

Node.jsモジュール 段階的課題 解答例

レベル1：CommonJSの基本

課題1：モジュールのインポート

```
// math.js (別のファイル)
exports.add = (a, b) => a + b;

// main.js
const math = require('./math');
console.log(math.add(5, 3)); // 8
```

解説: require('./ファイル名')でローカルモジュールをインポート。

課題2：モジュールのエクスポート

```
// greet.js
exports.sayHello = (name) => `Hello, ${name}!`;

// main.js
const greet = require('./greet');
console.log(greet.sayHello('World'));
```

解説: exports.プロパティ名 = 値 でエクスポート。

課題3：複数のエクスポート

```
// utils.js
exports.PI = 3.14159;
exports.sum = (arr) => arr.reduce((a, b) => a + b, 0);
exports.average = (arr) => exports.sum(arr) / arr.length;

// main.js
const utils = require('./utils');
console.log(utils.PI); // 3.14159
console.log(utils.sum([1, 2, 3, 4])); // 10
console.log(utils.average([1, 2, 3, 4])); // 2.5
```

解説: 複数のプロパティをexportsに追加。

レベル2 : ES Moduleの基本

課題4 : ES Moduleのインポート

```
// math.mjs
export const multiply = (a, b) => a * b;

// main.mjs
import { multiply } from './math.mjs';
console.log(multiply(4, 5)); // 20
```

解説: import { 名前 } from 'ファイル' で名前付きインポート。

課題5 : ES Moduleのエクスポート

```
// greet.mjs
export const sayHello = (name) => `Hello, ${name}!`;

// main.mjs
import { sayHello } from './greet.mjs';
console.log(sayHello('World'));
```

解説: export const 名前 = 値 でエクスポート。

課題6 : デフォルトエクスポート

```
// config.mjs
const config = { port: 3000 };
export default config;

// main.mjs
import config from './config.mjs';
console.log(config.port);
```

解説: export default でデフォルトエクスポート、インポート時は名前を付けて。

レベル3 : npmの基本

課題7 : npm init

```
npm init -y
```

解説: package.jsonが作成される。

課題8 : パッケージのインストール

```
npm install lodash
```

```
const _ = require('lodash');  
console.log(_.chunk([1, 2, 3, 4], 2)); // [[1,2], [3,4]]
```

解説: npm install でパッケージをインストール。

課題9 : package.jsonの編集

```
{  
  "scripts": {  
    "start": "node index.js"  
  }  
}
```

```
npm start
```

解説: scriptsにコマンドを追加。

レベル4 : モジュールの高度な使い方

課題10 : module.exports

```
// library.js  
module.exports = {  
  add: (a, b) => a + b,  
  subtract: (a, b) => a - b,  
  multiply: (a, b) => a * b,  
  divide: (a, b) => a / b  
};  
  
// calc.js  
const library = require('./library');  
console.log(library.add(10, 5)); // 15
```

```
console.log(library.subtract(10, 5)); // 5
console.log(library.multiply(10, 5)); // 50
console.log(library.divide(10, 5)); // 2
```

解説: module.exports = オブジェクト でエクスポート。

課題11：条件付きエクスポート

```
// config.js
if (process.env.NODE_ENV === 'production') {
  module.exports = { apiUrl: 'https://api.prod.com' };
} else {
  module.exports = { apiUrl: 'http://localhost:3000' };
}
```

解説: 条件に応じて異なるものをエクスポート。

課題12：サードパーティモジュールの使用

```
const axios = require('axios');

axios.get('https://api.github.com/users/octocat')
  .then(response => console.log(response.data))
  .catch(error => console.error(error));
```

解説: npmでインストールしたaxiosを使ってAPIコール。

レベル5：ファイルシステムモジュール

課題13：ファイルの読み込み

```
const fs = require('fs');

fs.readFile('example.txt', 'utf8', (err, data) => {
  if (err) throw err;
  console.log(data);
});
```

解説: fs.readFileでファイルを非同期読み込み。

課題14：ファイルの書き込み

```
const fs = require('fs');

fs.writeFile('output.txt', 'Hello World!', (err) => {
  if (err) throw err;
  console.log('File written!');
});
```

解説: fs.writeFileでファイルに書き込み。

課題15 : ディレクトリの操作

```
const fs = require('fs');

fs.mkdir('newdir', (err) => {
  if (err) throw err;
  console.log('Directory created!');
});
```

解説: fs.mkdirでディレクトリ作成。

レベル6 : カスタムモジュールの作成

課題16 : 計算機モジュール

```
// calculator.js
exports.add = (a, b) => a + b;
exports.subtract = (a, b) => a - b;
exports.multiply = (a, b) => a * b;
exports.divide = (a, b) => a / b;

// main.js
const calc = require('./calculator');
console.log(calc.add(1, 2));
```

解説: 計算関数を集めたモジュール。

課題17 : ユーティリティモジュール

```
// utils.js
exports.capitalize = (str) => str.charAt(0).toUpperCase() + str.slice(1);
exports.reverse = (arr) => arr.slice().reverse();
```

```
// main.js
const utils = require('./utils');
console.log(utils.capitalize('hello'));
```

解説: 再利用可能なユーティリティ関数。

課題18：設定モジュール

```
// settings.mjs
export const database = {
  host: 'localhost',
  port: 5432,
  name: 'myapp'
};

export const server = {
  port: 3000,
  timeout: 5000
};

// app.mjs
import { database, server } from './settings.mjs';
console.log(database);
console.log(server);
```

解説: 設定値をエクスポート。

レベル7：モジュールのテストとデバッグ

課題19：モジュールのテスト

```
const calc = require('./calculator');

console.log('Test add:', calc.add(2, 3) === 5);
console.log('Test subtract:', calc.subtract(5, 3) === 2);
```

解説: 関数をテスト。

課題20：総合課題

```
const fs = require('fs');
const _ = require('lodash');
```

```
fs.readFile('data.json', 'utf8', (err, data) => {  
  if (err) throw err;  
  const json = JSON.parse(data);  
  const chunked = _.chunk(json, 2);  
  console.log(chunked);  
});
```

解説: fsとlodashを組み合わせたプログラム。

c:\Users\user\nodejs\3.nodejs_modules\exercise\exercise_answer.md