

NodeJS assignments:

1. Print the following (use nested loop):

```
1
22
333
4444
55555
666666
7777777
88888888
99999999
```

2. Write a function nameLength() that accepts an array of employee names. Find and display those employee names, which are more than 4 characters.
For example : ["Joy", "Meena", "Anne", "Xi", "Veena"]
The following should get displayed : Meena Veena
3. Write a function min() that takes a array and returns minimum number in the set of arguments. . Do not use the Math predefined object

Local Modules:

- 1) Create a user defined local module greet.js containing a function greet() that greets user based on time of the day. If its morning, greet user as "Good morning", if its afternoon, greet user as "Good Afternoon" else as "Good Evening"

Create a client module that will bring in the greet.js module and invoke the greet function

- 2) Create a user defined local module calc.js that exposes the following functions: add(a,b), subtract(a,b), multiply(a,b), divide(a,b), square(a), min(a,b,c), max(a,b,c)
 - Create a client application that invokes each of these methods
 - Make use of Math predefined core object. Create a client application that invokes each of these methods
- 3) Create 3 user defined modules that deal with shapes: circle.js, rectangle.js, triangle.js
 - Circle.js has functions like : calcArea(radius), calcCircumference(radius), calcDiameter(radius)
 - Rectangle.js – calcArea(length, breadth), calcPerimeter(length, breadth)
 - Triangle.js – isEquilateral(side1, side2, side3), calcPerimeter()

File handling:

- 4) Create an array of names. Read the contents of array and save into a file – names.txt as a string delimited with | (pipe)
eg var arr = ["aaa", "bbb", "ccc"] must be stored in file as aaa|bbb|ccc
- 5) File handling: Assume I have a file emp.txt that contains employee data in the following format:
1001:Harry :Sales :23000
1002:Sarita :Accounts :20000
1003:Monika:TechSupport:35000

Read the file. Display total sum of salary of all employees. In this example, it must display 78000

- 6) Consider an array that holds 4 employee objects. Emp details could be empid, empname and salary.
Loop thru this Employee array and store the employee objects in a text file using same format as follows:

1001:Harry :Sales :23000
1002:Sarita :Accounts :20000
1003:Monika:TechSupport:35000
...

- 7) Assume a customer.json that has an array of customers with details like customer name, address, ph no, credit-rating etc for each customer. Read this file, parse the json and display all customer names along with their contact details

```
[
  {
    "custname":"Anil Patil",
    "address":"abd 123",
    "phno": "8877669988",
    "rating":8
  },
  {
    "custname":"Anita Kulkarni",
    "address":"A102, Highstreet",
    "phno": "99675456",
    "rating":7
  },
  {
    "custname":"Kavita Menon",
    "address":"BB203, Pune",
    "phno": "123456789",
    "rating":9
  }
]
```

- 8) Consider a json array object that describes an array of books with bookid, name, author and selling price. Loop thru array, calculate discount on each book and hence final price, convert each book json object into string in below format and save into a file book.txt
eg, assuming discount of 10% on all books:
Bookid|bookname|author|Sellingprice|finalprice
1001|Core Java|Cay Horstman|500|450

- 9) Create a simple text file. Now use node file handling to read each line from text file, prefix line with a number and display.

Sample text file:

Node is a JS runtime platform
Created with C++ and Javascript
Runs on the V8 engine
Created by Ryan Dhal

Output :

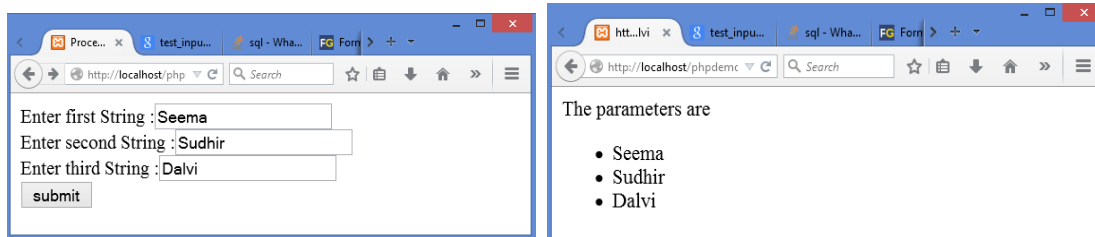
1. Node is a JS runtime platform
2. Created with C++ and Javascript
3. Runs on the V8 engine
4. Created by Ryan Dhal

Node HTTP and the web:

- 10) Write a browser based app. Everytime a http request comes in, server must send a greeting back to user depending on time of the day. Use the Greet.js module created earlier.

ExpressJS ass:

- 11) Create a html form that takes three strings and passes these parameters to ExpressJS application. This application must respond with HTML that displays all parameters in an unordered list:



- 12) Develop a web application as follows:
- Design **simpleInt.html** to capture Principal Amount, No of years and rate of Interest
 - A **ExpressJS application** will receive these numbers and will generate response with simple interest value to user
- 13) Develop a login app. The login.html must accept username and password and submit to node. At server side, create a Javascript array that contains username and password as comma separated values. Enter 3 user details.

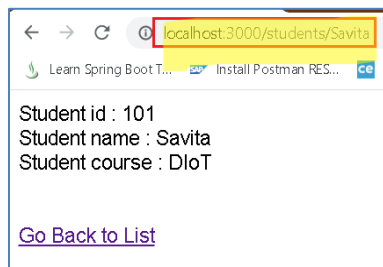
```
let users = [  
  {uname:"shrilata", pass:"secret"},  
  {uname:"admin", pass:"admin123"},  
  {uname:"user1", pass:"pass1"},  
]
```

Middleware must check if username and password is available in a Javascript array. If yes, send success.html, else failure.html

- 14) Create a ExpressJS application with an HTML that accepts username and password. HTML has a submit button called "Signup". On submitting the form, store the username and password details as a json object into an array
- 15) Create a ExpressJS application. At server side create an array of students with studentid, student name, course.

```
let studs = [  
  {sid:101, sname:"Savita", course:"DIoT"},  
  {sid:102, sname:"Kavita", course:"DAC"},  
  {sid:103, sname:"Anita", course:"DESD"},  
  {sid:104, sname:"Sunita", course:"DIoT"},  
  {sid:105, sname:"Babita", course:"DMC"},  
]
```

Use Restful URL to search on name (eg URL highlighted in below screenshot). On receiving name, search thru array and to find the correct student and return student details as shown below.



- 16) In the previous App, create another path (localhost:3000/students) to fetch all students. This will be my Rest endpoint.
- Create an ajax application that will connect to this endpoint, fetch all students details and display in HTML as a neat HTML table
 - Create another ajax application that has an HTML to accept student name. Concatenate this name to a get request URL and fetch and display that student details in browser

Next gen Javascript ass:

- Write a function transformToObjects() that transforms a list of numbers into a list of JavaScript objects. For the provided input [1, 2, 3] the transformToObjects() function should return [{val: 1}, {val: 2}, {val: 3}].

```
function transformToObjects(numberArray) {  
  // Todo: Add your logic  
  // should return an array of objects  
}
```

- Consider an array of products objects. For each product, we store productid, productname, price and category. Eg productArr = [{pid:1001, prodName:"Gaming headset",3000,"Headset"}....]
Using Javascript forEach, loop thru the product array and display contents in neat HTML table
- Consider the product array in last example. Iterate thru the array, filter out all products that have the category "toys" and put in a new array ToyArray. Display the array. Use Javascript's array.filter() for this