

# pi-4. Javaプログラミングにおける条件分岐と実行フロー制御

トピックス：条件分岐, if, else, ステップ実行

URL: <https://www.kkaneko.jp/pro/pi/index.html>

(Java の基本, スライド資料とプログラム例)

金子邦彦



```
1 public class YourClassNameHere {
2     public static void main(String[] args) {
3         int age = 10;
4         if (age <= 12) {
5             System.out.println("500 yen");
6         } else {
7             System.out.println("1200 yen");
8         }
9     }
10 }
```

Print output (drag low  
500 yen



Frames

main:9	
age	10
Return value	void

条件分岐, age <= 12 のときは 500 yen,  
age > 12 のときは 1200 yen

Java 8  
(known limitations)

```
1 public class YourClassNameHere {
2     public static void main(String[] args) {
3         int age = 20;
4         if (age <= 12) {
5             System.out.println("500 yen");
6         } else {
7             System.out.println("1200 yen");
8         }
9     }
10 }
```

[Edit this code](#)

→ line that just executed  
→ next line to execute

<< First < Prev Next > Last >>

Step 4 of 5

Print output (drag lower right corner)

Frames      Objects

main:7	
age	20

ステップ実行により, ジャンプの様子を観察

# 全体まとめ



- **条件分岐**では、変数や式の値によって結果が変わるなどの判断を行う  
条件式は「**age <= 12**」のようになる
- プログラムは、通常は、上から順に1つずつ実行される（**逐次実行**）
- **条件分岐**や**繰り返し（ループ）**では、逐次実行とは違う実行の流れになる
- **ステップ実行**により、変数等の変化、プログラム実行の流れを観察できる

番号	項目
	条件分岐
4-1	演習
4-2	ステップ実行
4-3	条件分岐

各自、資料を読み返したり、課題に取り組んだりも行う

この授業では、**Java** を用いて基礎を学び、マスターする

# オブジェクトとメソッド



**hero.moveDown()**

**hero**            **オブジェクト**  
**moveDown()**   **メソッド**  
間を「.」で区切っている

- **メソッド: オブジェクト**に属する操作や処理.
- **メソッド**呼び出しでは, **引数**を指定することがある. **引数** (ひきすう) は, **メソッド**に渡す値のこと

**hero.attack("fence", 36, 26)**

# Java プログラムの書き方



## プログラムの例

```
x = 100
a = x + 200
enemy1 = hero.findNearestEnemy()
hero.attack(enemy1)
```

- **代入** : **オブジェクト名** + 「**=**」  
+ 式または値またはメソッド呼び出し
- **メソッドアクセス** : **オブジェクト名** + 「**.**」  
+ **メソッド名** + 「**()**」 (引数を付けることも)

その他, 属性アクセス, 関数呼び出し, 制御, 「\*」, 「+」などの演算子, コマンド, 定義など

# Java Tutor の起動



① **ウェブブラウザ**を起動する

② **Java Tutor** を使いたいのので, 次の URL を開く  
**<http://www.pythontutor.com/>**

③ 「**Java**」 をクリック ⇒ **編集画面**が開く

## Learn Python, JavaScript, C, C++, and Java

This tool helps you learn Python, JavaScript, C, C++, and Java programming by [visualizing code execution](#). You can use it to debug your homework assignments and as a supplement to online coding tutorials.

Start coding now in [Python](#), [JavaScript](#), [C](#), [C++](#), and [Java](#)

**Over 15 million people in more than 180 countries** have used Python Tutor to visualize over 200 million pieces of code. It is the most widely-used program visualization tool for computing education.

You can also embed these visualizations into any webpage. Here's an example showing recursion in Python:

# Java Tutor でのプログラム実行手順



```
1 public class YourClassNameHere {
2     public static void main(String[] args) {
3         int x = 100;
4         int y = 200;
5         System.out.printf("%d\n", x + y);
6     }
7 }
```

Visualize Execution



Java 8  
(known limitations)

```
1 public class YourClassNameHere {
2     public static void main(String[] args) {
3         int x = 100;
4         int y = 200;
5         System.out.printf("%d\n", x + y);
6     }
7 }
```

Edit this code

→ line that just executed  
→ next line to execute

<< First < Prev Next > Last >>

Step 1 of 5



(1) 「Visualize Execution」をクリックして実行画面に切り替える

(2) 「Last」をクリック。

Print output (drag lower right corner to resize)

300

Frames	Objects
main:6	
x	100
y	200
Return value	void



Java 8  
(known limitations)

```
1 public class YourClassNameHere {
2     public static void main(String[] args) {
3         int x = 100;
4         int y = 200;
5         System.out.printf("%d\n", x + y);
6     }
7 }
```

Edit this code

→ line that just executed  
→ next line to execute

<< First < Prev Next > Last >>

Done running (5 steps)

(3) 実行結果を確認する。

(4) 「Edit this code」をクリックして編集画面に戻る

# Java Tutor 使用上の注意点①



- 実行画面で、次のような赤の表示が出ることもある →  
無視してよい

過去の文法ミスに関する確認表示  
邪魔なときは「Close」

## Python Tutor: Visualize code in [Python](#), [JavaScript](#), [C](#), [C++](#), and [Java](#)

Java 8  
([known limitations](#))

```
1 public class YourClassNameHere {
2     public static void main(String[] args) {
3         int x = 100;
4     }
5 }
```

[Edit this code](#)

→ line that just executed  
→ next line to execute

<< First < Prev Next > Last >>

Step 1 of 3

[Customize visualization](#)

Frames      Objects

main:3

You just fixed the following error:

```
1 public class YourClassNameHere {
2     public static void main(String[] args) {
3         int x = 100
4     }
5 }
```

Error: ';' expected

Please help us improve this tool with your feedback.  
What misunderstanding do you think caused this error?

Submit    **Close**    [hide all of these pop-ups](#)

# Java Tutor 使用上の注意点②



「please wait ... executing」のとき、10秒ほど待つ。

Python Tutor: Visualize code in [Python](#), [Ja](#)

Please wait ... your code is running (up to 10 seconds)

Write code in [Java 8](#)

```
1 public class YourClassNameHere {
2     public static void main(String[] args) {
3         int x = 100;
4     }
5 }
```

Please wait ... executing (takes up to 10 seconds)

- 混雑しているときは、「Server Busy・・・」  
というメッセージが出ることがある。  
混雑している。少し（数秒から数十秒）待つと自  
動で表示が変わる（変わらない場合には、操作を  
もう一度行ってみる）

# 4-1. 条件分岐

# 条件分岐



条件分岐では、変数や式の値によって結果が変わるなどの判断を行う

**age** の値が **12**以下 → **500**  
**13**以上 → **1800**

条件式は「**age <= 12**」のようになる

# 条件分岐のプログラム例



```
public class YourClassNameHere {  
    public static void main(String[] args) {  
        int age = 10;  
        if (age <= 12) {  
            System.out.println("500 yen");  
        } else {  
            System.out.println("1200 yen");  
        }  
    }  
}
```

字下げを行うことで、  
プログラムを読みやすくしている

## 4-2. 演習

# 演習

資料 : 16 ~ 19

## 【トピックス】

- 条件分岐
- if
- else

① Java Tutor のエディタで次のプログラムを入れる



## 10 歳のときは 500円

```
1 public class YourClassNameHere {
2     public static void main(String[] args) {
3         int age = 10;
4         if (age <= 12) {
5             System.out.println("500 yen");
6         } else {
7             System.out.println("1200 yen");
8         }
9     }
10 }
```

## ② 実行し，結果を確認する

# 10 歳のときは 500円

Print output (drag low

500 yen

結果の  
「500 yen」が表示  
されるので確認

### Frames

main:9	
age	10
Return value	void

「Visual Execution」をクリック，そして「Last」をクリック，結果を確認。  
「Edit this code」をクリックすると，エディタの画面に戻る

③ Java Tutor のエディタで次のプログラムを入れる



## 20 歳のときは 1200円

```
1 public class YourClassNameHere {
2     public static void main(String[] args) {
3         int age = 20;
4         if (age <= 12) {
5             System.out.println("500 yen");
6         } else {
7             System.out.println("1200 yen");
8         }
9     }
10 }
```

## ④ 実行し，結果を確認する

### 20 歳のときは 1200円

Print output (drag low

1200 yen

結果の  
「1200 yen」が表示  
されるので確認

Frames

main:9	
age	20
Return value	void

★ プログラムは消さずに  
残しておくこと

「**Visual Execution**」をクリック。そして「**Last**」をクリック。結果を確認。  
「**Edit this code**」をクリックすると，エディタの画面に戻る

## 4-3. 条件分岐のステップ実行

# プログラム実行

- **命令（コマンド）** は、順番に並んでいる
- 通常は、上から順に1つずつ実行される（逐次実行）
- **条件分岐**や**繰り返し（ループ）** では、逐次実行とは違う実行の流れになる

※ プログラミング言語の種類によっては、順番に並んでいないという場合もある： SQL, Scheme, Prolog など

# ステップ実行



- **ステップ実行により, プログラム実行の流れをビジュアルに観察**

# Java Tutor でのステップ実行



## ステップ実行により、プログラム実行の流れを確認できる

Java 8  
([known limitations](#))

```
1 public class YourClassNameHere {
2     public static void main(String[] args) {
3         int age = 20;
4         if (age <= 12) {
5             System.out.println("500 yen");
6         } else {
7             System.out.println("1200 yen");
8         }
9     }
10 }
```

[Edit this code](#)

- line that just executed
- next line to execute

Print output (drag lower right corner)

Frames	Objects
main:7	
age	20

Step 4 of 5

<< First   < Prev   Next >   Last >>

## 演習

資料：25 ～ 31

### 【トピックス】

- Java Tutor でのステップ実行の操作
- 変数の値の変化
- 実行の流れの変化（ジャンプ）

# ① まずは、「Visualize Execution」をクリックして、実行開始



Write code in Java 8 ▼

```
1 public class Main {
2     public static void main(String[] args) {
3         int age = 20;
4         if (age <= 12) {
5             System.out.printf("500 yen");
6         } else {
7             System.out.printf("1200 yen");
8         }
9     }
10 }
11 |
```

Help improve this tool by completing a [short user survey](#).

Visualize Execution

- ② 「**Step 1 of 5**」 と表示されているので、  
全部で、ステップ数は 5 あることが分かる  
(ステップ数と、プログラムの行数は**違うもの**)

Java

```
1 public class Main {  
2     public static void main(String[] args) {  
→ 3         int age = 20;  
4         if (age <= 12) {  
5             System.out.printf("500 yen");  
6         } else {  
7             System.out.printf("1200 yen");  
8         }  
9     }  
10 }
```

[Edit this code](#)

What has just executed

Line to execute

Use the Back and Forward buttons to jump there.

<< First

< Back

Step 1 of 5

Forward >

Last >>

③ **ステップ実行**したいので、「Next」をクリックしながら、矢印の動きを確認.

※ 「Next」 ボタンを何度か押し、それ以上進めなくなったら終了

Java

```
1 public class Main {  
2     public static void main(String[] args) {  
3         int age = 20;  
4         if (age <= 12) {  
5             System.out.printf("500 yen");  
6         } else {  
7             System.out.printf("1200 yen");  
8         }  
9     }  
10 }
```



見どころ

**4行目から 7行目へ  
ジャンプ**するところ

[Edit this code](#)

rat has just executed  
ine to execute

: of code to set a breakpoint; use the Back and Forward buttons to jump there.

<< First

< Back

Step 1 of

Forward >

Last >>

# ④ 最後の行まで達したら、「**Edit this code**」をクリックして、エディタの画面に戻る

```
Java
-----
1 public class Main {
2     public static void main(String[] args) {
3         int age = 20;
4         if (age <= 12) {
5             System.out.printf("500 yen");
6         } else {
7             System.out.printf("1200 yen");
8         }
9     }
10 }
```

[Edit this code](#)

that has just executed  
line to execute

ie of code to set a breakpoint; use the Back and Forward buttons to jump there.

Print output (drag lowe

1200 yen

Frames

main:9  
age 20



<< First < Back Step 5 of 5 Forward > Last >>

⑤ 次のように「age = 10」と書き換えて、いままでと同じことを行う

```
1 public class Main {
2     public static void main(String[] args) {
3         int age = 10;
4         if (age <= 12) {
5             System.out.printf("500 yen");
6         } else {
7             System.out.printf("1200 yen");
8         }
9     }
10 }
```



見どころ

5行目から9行目へ  
ジャンプするところ

- 次のプログラムを作成

- **条件分岐**

**weight** の値が **100未満** → **0 yen**

**100以上** → **1000 yen**

- 「**weight = 50**」に設定してテスト実行

正解の例は次のページ

```
1 public class Main {
2     public static void main(String[] args) {
3         int weight = 50;
4         if (weight < 100) {
5             System.out.printf("0 yen");
6         } else {
7             System.out.printf("1000 yen");
8         }
9     }
10 }
```

## 関連ページ

- **Java プログラミング入門**

GDB online を使用

<https://www.kkaneko.jp/pro/ji/index.html>

- **Java の基本**

Java Tutor, GDB online を使用

<https://www.kkaneko.jp/pro/pi/index.html>

- **Java プログラム例**

<https://www.kkaneko.jp/pro/java/index.html>

# 資料中のソースコード 4-1



```
public class YourClassNameHere {  
    public static void main(String[] args) {  
        int age = 10;  
        if (age <= 12) {  
            System.out.println("500 yen");  
        } else {  
            System.out.println("1200 yen");  
        }  
    }  
}
```