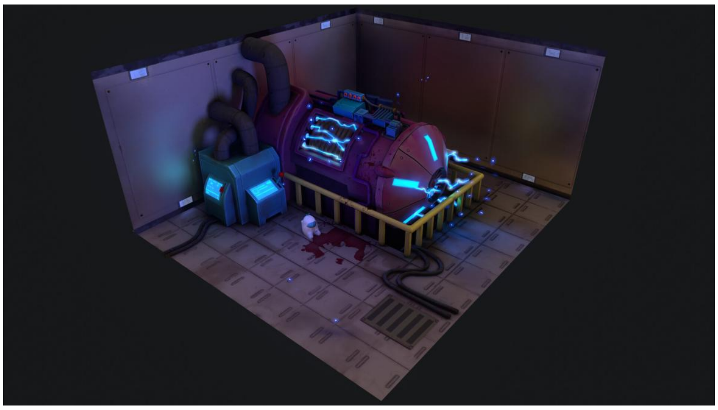
B1 - Unix & C Lab Seminar

B-CPE-101

Crewmate on an isolated planet

A javascript tutorial



Crewmate on an isolat

binary name: None
repository name: Your computer
language: everything working on “the dump“
build tool: Node

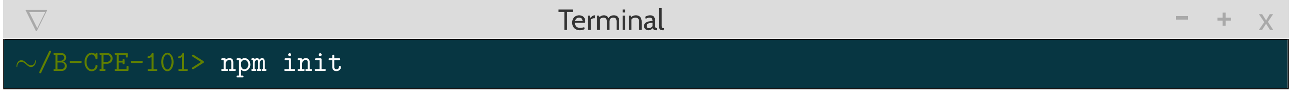
- The totality of your source files, except all useless files (binary, temp files, obj files,...), must be included in your delivery.
- All the bonus files (including a potential specific Makefile) should be in a directory named *bonus*.
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

Your crewmates mistook you for an alien!!! You were ejected from the mothership and left for dead, but you landed on an isolated planet full of danger. Luckily, you've found an old base where the system is powered by an old backup generator. The call for help system is unfortunately no longer functional, it's up to you to fix it!

There are no grammar or spelling mistakes in these lines, any mistakes you may see is due to the chip that the Rothschild family implemented in your brain during your childhood in order to make me look like an idiot.

BOOT ENGINE

First, you will have to restart the system, and start the procedures, for this, open the terminal and type the following command to create the package.json, this is this package that will make the link between your different .js file, and will manage your version and your libs.



```
~/B-CPE-101> npm init
```

The previous explorer (the skeleton in the corner of the room) already did it for you, this is just an indication in case you wanted to boot your own engine.

TASK 1 NANOBOT

You must use some Nanobot to repair the computer, their code is already implemented in NanoBot.js, the number stored in them is the power their use to moove, but be carefull! there is a NanoBot shutting zone where the power must be 0, assert the NanoBot present in this zone are equal to 0.

Ever heard of the difference between *let* and *var* declaration ?

TASK 2 BATTERY

Perfect ! It works just fine now ! But... There is no batteries in this machine, and worst, you found a report saying that only one of these batteries work for the S.O.S terminal, you must write a little program to know wich one.

Use the provided function Load to get the list of the batteries, there are all referenced the same, by a 1, but one of them, the good one have “ “ ““twist”“ “ “

Things always come in threes

TASK 3 BORKEN

Yay ! It's on, but now, you have to find the password! There is a paper indicating the list of int that composed it, but to protect from hacker there is a tricks, you must get rid of all the non-even number and **you can't use loop** (because thats the rule now, I make the rules...).

> *arr.filter* is a thing ??? Of course it is, Javascript is fantastic.

TASK 4 EMERGENCY CALL MANAGER

You can now finnaly use this D A M N machine...Or at least... you think so *evil laughter* you must repair the button to launch it, once it print "button clicked" when you click on the button, you should be good.

> You must use the button in the provided SOSTerminal.html, to test it, you can simply double click on it in your file folder, and the site will appear on your computer, FANTASTIC I TOLD YOU!

> No need to go on a complicated parse, there is a document variable you can use and which always exist in your js code if it is called in an html. What a language !

> getElementBy

TASK 5 S.O.S

You can now send S.O.S, but for that, you will all the infos of the captain near you to know who you can call. It's time to do some web request and so... some ASYNCHRONOUS.
the goal of this exercice, is make a call to the https://randomuser.me/api/ to get the infos of at least 10 person, and store them in an array.

> look at fetch command

API response can be slow… Maybe we need to *await* it before going further in our program

this is called asynchronous programming !

Get an error message when *awaiting* maybe because the word **async**chronous MUST be somewhere in your program

Like, in the prototype

async function [name of the function]… but you really force it…