

QCM

- Node package manager

Node Package Manager (NPM) is a command line tool that installs, updates or uninstalls **Node.js packages** in your application.

- Node module import

```
var x = require("moduleB");
```

- Express definition

Express.js est un **framework pour construire des applications web** basées sur Node.js.

- Package.json

A **package.json** is a **JSON** file that exists at the root of a Javascript/Node project. It holds **metadata** relevant to the project and it is used for managing the project's **dependencies, scripts, version and a whole lot more**

- Events

```
var events = require('events');
var EventEmitter = new events.EventEmitter();

//Create an event handler:
var myEventHandler = function () {
  console.log('I hear a scream!');
}

//Assign the event handler to an event:
eventEmitter.on('scream', myEventHandler);

//Fire the 'scream' event:
eventEmitter.emit('scream');
```

- Callback signature

```
function request (url, callback) {  
  // doing something callback (error, response, body);  
}
```

- Exact Package version

value	desc
~version	"Approximately equivalent to version" See npm semver - Tilde Ranges
^version	"Compatible with version" See npm semver - Caret Ranges
version	Must match version exactly
>version	Must be greater than version
>=version	etc
<version	
<=version	
1.2.x	1.2.0, 1.2.1, etc., but not 1.3.0
*	Matches any version
latest	Obtains latest release

- Global class

```
var Globals = { 'domain': 'www.MrGlobal.com'; }  
module.exports = Globals;  
Then if you want to use these, use require.  
var globals = require('globals'); // << globals.js path globals.domain // << Doma
```

The Buffer **object is a global object in Node.js, and it is not necessary to import it using the `require` keyword.**

Callback argument

`callback (error, response, body);`

- Event loop

The event loop is **what allows Node.js to perform non-blocking I/O operations** — despite the fact that JavaScript is single-threaded — by offloading operations to the system kernel whenever possible.

- Node package.json
- Route matching Url

```
app.use(function(req, res, next) {  
  if (!req.url.match(/aV*/g)) {  
    res.sendFile("index.html"); }  
  else { next(); } });  
  
var router = express.Router()  
; router.get(...)  
  router.post(...)  
app.use('/a', router);
```

- Testframework

`mocha`

- Request params

[req.params](#) contains route parameters (in the path portion of the URL), and [req.query](#) contains the URL query parameters (after the ? in the URL).

You can also use [req.param\(name\)](#) to look up a parameter in both places (as well as [req.body](#)), but this method is now deprecated.
`req.param('<param_name>')`

- Command line argument

Npm install

aliases: `npm i`, `npm add`

common options:

```
[-P|--save-prod|-D|--save-dev|-O|--save-optional|--save-peer]
[-E|--save-exact] [-B|--save-bundle] [--no-save] [--dry-run]
```

- Node threading

js follows **Single-Threaded** with Event Loop Model inspired by JavaScript **Event-based model with JavaScript callback mechanism**. So, **node.js** is **single-threaded** similar to JavaScript but not purely JavaScript code which implies things that are done asynchronously like network calls, file system tasks, DNS lookup, etc.

- Promosify

- cookies

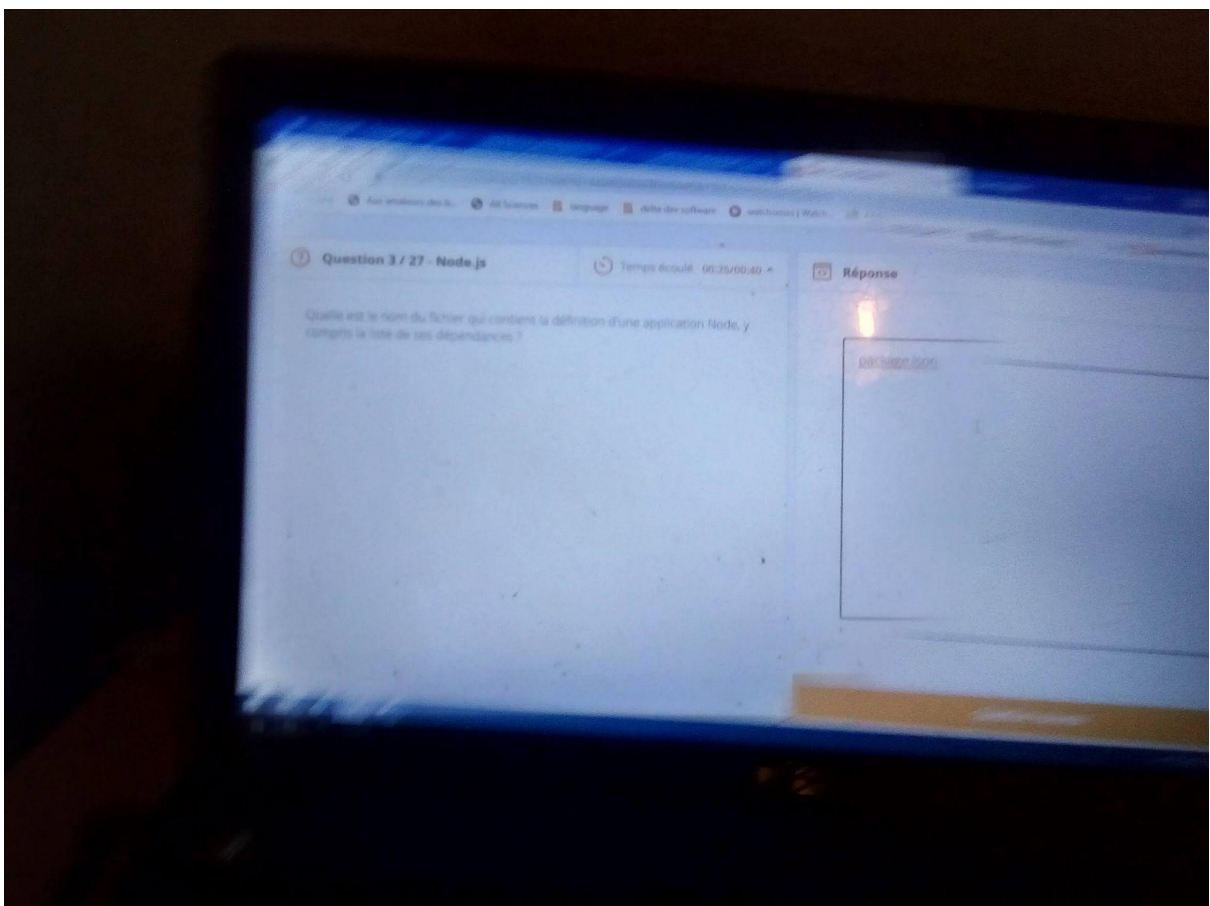
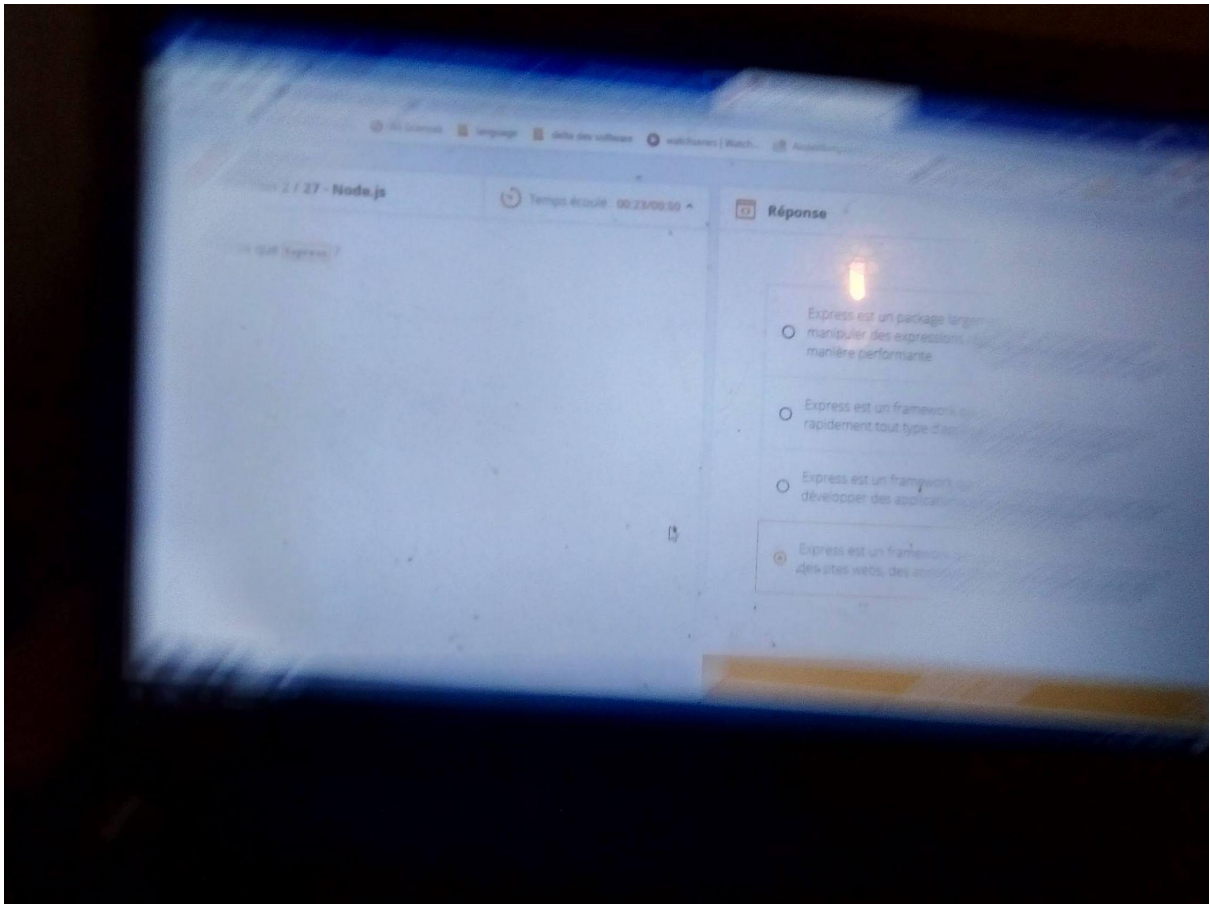
```
const app = require('express')();

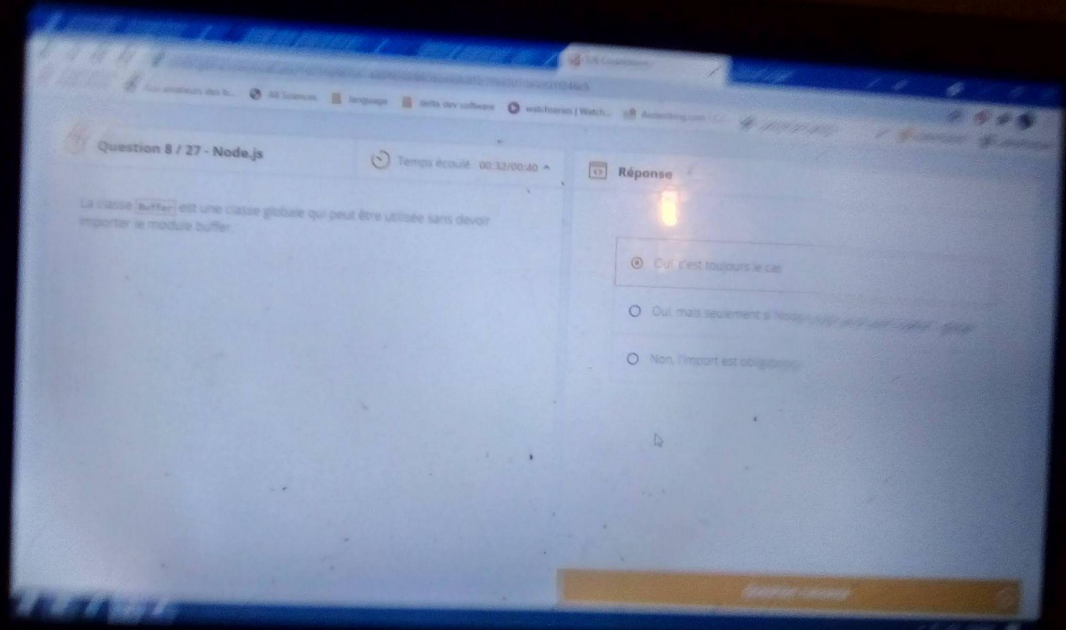
app.use('/', (req, res) => {
  var cookie = getcookie(req); console.log(cookie);
});

function getcookie(req) {
  var cookie = req.headers.cookie;
  // user=someone; session=QyhYzXhkTZawlb5qSI3KKyPVN (this is my cookie i get)

  return cookie.split('; ');
}

output
['user=someone', 'session=QyhYzXhkTZawlb5qSI3KKyPVN']
```





Question 8 / 27 - Node.js

Temps écoulé : 00:32/00:40

La classe `Buffer` est une classe globale qui peut être utilisée sans devoir importer le module `buffer`.

Réponse

- ☒ Oui, c'est toujours le cas
- ☐ Oui, mais seulement si Node.js est installé globalement
- ☐ Non, l'import est obligatoire

Valider la réponse

