> templates

1ο βήμα

- Παραπάνω βλέπουμε ένα παράδειγμα ιστοσελίδας

- Η σελίδα αυτή μπορεί να χωριστεί στα παρακάτω δύο μέρη:
  - 1. Στο γενικό κομμάτι που περιλαμβάνει το όνομα του ατόμου, τα χρώματα και την εμφάνιση της σελίδας

  - 2. Τις κάρτες αφορούν τις γλώσσες προγραμματισμού οι οποίες έχουν την ίδια δομή αλλά που λαμβάνουν δυναμικά το περιεχόμενό τους.

1ο βήμα

- Ο ένα μέρος του κώδικα από το overview template, που διατηρεί τη γενική δομή της σελίδας δίνεται παρακάτω.

```
  <body>
    <div class="container">
      <div class="text-center">
        <h1>Aristea Kontogianni</h1>
        <h2>Ph.D. Candidate</h2>
      </div>
      <div class="cards-container">
        {%PROGRAM_CARDS%}
      </div>
    </div>
    <!-- Bootstrap JS -->
    <script src="https://stackpath.bootstrapcdn.com/bootstrap

  </body>
</html>
```

εδώ μορφώνονται οι κάρτες

1ο βήμα

- Το κάθε html template, περιλαμβάνει internal css κώδικα, έτσι ώστε να μην κάνουμε requests σε εξωτερικά αρχεία

- Επίσης στο παράδειγμα αυτό χρησιμοποιώ bootstrap

1ο βήμα

Για την δημιουργία των καρτών χρησιμοποιείται ένα ξεχωριστό αρχείο, το program-card.html το οποίο είναι το εξής

ates > program-card.html > ...

```
<figure class="card container">
  <div class="row">
    <div class="col-sm-3 card__emoji">
      <img src="{%IMAGE%}" style="width:50px;" class="img-thumbnail"></img>
    </div>
    <div class="col-sm-3 card__title-box" >
        <h2 class="card__title">>{%NAME%}</h2>
    </div>
    <div class="col-sm-2">
      <h6 class="card__detail" id="star"> {%STARS%} ★</h6>
    </div>
    <div class="col-sm-2">
      <a class="card__link" href="/program?id={%ID%}">
        <span>Detail <i class="emoji-right">👉</i></span>
      </a>
    </div>
    </div>
</figure>
```

1ο βήμα

Στη συνέχεια δημιουργούμε μια ακόμη ιστοσελίδα η οποία θα φορτώνεται όταν ο χρήστη κλικάρει για να δει περισσότερες πληροφορίες για την κάθε γλώσσα προγραμματισμού ξεχωριστά

2ο βήμα

1. Αναπτύσσω ένα index.js αρχείο
2. Διαβάζω και αποθηκεύω τα html αρχεία σε μεταβλητές

```
//TOP LEVEL CODE
// The first thing we need to do is read our overview template
// We can actually do this here, because we need to access it only once as it remains the same

const tempOverview=filesystem.readFileSync(`${__dirname}/templates/overview.html`, "utf8");
const tempProgram=filesystem.readFileSync(`${__dirname}/templates/program.html`, "utf8");
const tempCard=filesystem.readFileSync(`${__dirname}/templates/program-card.html`, "utf8");
```

2ο βήμα

1. Διαβάζω το json αρχείο και το αποθηκεύω σε μια μεταβλητή

```
//use sync version, because it is easier to handle-> we put data in a var and we use it right away
// we do not care if data is blocking code execution because this top level code
//is going to be executed only once, at the beggining

const data=filesystem.readFileSync(`${__dirname}/json/programmingl.json`, "utf8");
const dataObject= JSON.parse(data);
```

Να θυμηθούμε

- JavaScript Arrow Function

```
hello = function() {
  return "Hello World!";
}
```

```
hello = () => {
  return "Hello World!";
}
```

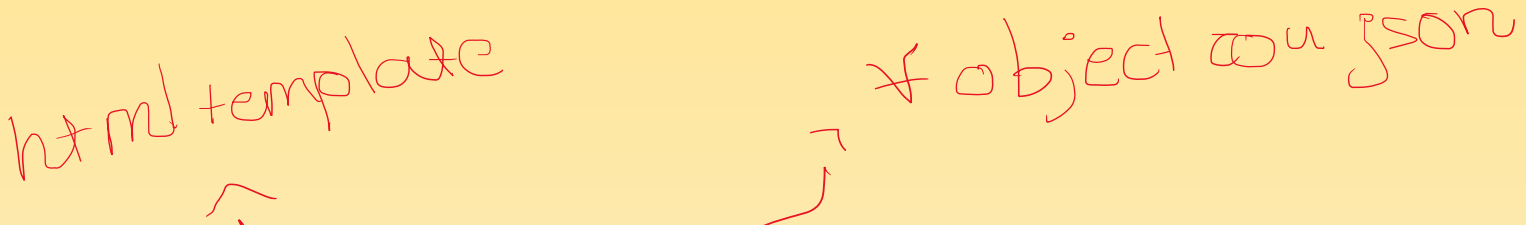
```
hello = () => "Hello World!";
```

```
hello = (val) => "Hello " + val;
```

// όλα το ίδιο!

// με παραμέτρους

//we can use the expression /g to search or extract a pattern more than once



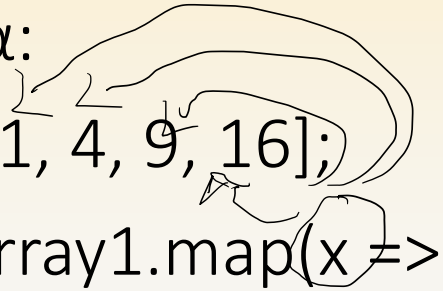

```
//lets create here the function replaceTemplate
const replaceTemplate=(temp,program) =>{
    var output = temp.replace(/{%NAME%}/g, program.name);
       output = output.replace(/{%STARS%}/g, program.stars);
       output = output.replace(/{%IMAGE%}/g, program.img);
       output = output.replace(/{%HISTORY%}/g, program.history);
       output = output.replace(/{%INFO%}/g, program.info);
       output = output.replace(/{%ID%}/g, program.id);

     if(!program.stars==5) {
         output = output.replace(/{%SENIOR%}/g, 'senior');
     }else{
        output = output.replace(/{%SENIOR%}/g, 'not-senior');

     }
    return output;
```

Να θυμηθούμε

1. Array.prototype.map()
2. Η μέθοδος map () δημιουργεί έναν νέο πίνακα(array) που συμπληρώνεται με τα αποτελέσματα της κλήσης μιας συνάρτησης σε κάθε στοιχείο του πίνακα
3. Παραδειγμα:



```
const array1 = [1, 4, 9, 16];

const map1 = array1.map(x => x * 2);

console.log(map1);// expected output: Array [2, 8, 18, 32]
```

Αυτό που θέλουμε τώρα είναι να αντικαταστήσουμε όλες τις «μεταβλητές» που έχουμε στα templates με τα δεδομένα από το json αρχείο.

Άρα θα πρέπει να έχουμε μια μέθοδο η οποία θα λαμβάνει 2 εισόδους:

Template & κάθε **element** του json αρχείου, κάθε γλώσσα προγραμματισμού δηλαδή

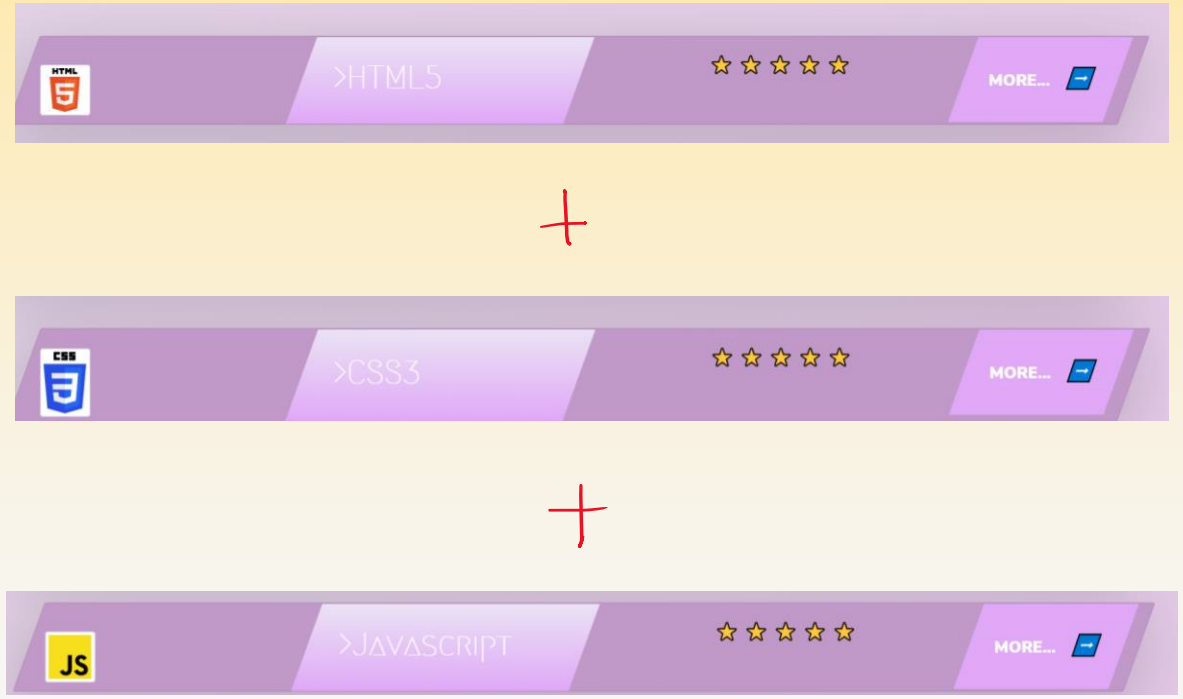
- Θα επιστρέφει μια μεταβλητή που περιλαμβάνει τον κώδικα από το template με τα δεδομένα του του json αρχείου

- Πρέπει όμως να πάρουμε ένα προς ένα τα elements του json array

- Για να το κάνουμε αυτό θα χρησιμοποιήσουμε τη map method, η οποία για κάθε element του αρχείου, κάθε γλώσσα προγραμματισμού δηλαδή, θα καλεί τη παραπάνω function η οποία παίρνει την κάθε τιμή που θέλουμε από το name/value pair στο json, και θα την βάζει στο αντίστοιχο placeholder του json αρχείου

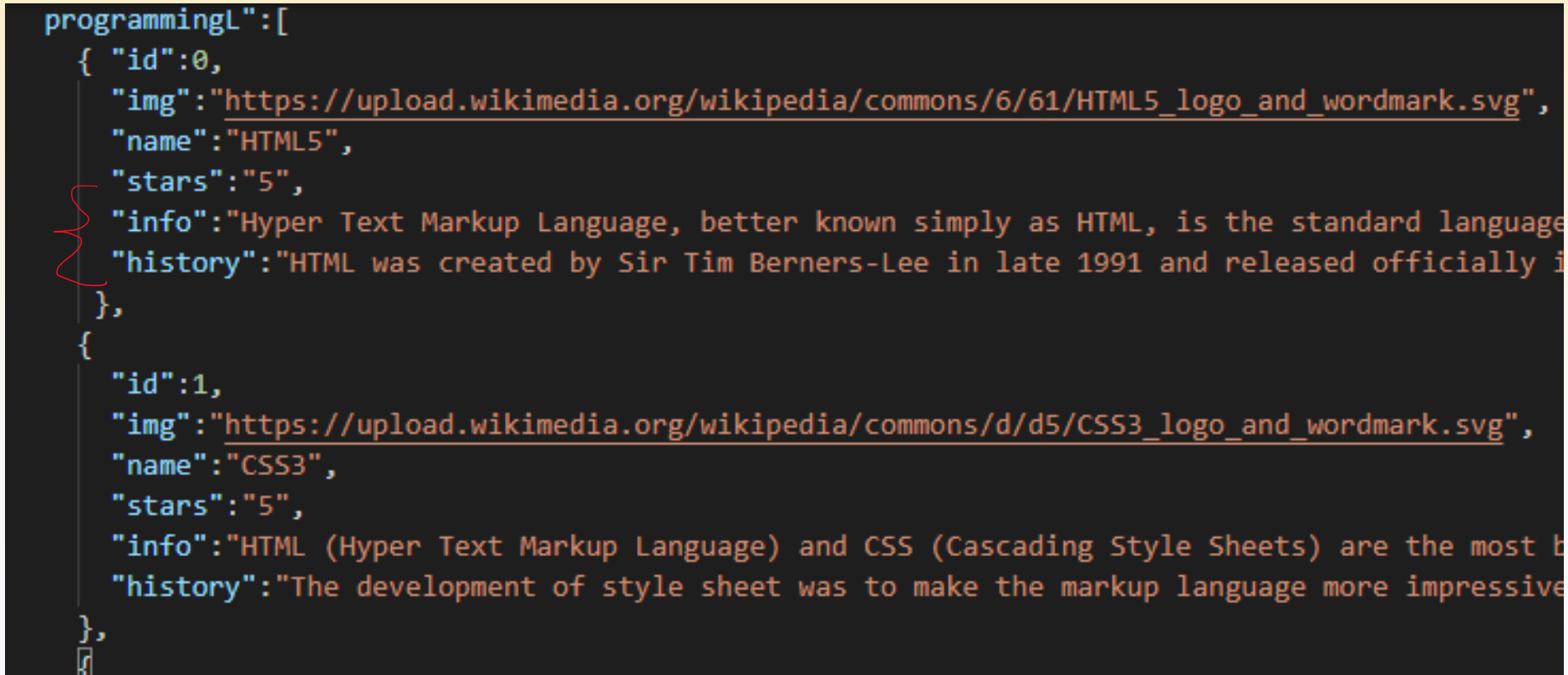
```
const cardsHtml = dataObject.programmingL.map(el => replaceTemplate(tempCard,el));
console.log(typeof(cardsHtml));
var program_cards= tempOverview.replace('{%PROGRAM_CARDS%}',cardsHtml);
```

➢Έτσι συμπληρώνονται και γίνονται join μια - μια οι κάρτες



3ο βήμα

- Αυτό που θέλουμε τώρα είναι όταν ο χρήστης πατάει το link “more” να φορτώνονται ορισμένες πληροφορίες για την κάθε γλώσσα, τις οποίες βέβαια τις αντλούμε από το json αρχείο.



3ο βήμα

- Και εδώ λοιπόν το πρώτο βήμα είναι να αναπτύξουμε την html σελίδα η οποία θα κρατάει τα δεδομένα που διαβάζουμε από το json αρχείο.
- Και εδώ προφανώς θα έχουμε μια μόνο σελίδα, η οποία καλείται κάθε φορά και συμπληρώνεται με τα καινούργια δεδομένα.

- Τα δεδομένα αυτή τη φορά βέβαια **καθορίζονται από το id** που περνάει ως μεταβλητή στο url.
- Ας δούμε λοιπόν πως…

- Για να μπορούμε να **προσπελάσουμε τις μεταβλητές** σε ένα url πρέπει αρχικά να εισάγουμε το url module:

- **const url = require('url’);**

- Στη συνέχεια λαμβάνουμε τη μεταβλητή που αποστέλλεται
- **var pathName=url.parse(path_name,true).pathname;**
- **var query=url.parse(path_name,true).query;**

Ίδια λογική με πριν!

```
//API
}else if(pathName==='/program'){
        //here we just have a single template where we need to replace eve
        //dataObject.programmingL is an array so we acceess the position t
        res.writeHead('404',{'Content-Type': 'text/html'});


        var programl=dataObject.programmingL[query.id];
        console.log(programl);
        const output= replaceTemplate(tempProgram,programl);

        res.end(output); //end the response
//NOT FOUND
}else{
```

Δημιουργία ενός module

- Μπορούμε επίσης να δημιουργήσουμε το δικό μας module το οποίο θα περιλαμβάνει τη μέθοδο replaceTemplate που είδαμε παραπάνω.

- Δημιουργούμε ένα .js αρχείο με τον παρακάτω κώδικα:

Δημιουργία ενός module

- Θέλουμε να εξάγουμε αυτή την anonymous function
  → Γι αυτό την αναθέτουμε στην ιδιότητα module.exports

```
module.exports=(temp, program ) => {
    var output = temp.replace(/{%NAME%}/g, program.name);
       output = output.replace(/{%STARS%}/g, program.stars);
       output = output.replace(/{%IMAGE%}/g, program.img);
       output = output.replace(/{%HISTORY%}/g, program.history);
       output = output.replace(/{%INFO%}/g, program.info);
       output = output.replace(/{%ID%}/g, program.id);

     if(!program.stars==5) {
        output = output.replace(/{%SENIOR%}/g, 'senior');
     }else{
        output = output.replace(/{%SENIOR%}/g, 'not-senior');

     }
    return output;
}
```

Δημιουργία ενός module

```
const http = require('http');
//const path = require('path/posix');
const url = require('url');
const filesystem=require('fs');
//import our own modules
//in require . means the location of the file,like dir name
const replaceTemplate= require('./modules/replaceTemplate')
```

To be continued…