

# 関数の作成

2026年新歓

# 関数を使う場所、 効能

関数を使うときの方針

ひとまとまりの手続きは、一つの「部品」としてまとめる

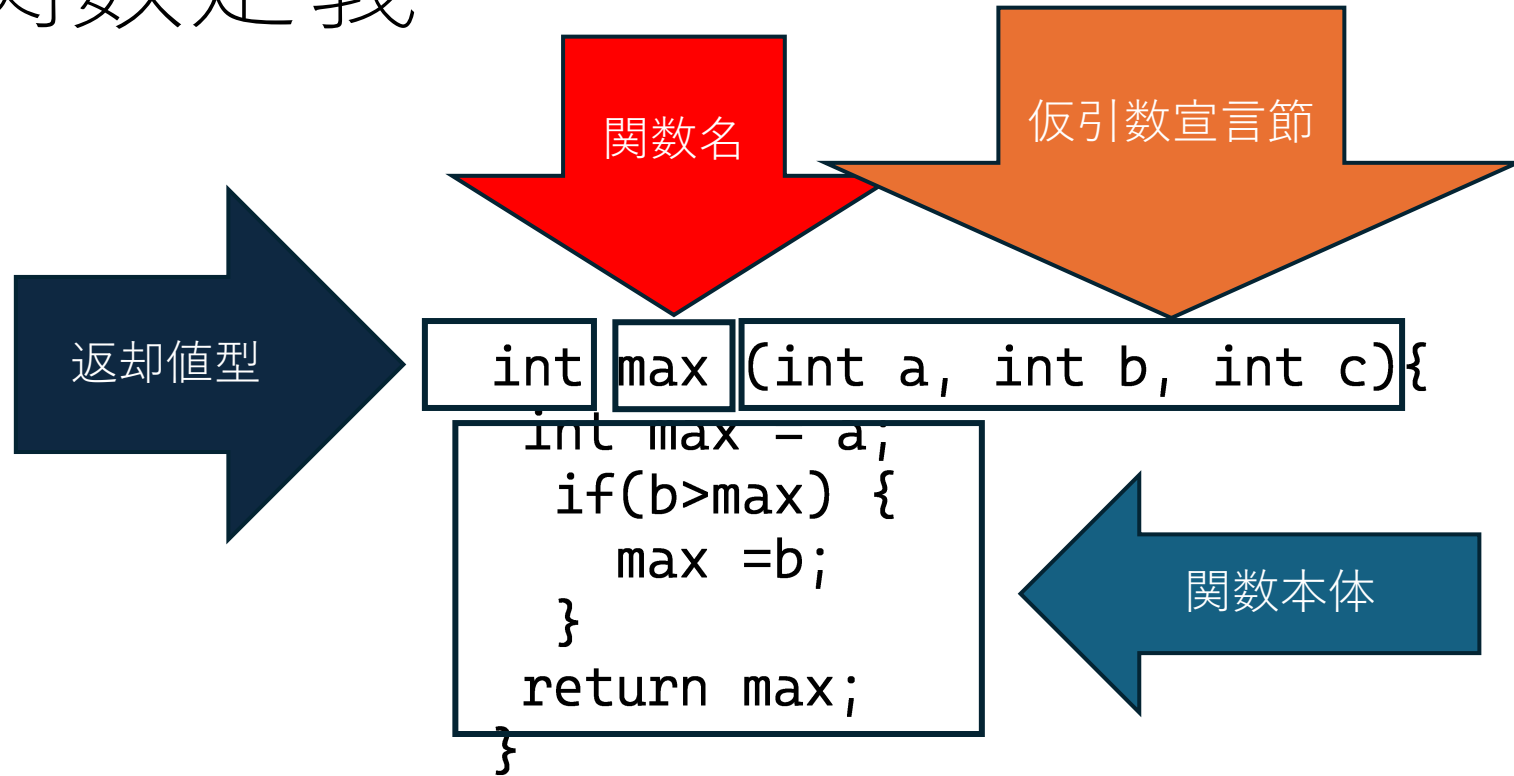
関数を使った方が便利なシーン

```
1  const int ledPin = 13;
2
3  // 点滅用の関数
4  void blinkLED(int times, int interval) {
5      for (int i = 0; i < times; i++) {
6          digitalWrite(ledPin, HIGH);
7          delay(interval);
8          digitalWrite(ledPin, LOW);
9          delay(interval);
10     }
11 }
12
13 void setup() {
14     pinMode(ledPin, OUTPUT);
15 }
16
17 void loop() {
18     blinkLED(3, 200); // 3回、200msで点滅
19     delay(1000);
20
21     blinkLED(5, 100); // 5回、100msで点滅
22     delay(2000);
23 }
```

```
1  const int ledPin = 13;
2
3  void setup() {
4      pinMode(ledPin, OUTPUT);
5  }
6
7  void loop() {
8      // 3回点滅
9      digitalWrite(ledPin, HIGH);
10     delay(200);
11     digitalWrite(ledPin, LOW);
12     delay(200);
13
14     digitalWrite(ledPin, HIGH);
15     delay(200);
16     digitalWrite(ledPin, LOW);
17     delay(200);
18
19     digitalWrite(ledPin, HIGH);
20     delay(200);
21     digitalWrite(ledPin, LOW);
22     delay(200);
23
24     delay(1000);
25
26     // 5回点滅
27     digitalWrite(ledPin, HIGH);
28     delay(100);
29     digitalWrite(ledPin, LOW);
30     delay(100);
31
32     digitalWrite(ledPin, HIGH);
33     delay(100);
34     digitalWrite(ledPin, LOW);
35     delay(100);
36
37     digitalWrite(ledPin, HIGH);
38     delay(100);
39     digitalWrite(ledPin, LOW);
40     delay(100);
41
42     digitalWrite(ledPin, HIGH);
43     delay(100);
44     digitalWrite(ledPin, LOW);
45     delay(100);
46
47     digitalWrite(ledPin, HIGH);
48     delay(100);
49     digitalWrite(ledPin, LOW);
50     delay(100);
51
52     delay(2000);
53 }
```

同じ動作でも簡潔かつ分かりやすくなる

# 関数定義



# 返却値型とは

関数が返す値である返却値の型です

型の種類

int型 -32768~32767の整数の値を返す  
返す値がint型である

注意！

return を書いて返さないとエラーがでます  
逆に返さない型もある→void型

```
1 # include <Arduino.h>
2
3 int addNumbers(int x, int y) {
4     int result = x + y;
5     return result; // 計算した結果を返す
6 }
7
8 void setup() {
9     Serial.begin(9600);
10 }
11
12 void loop() {
13     int a = 5;
14     int b = 10;
15
16     // 自作関数 addNumbers を呼び出し、結果を sum に代入
17     int sum = addNumbers(a, b);
18
19     Serial.print("合計は: ");
20     Serial.println(sum);
21
22     delay(2000);
23 }
24
```

## void型 値を返さない型

何も返さないが、値を更新することはできる

Ex. ある変数を事前に定義しておき（グローバル変数）、ある関数を呼んで実行すると更新ができる。

```
1  #include <Arduino.h>
2
3  const int ledPin = 13;
4
5  // void型の関数定義：指定された回数だけ点滅させる
6  void flashLED(int times, int wait) {
7      for (int i = 0; i < times; i++) {
8          digitalWrite(ledPin, HIGH);
9          delay(wait);
10         digitalWrite(ledPin, LOW);
11         delay(wait);
12     }
13     // return 文は不要（書いても良いが、値を返せない）
14 }
15
16 void setup() {
17     pinMode(ledPin, OUTPUT);
18 }
19
20 void loop() {
21     // 自作関数 flashLED を呼び出す（値を返さないなので代入はしない）
22     flashLED(3, 200);
23
24     delay(2000);
25 }
26
27
```

# 仮引数宣言節とは

()で囲まれた部分は、補助的な指示を受けるための変数である**仮引数**の宣言節

引数として、値のやり取りが行われるメカニズムのことを、値渡しと呼ぶ

関数間の引数の受け渡しは、値渡しによって行われる

注意！実引数との違い

```
1  #include <Arduino.h>
2
3  const int ledPin = 13;
4
5  // void型の関数定義：指定された回数だけ点滅させる
6  void flashLED(int times, int wait) {←仮引数
7      for (int i = 0; i < times; i++) {
8          digitalWrite(ledPin, HIGH);
9          delay(wait);
10         digitalWrite(ledPin, LOW);
11         delay(wait);
12     }
13     // return 文は不要（書いても良いが、値を返せない）
14 }
15
16 void setup() {
17     pinMode(ledPin, OUTPUT);
18 }
19
20 void loop() {
21     // 自作関数 flashLED を呼び出す（値を返さないなので代入はしない）
22     flashLED(3, 200);←実引数
23
24     delay(2000);
25 }
26
27
```