Name - Sandarbh Singhal   Role - Full Stack Web Developer

Node.js CheatSheet

Event - Driven Architecture

Node js follows an event - driven architecture. Events trigger callbacks.

○ Handle multiple Connections simultaneously

when one connections is waiting for data, Node.js can handle other requests without blocking.

○ Perform I/o operations efficiently.

Node.js doesn't wait for I/o to finish before moving on. It triggers a callback once the operation completes.

Asynchronous Programming

Node.js is asynchronous by default. Code execution continues without waiting for I/o operations or functions

• Callbacks - functions passed to be executed when an event occurs.

Example:

```
function getUser (id, callback){
        setTimeout (() => {
                console. log ('Reading from a db');
                callback ( {id: id, username : "Sandy" });
        });
}
```

• Promises - Objects representing the eventual completion of an asynchronous operations.

Example

```
const p = new Promise((resolve, reject) => {
    setTimeout(() => {
        resolve(1);
    }, 2000);
});

P
    .then(result => console.log('Result', result));
    .catch(err => console.log(error.message));
```

○ **Async/Await**

Syntatic sugar to write asynchronous code in a more synchronous like style.

Example:

```
async function display() {
    try {
        const user = await getUser(1);
        console.log(user);
    }
    catch(err) {
        console.log(err);
    }
}
```

npm

npm is the default package manager for Node.js. It allows you to →

- Install packages → Reusable module of code published by Node.js
- Manage dependencies → Packages required by your project.
- Publish your own packages → Share your code with the Node.js community.

Modules

Modules are self-contained units of code that can be reused throughout your application.

CommonJS Modules

- module.exports : Used to export code from a module.
- require : Used to import code from other modules.

Example : file1.js

```
function sayHello(){
        console.log("Hello!");
}
module.exports = sayHello;
```

file2.js

```
const sayHello = require('./file1');
sayHello();
```