

# 関数練習問題

## Ver 1.2

Seiichi Nukayama

2024 年 3 月 11 日

### 目次

1	三角形の面積	1
2	円の面積	1
3	階乗を求める	2
4	肥満度判定プログラム	3
5	変数の有効範囲 (1)	4
6	変数の有効範囲 (2)	4

## 1 三角形の面積

次のように呼び出すことのできる関数 triangle() を作成してください。

1. パラメータ (引数) として、(1) 底辺の長さ (cm)、(2) 高さ (cm) を渡す。
2. リターン (戻り値) として、三角形の面積 (cm2) を返す。

関数を呼び出すコード

```
1 'use strict';
2
3 const bottom = parseFloat(window.prompt('底辺の長さは?(cm)'));
4 const height = parseFloat(window.prompt('高さは?(cm)'));
5 const area = (1)
6 window.alert('三角形の面積は ' + area + '平方センチメートル');
```

(1) のところに、適切なコードを書いてください。

## 2 円の面積

次のように呼び出すことのできる関数 circle を作成してください。

1. パラメータ (引数) として、半径の長さ (cm) を渡す。
2. リターン (戻り値) として、円の面積を返す。
3. 円周率は Math.PI を使うこと。

関数 circle を呼び出すコード

```
1 'use strict';
2
3 const radius = parseFloat(window.prompt('半径は?(cm)'));
4 const area = (1)
5 window.alert('半径' + radius + 'cmの面積は ' + area + '平方センチメートル');
```

(1) のところに、適切なコードを書いてください。

### 3 階乗を求める

次のように呼び出すことのできる関数 fact を作成してください。

1. パラメータ (引数) として、整数 (n) を渡す。
2. リターン (戻り値) として、n の階乗 (n!) を返す。

たとえば、6 の階乗  $6! = 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 720$  です。

関数 fact を呼び出すコード

```
1 'use strict';  
2  
3 const num = parseInt(window.prompt('整数を入力してください'));  
4 const ans = (1);  
5 window.alert(`${num} の階乗は ${ans} です。');`;
```

(1) のところに、適切なコードを書いてください。

## 4 肥満度判定プログラム

BMI 指数というものがあります。以下の式によって求めることができます。

$$\text{BMI 指数} = \frac{\text{体重 (kg)}}{\text{身長 (m)} \times \text{身長 (m)}}$$

この BMI 指数の値によって、以下のように肥満度を求めることができます。

bmi	評価
18.5 未満	低体重
18.5 以上 25.0 未満	普通体重
25.0 以上 30.0 未満	肥満 1 度
30.0 以上 35.0 未満	肥満 2 度
35.0 以上 40.0 未満	肥満 3 度
40.0 以上	肥満 4 度

次のように呼び出すことのできる関数 `getLevel()` を作成してください。

1. パラメータ (引数) として、身長 (cm) と 体重 (Kg) を受け取る。
2. リターン (戻り値) として、“低体重” ~ “肥満 4 度” の文字列を返す。

この関数は、以下のように呼び出すことができます。

```
1 'use strict';
2
3 const height = parseFloat(window.prompt('身長(cm)'));
4 const weight = parseFloat(window.prompt('体重(kg)'));
5 const level = (1)
6 window.alert('あなたは ' + level + ' です。');
```

(1) のところに適切なコードを書いてください。

## 5 変数の有効範囲 (1)

次のコードの (1) ~ (3) に表示される値を答えなさい。

```
1 'use strict';
2
3 function twice(okane) {
4   okane = okane * 2;
5   console.log(okane);      // (1)
6   return okane;
7 }
8
9 let money = 1000;
10 money = twice(money);
11 console.log(money);        // (2)
12 console.log(okane);        // (3)
```

## 6 変数の有効範囲 (2)

次のコードの (1) ~ (2) に表示される値を答えなさい。

```
1 'use strict'
2
3 function twice() {
4   console.log(money);      // (1)
5   money = money * 2;
6 }
7
8 let money = 1000;
9 twice();
10 console.log(money);        // (2)
```