

# プログラミング演習I

## 第4回: 変数

担当: 小久保温

# 変数

## ▶ 値を取っておくのに使う

- ▶ 同じ値を何回もくり返し使う場合に便利

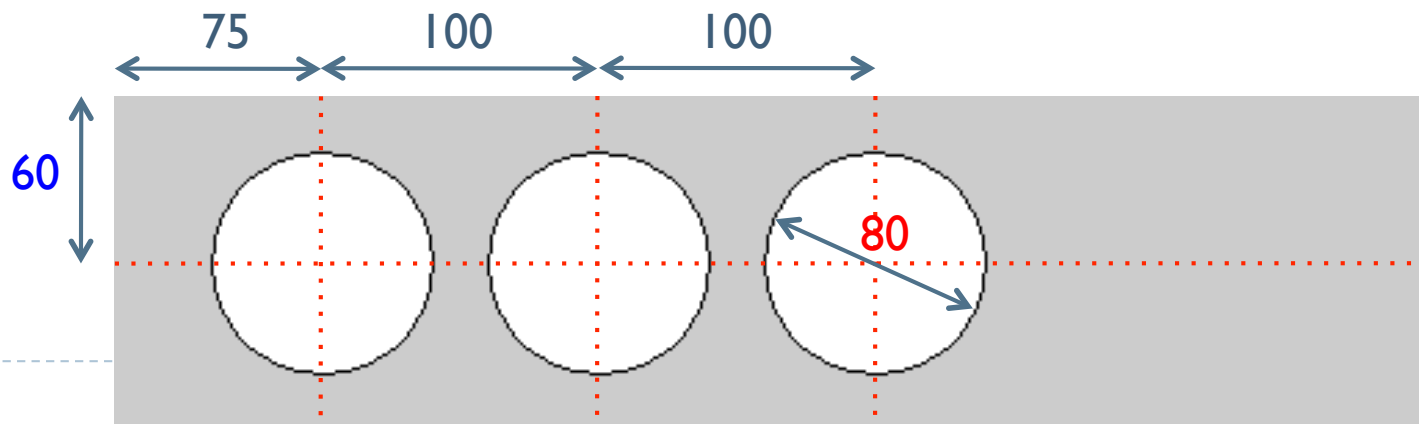
Before: 変数使わない

```
size(480, 120);  
ellipse(75, 60, 80, 80);  
ellipse(175, 60, 80, 80);  
ellipse(275, 60, 80, 80);
```

After: 変数使う

```
size(480, 120);  
int y = 60; // y座標  
int d = 80; // 直径  
ellipse(75, y, d, d);  
ellipse(175, y, d, d);  
ellipse(275, y, d, d);
```

※diameter 直径



# 値を変更したい場合

## ▶ 変数を使うと、最小限の修正で済む

Before: 変数使わない

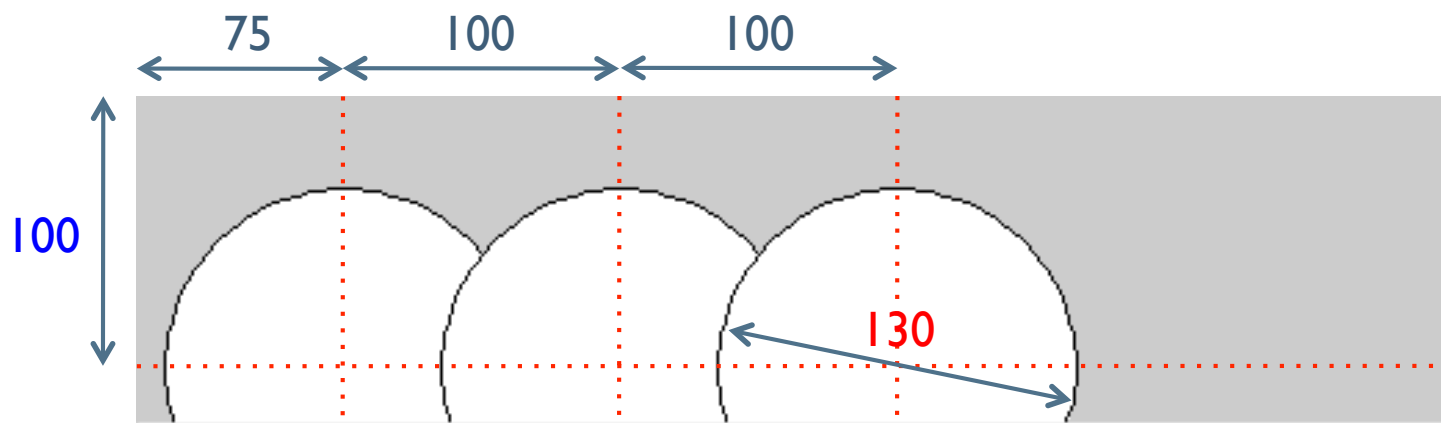
```
size(480, 120);  
ellipse(75, 100, 130, 130);  
ellipse(175, 100, 130, 130);  
ellipse(275, 100, 130, 130);
```

修正9ヶ所

After: 変数使う

```
size(480, 120);  
int y = 100; // y座標  
int d = 130; // 直径(diameter)  
ellipse(75, y, d, d);  
ellipse(175, y, d, d);  
ellipse(275, y, d, d);
```

修正2ヶ所



# 変数を使う: 1

---

## ▶ 事前の準備(必須)

### ▶ 変数の宣言

- ▶ 「〇〇という変数を使うよ」という意味
- ▶ 変数の型(種類)、名前を指定する

型	名前	
int	x;	// xという名前の整数の変数を使う

※integer 整数  
他の種類は後で

## ▶ 使用方法

### ▶ 値を「代入」して使う

x = 12;	// xに12を代入
---------	------------

- ▶ 「=」は「代入」を意味する
- ▶ 「等しい」ではない

## 変数を使う: 2

---

- ▶ 「初期化」: 変数の宣言と代入を同時に

宣言	代入
<code>int x</code>	<code>= 12;</code>

// 整数の変数xを宣言し、12を代入

- ▶ 注意

- ▶ 宣言は1回だけ

✗

```
int x;  
int x = 12;
```

※Duplicate local variable ○○...  
ローカル変数○○が重複してます

## 特殊な変数

---

- ▶ ウィンドウの大きさ
  - ▶ width 幅
  - ▶ height 高さ
  
- ▶ 他にもいくつか特殊な変数がある

# 計算

---

## ▶ 変数や数を使って計算できる

```
x = 4 + 4 * 5;
```

## ▶ 数学の場合と記号が異なるものもある

+	足す
-	引く
*	かける。「×」ではない
/	割る。「%」ではない
=	代入。「等しい」という意味ではない

※計算の記号などを  
プログラミング言語では  
「演算子」という

## ▶ 「代入」は「上書き」

```
x = 25; // xに25を代入  
x = x + 100; // xが125になる
```

# 計算順序

---

- ▶ かけ算や割り算の方が、足し算や引き算よりも先に

```
x = 4 + 4 * 5; // 4 + (4 * 5)に
```

- ▶ 計算順序を変えるには()を使う

```
x = (4 + 4) * 5;
```

- ▶ この順番をプログラミング言語では「演算子の優先順位」という
  - ▶ Processingの場合はp.202



# 特殊な計算

---

## ▶ 計算して代入

```
x += 10; // xに10を足して、それをxに代入  
y -= 15; // yから15を引いて、それをyに代入
```

## ▶ 読み方

① ②

**十二**

①足して、②代入

## ▶ 1つ増やす、1つ減らす

```
x++; // xを1つ増やす  
y--; // yを1つ減らす
```

# プログラムの基本3大要素

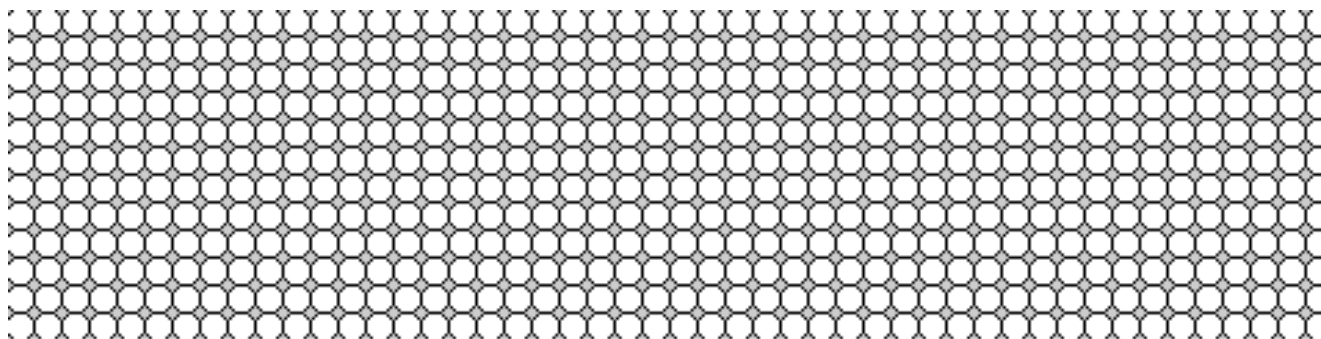
---

- ▶ 逐次実行
  - ▶ 順番に実行する
- ▶ 条件分岐
  - ▶ 状況に応じて、実行する内容を変える
- ▶ くり返し
  - ▶ 何回もくり返す

## 同じことのくり返し

---

- ▶ 丸をたくさん描く
  - ▶ 数が増えてくると、とても大変...



48×12回 ellipse()を書く？

- ▶ くり返しの機能を使うと、超簡単に

# くり返しの例

## ▶ 線を7本ひく

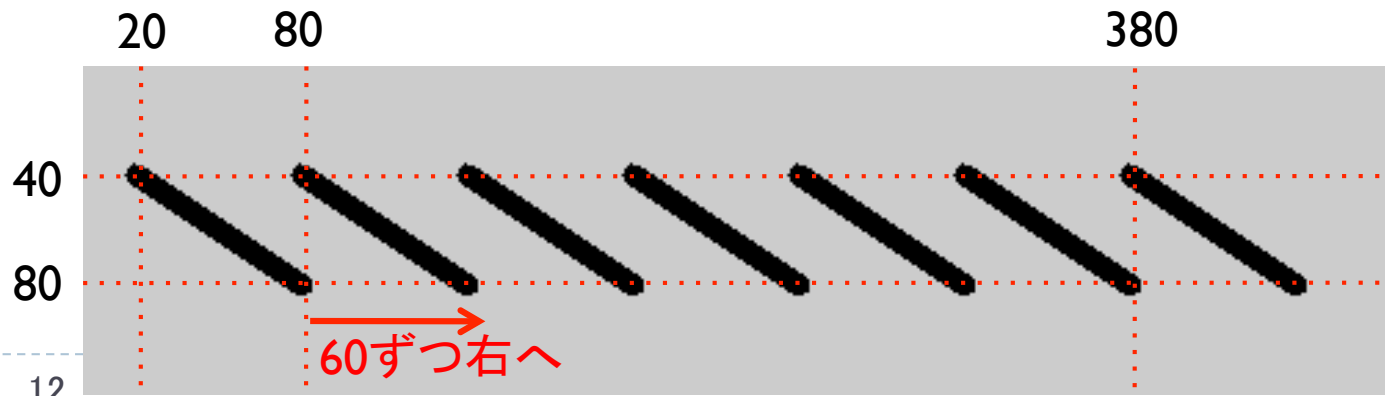
Before: くり返し機能を使わない

```
size(480, 120);  
strokeWeight(8);  
line(20, 40, 80, 80);  
line(80, 40, 140, 80);  
line(140, 40, 200, 80);  
line(200, 40, 260, 80);  
line(260, 40, 320, 80);  
line(320, 40, 380, 80);  
line(380, 40, 440, 80);
```

After: くり返し機能を使う

```
size(480, 120);  
strokeWeight(8);  
for (int i = 20; i <= 380; i += 60) {  
  line(i, 40, i + 60, 80);  
}
```

赤い部分: 20から、380まで、60ずつ増える  
青い部分: 赤い部分に60を足した値



# for文

---

- ▶ 初期設定(init)、テスト (test)、更新(update)を指定して繰り返す

```
for (初期設定; テスト; 更新) {  
    くり返し実行する内容  
}
```

- ▶ 「テスト」とは？
  - ▶ くり返しを「続けるか」どうかを、テストする
    - ▶ テスト結果がOKだったら、くり返しを続ける
    - ▶ テスト結果がダメだったら、くり返しをやめる

## for文の例

---

- ▶ **i**の値を1から10まで1つずつ増やす、くり返し

```
for (int i = 1; i <= 10; i++) {  
    くり返し実行する内容  
}
```

- ▶ **i**の値を20から380まで20ずつ増やす、くり返す

```
for (int i = 20; i <= 380; i += 20) {  
    くり返し実行する内容  
}
```

# for文のフローチャート

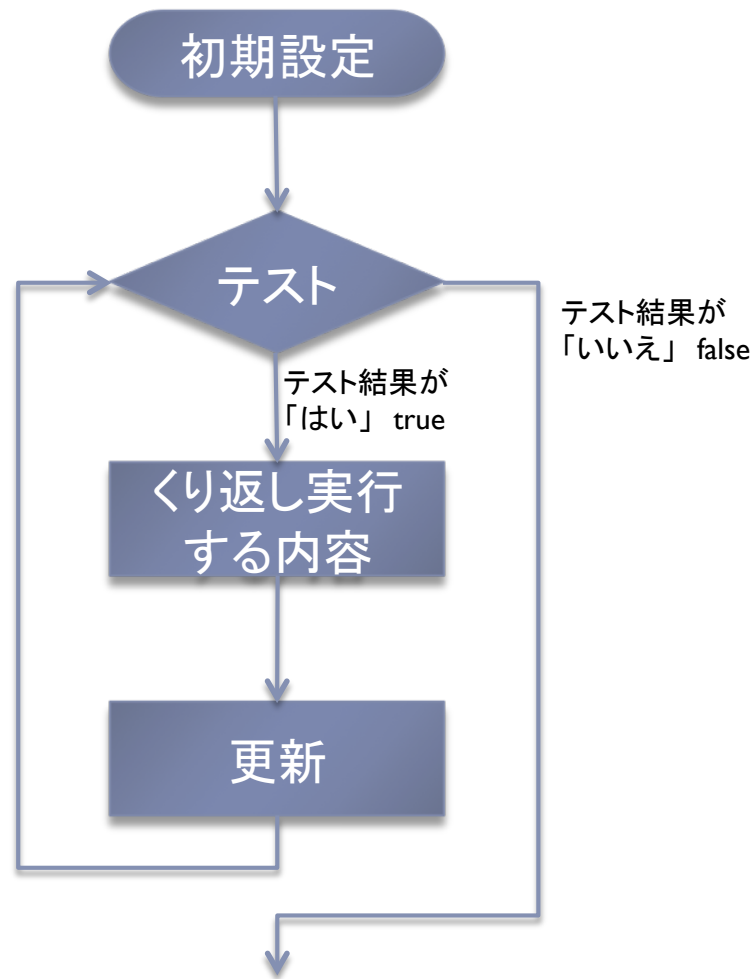
## ▶ フローチャート: プログラムの動きを図にしたもの

```
for (初期設定; テスト; 更新) {  
    くり返し実行する内容  
}
```

## ▶ テストの結果

例 テスト「 $x < 10$ 」

- ・  $x$ が3のとき  
 テスト結果は「はい」 true
- ・  $x$ が20のとき  
 テスト結果は「いいえ」 false



# 比較する：関係演算子

---

## ▶ 比較するのに使う演算子を関係演算子という

<	より小さい
>	より大きい
<=	以下
>=	以上
=	等しい
!=	等しくない

## ▶ 数学と異なるので注意

- ▶  $\leq$ 、 $\geq$ のような記号はない
  - ▶  $<=$ 、 $>=$ になっている



# くり返しを使うコツ

## ▶ 何がくり返されているのか

▶ 変化しないもの

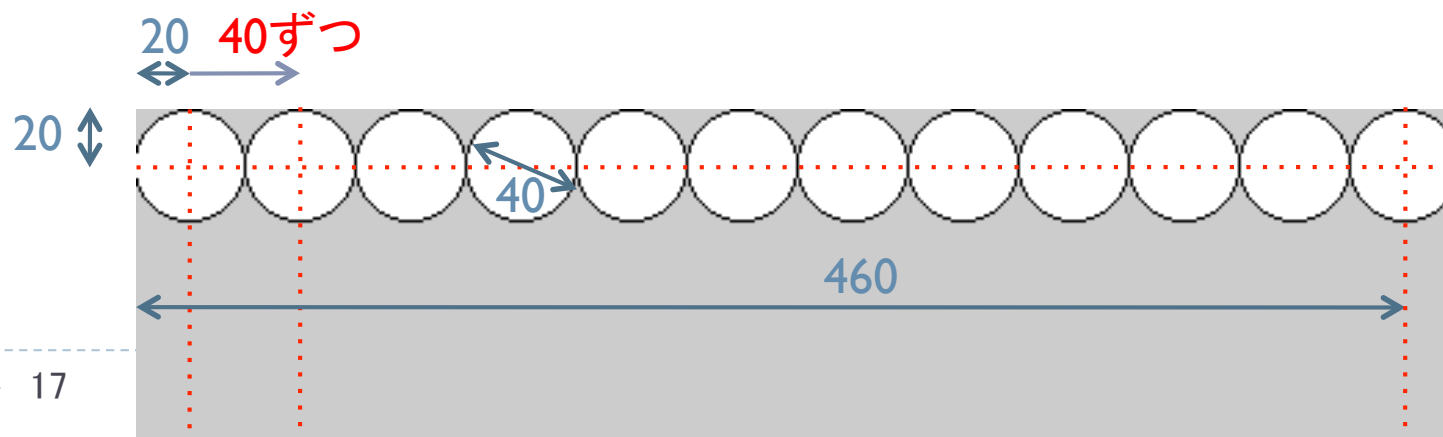
▶ 変化するもの

▶ 最初が何で、終わりが何で、どのくらいずつ変化するか

```
size(480, 120);  
ellipse(20, 20, 40, 40);  
ellipse(60, 20, 40, 40);  
...  
ellipse(460, 20, 40, 40);
```

```
size(480, 120);  
for (int x = 20; x <= 460; x += 40) {  
    ellipse(x, 20, 40, 40);  
}
```

xは、最初20で、最後460で、40ずつ増える



## 2重のくりかえし

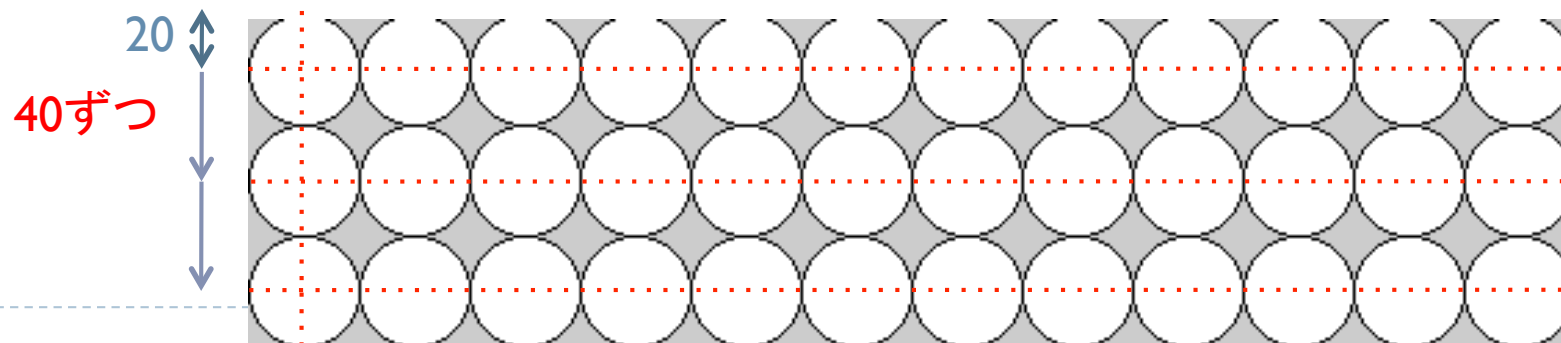
- ▶ くり返しは2重、3重に書ける

- ▶ x方向、y方向のように2つの方向にくり返す場合など

```
size(480, 120);  
for (int x = 20; x <= 460; x += 40) {  
  ellipse(x, 20, 40, 40);  
}  
...  
for (int x = 20; x <= 460; x += 40) {  
  ellipse(x, 100, 40, 40);  
}
```

```
size(480, 120);  
for (int y = 20; y <= 100; y += 40) {  
  for (int x = 20; x <= 460; x += 40) {  
    ellipse(x, y, 40, 40);  
  }  
}
```

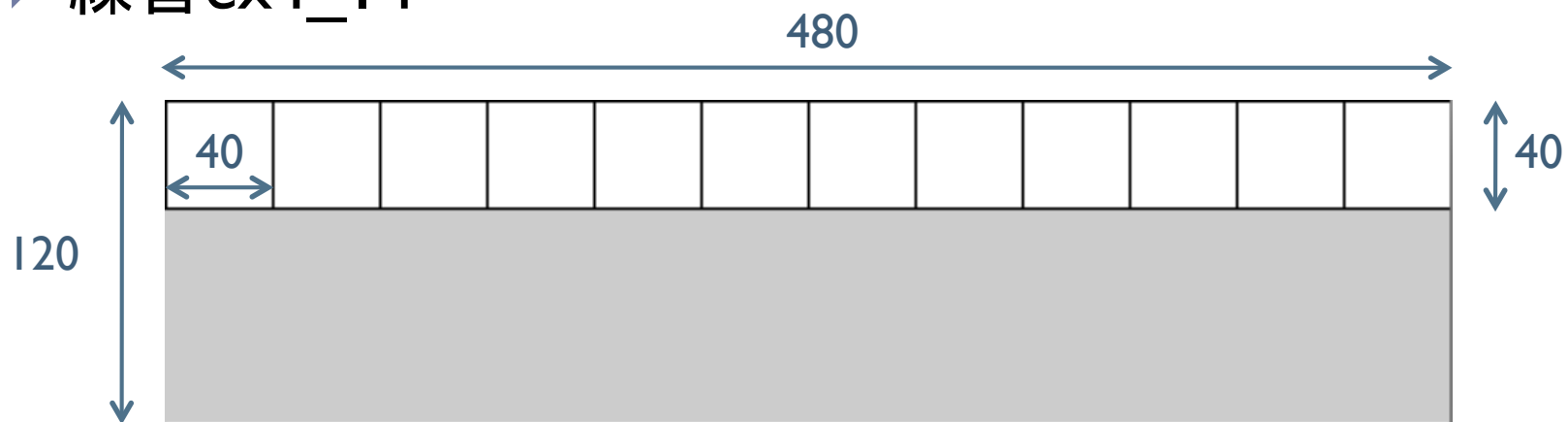
yは、最初20で、最後100で、40ずつ増える



# 練習

---

## ▶ 練習ex4\_14



## ▶ 練習ex4\_15

