

# owl関数(p136)の補足

プログラム

```
void setup() {  
  size(480, 120);  
}  
void draw() {  
  background(176, 204, 226);  
  owl(110, 110);  
  owl(180, 110);  
}  
void owl(int x, int y) {  
  途中省略  
}
```

# owl関数の動作1

プログラム

```
void setup() {  
    size(480, 120);  
}  
void draw() {  
    background(176, 204, 226);  
    owl(110, 110);  
    owl(180, 110);  
}  
void owl(int x, int y) {  
    途中省略  
}
```

setup関数が実行され、その中のsize関数が実行される。

## owl関数の動作2

プログラム

```
void setup() {  
  size(480, 120);  
}  
void draw() {  
  background(176, 204, 226);  
  owl(110, 110);  
  owl(180, 110);  
}  
void owl(int x, int y) {  
  途中省略  
}
```

次にdraw関数が実行され、  
background関数が実行される。

# owl関数の動作3

プログラム

```
void setup() {  
  size(480, 120);  
}  
void draw() {  
  background(176, 204, 226);  
  owl(110, 110);  
  owl(180, 110);  
}  
void owl(int x, int y) {  
  途中省略  
}
```

次にowl関数が実行される。

# owl関数の動作4

プログラム

```
void setup() {  
  size(480, 120);  
}
```

```
void draw() {  
  background(176, 204, 226);  
  owl(110, 110); 実引数  
  owl(180, 110);  
}
```

```
void owl(int x, int y) {  
  途中省略 仮引数  
}
```

実引数の値110と110が  
仮引数xとyに代入される。

# owl関数の動作5

プログラム

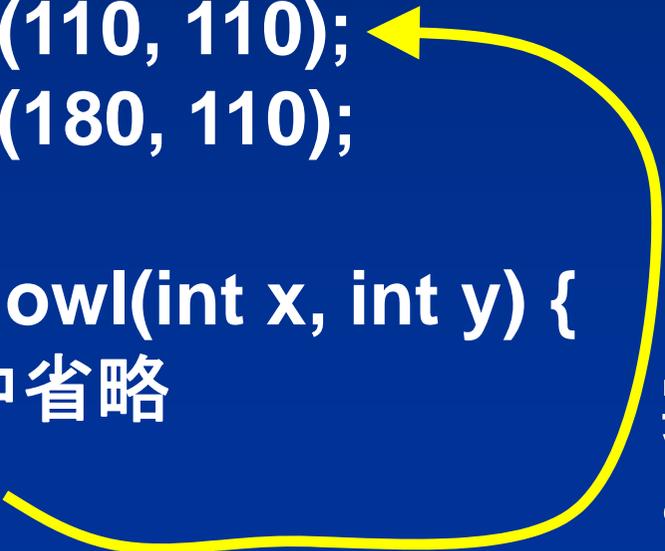
```
void setup() {  
  size(480, 120);  
}  
void draw() {  
  background(176, 204, 226);  
  owl(110, 110);  
  owl(180, 110);  
}  
void owl(int x, int y) {  
  途中省略  
}
```

仮引数xとyの値に従って  
フクロウが表示される。

# owl関数の動作6

プログラム

```
void setup() {  
  size(480, 120);  
}  
void draw() {  
  background(176, 204, 226);  
  owl(110, 110);  
  owl(180, 110);  
}  
void owl(int x, int y) {  
  途中省略  
}
```



実行がowl関数を呼び出した  
ところに戻る。

# owl関数の動作7

プログラム

```
void setup() {  
  size(480, 120);  
}
```

```
void draw() {  
  background(176, 204, 226);  
  owl(110, 110);  
  owl(180, 110);  
}
```

```
void owl(int x, int y) {  
  途中省略  
}
```

次のowl関数が実行される。

# owl関数の動作8

プログラム

```
void setup() {  
  size(480, 120);  
}  
void draw() {  
  background(176, 204, 226);  
  owl(110, 110);  
  owl(180, 110); 実引数  
}  
void owl(int x, int y) {  
  途中省略 仮引数  
}
```



実引数の値180と110が  
仮引数xとyに代入される。

# owl関数の動作9

プログラム

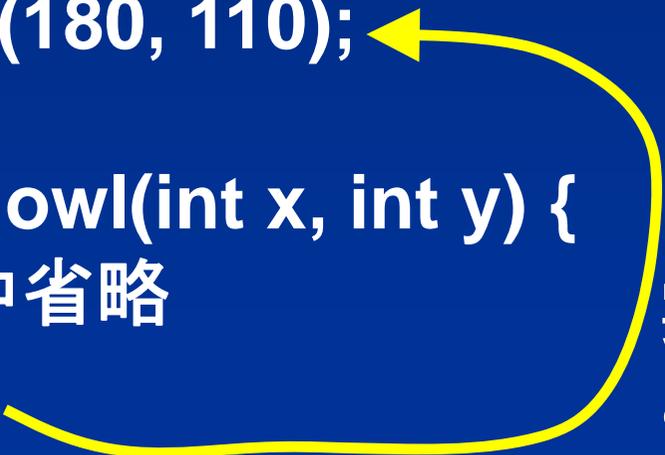
```
void setup() {  
  size(480, 120);  
}  
void draw() {  
  background(176, 204, 226);  
  owl(110, 110);  
  owl(180, 110);  
}  
void owl(int x, int y) {  
  途中省略  
}
```

仮引数xとyの値に従って  
フクロウが表示される。

# owl関数の動作10

プログラム

```
void setup() {  
  size(480, 120);  
}  
void draw() {  
  background(176, 204, 226);  
  owl(110, 110);  
  owl(180, 110);  
}  
void owl(int x, int y) {  
  途中省略  
}
```



実行がowl関数を呼び出した  
ところに戻る。

# 値を返さない関数の見分け方

プログラム

```
void setup() {  
  size(480, 120);  
}
```

```
void draw() {  
  background(176, 204, 226);  
  owl(110, 110);  
  owl(180, 110);  
}
```

```
void owl(int x, int y) {  
  途中省略  
}
```

関数名の前に**void**がある場合  
値を返さない関数になります。  
**void**は返却値が無いことを  
意味します。