

# Node.js構文 段階的課題 20問

## レベル1：変数とスコープ (課題1-3)

### 課題1：変数宣言の基本

次の要件を満たすプログラムを作成してください。

- `const` を使って自分の名前を `myName` という変数に代入
- `let` を使って年齢を `age` という変数に代入
- `age` を1増やして、両方の値を`console.log`で出力

**ポイント:** `const` と `let` の違い

### 課題2：スコープの理解

次のコードを実行するとどうなるか予想し、実際に実行して確認してください。

```
if (true) {  
  var x = 10;  
  let y = 20;  
  const z = 30;  
}  
console.log(x);  
console.log(y);  
console.log(z);
```

**ポイント:** `var`, `let`, `const` のスコープの違い

### 課題3：変数の再代入

次の要件を満たすプログラムを作成してください。

- `let score = 0;` からスタート

- scoreに10を加算
- scoreに5を掛ける
- 最終的なscoreの値を出力

**ポイント:** let の再代入

## レベル2：演算子とデータ型 (課題4-6)

### 課題4：計算プログラム

次の要件を満たすプログラムを作成してください。

- 2つの数値（例：7と3）を変数に代入
- 加算、減算、乗算、除算の結果をそれぞれ出力
- それぞれの計算結果が10より大きいかどうかを判定して出力

**ポイント:** 算術演算子と比較演算子

### 課題5：型の確認

次の値の型を typeof を使って確認し、出力してください。

- "JavaScript"
- 42
- true
- null
- undefined
- []
- {name: "Taro"}

**ポイント:** typeof演算子（nullの結果に注目）

### 課題6：等価演算子の違い

次のコードを実行し、== と === の違いを確認してください。

```
const num = 10;  
const str = "10";  
  
console.log(num == str);  
console.log(num === str);
```

なぜこのような結果になるか説明してください。

**ポイント:** == と === の違い

## レベル3：文字列とテンプレートリテラル (課題7-8)

### 課題7：自己紹介プログラム

次の要件を満たすプログラムを作成してください。

- 名前、年齢、趣味を変数に代入
- テンプレートリテラルを使って「私の名前は〇〇です。年齢は〇〇歳で、趣味は〇〇です。」と出力

**ポイント:** テンプレートリテラル（バッククォート）

### 課題8：複数行の文字列

次の要件を満たすプログラムを作成してください。

- 3行の住所（郵便番号、都道府県、市区町村）をテンプレートリテラルで作成
- それぞれの行を改行して出力

**ポイント:** \n とテンプレートリテラルでの改行

# レベル4：オブジェクトの基本 (課題9-11)

## 課題9：学生オブジェクト

次の要件を満たすプログラムを作成してください。

- 学生の情報（name, age, grade, club）を持つオブジェクトを作成
- ドット記法で名前を取得
- ブラケット記法で学年を取得
- 両方を出力

**ポイント:** オブジェクトリテラルとアクセス方法

## 課題10：ネストしたオブジェクト

次の要件を満たすプログラムを作成してください。

- 会社の情報を持つオブジェクトを作成
  - name: 会社名
  - address: オブジェクト（prefecture, city）
  - employees: 従業員数
- address.cityの値を取得して出力

**ポイント:** ネストしたオブジェクトへのアクセス

## 課題11：オブジェクトのプロパティ省略記法

次の要件を満たすプログラムを作成してください。

- 変数 title、author、year を定義
- これらを使ってbookオブジェクトを作成（プロパティ省略記法を使用）
- オブジェクトを出力

**ポイント:** プロパティ名の省略記法

# レベル5：配列の操作 (課題12-14)

## 課題12：配列の基本操作

次の要件を満たすプログラムを作成してください。

- 5つの数値を持つ配列を作成
- 配列の長さを出力
- 配列の最初と最後の要素を出力
- すべての要素を順番に出力（forループ使用）

**ポイント:** 配列の基本とforループ

## 課題13：配列のmap

次の配列があります。

```
const prices = [100, 200, 300, 400, 500];
```

この配列の各要素に消費税10%を加えた新しい配列を作成し、出力してください。

**ポイント:** mapメソッド

## 課題14：配列のfilter

次の配列があります。

```
const scores = [45, 78, 92, 35, 88, 60, 95, 42];
```

この配列から60点以上の点数だけを抽出した新しい配列を作成し、出力してください。

**ポイント:** filterメソッド

# レベル6：関数 (課題15-17)

## 課題15：関数の基本

次の要件を満たす関数を作成してください。

- 関数名: `calculateArea`
- 引数: `width` (幅), `height` (高さ)
- 戻り値: 長方形の面積
- 関数を呼び出して結果を出力

**ポイント:** 関数宣言と戻り値

## 課題16：アロー関数

課題15で作成した関数をアロー関数で書き直してください。

さらに、引数が1つで、その値を2倍にするアロー関数も作成してください（省略記法を使用）。

**ポイント:** アロー関数の記法

## 課題17：デフォルト引数

次の要件を満たす関数を作成してください。

- 関数名: `greet`
- 引数: `name` (デフォルト値は"`Guest`")
- 戻り値: "`Hello, ○○!`"という挨拶文
- 引数ありとなしで2回呼び出して結果を確認

**ポイント:** デフォルト引数

# レベル7：高度な配列操作とスプレッド構文 (課題18-19)

## 課題18：スプレッド構文で配列を結合

次の要件を満たすプログラムを作成してください。

- 2つの配列 `fruits1 = ['apple', 'banana']` と `fruits2 = ['orange', 'grape']`
- スプレッド構文を使ってこれらを結合した新しい配列を作成
- さらに 'melon' を追加
- 結果を出力

**ポイント:** スプレッド構文 (...)

## 課題19：分割代入

次の配列があります。

```
const userInfo = ['田中太郎', 25, 'Tokyo', 'developer'];
```

分割代入を使って、名前、年齢、都市、職業をそれぞれ別の変数に代入し、出力してください。

**ポイント:** 配列の分割代入

# レベル8：総合課題 (課題20)

## 課題20：学生管理システム

次の要件を満たすプログラムを作成してください。

1. 複数の学生情報を持つ配列を作成
  - 各学生は { name, age, score } の形式
  - 最低5人分のデータ
2. 次の関数を作成：
  - `getPassStudents` : 60点以上の学生だけを返す関数
  - `getAverageScore` : 全学生の平均点を返す関数
  - `formatStudentInfo` : 学生情報を「名前: ○○, 年齢: ○○歳, 点数: ○○点」の形式で返す関数

3. 以下を実行して結果を出力：

- 合格者（60点以上）のリストを表示
- 全体の平均点を表示
- 合格者の情報をformatStudentInfo関数を使って整形して表示

**ポイント:** これまでの知識の総合応用

## 補足情報

### 学習の進め方

1. 各課題を順番に取り組んでください
2. エラーが出たら、エラーメッセージをよく読んで原因を考えてください
3. わからない場合は、該当するサンプルコード（2.nodejs\_syntax内）を参照してください
4. 余裕があれば、課題をアレンジして自分なりの機能を追加してみてください

### 参考：各課題と対応するサンプルコードの関連

- 課題1-3: 2.2.1/
- 課題4-6: 2.2.2/, 2.2.3/
- 課題7-8: 2.2.3/02.js
- 課題9-11: 2.2.4/
- 課題12-14: 2.2.5/, 2.2.6/
- 課題15-17: 2.4/, 2.6/
- 課題18-19: 2.5.1/, 2.5.2/
- 課題20: 総合