

pi-1. プログラミング入門： Java言語の基礎と実行環境の 理解

トピックス：プログラミング, Java Tutor での
Java プログラム実行, GDB online での Java プロ
グラム実行, 計算誤差, さまざまなプログラミン
グ言語

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



```
public class Main {
    public static void main(String[] args) throws Exception {
        int x = 100;
        if (x > 20) {
            System.out.printf("big%n");
        } else {
            System.out.printf("small%n");
        }
        int s = 0;
        for(int i = 1; i <= 5; i++) {
            s = s + i;
        }
        System.out.printf("%d%n", s);
    }
}
```



```
C:¥Users¥user>javac Main.java
C:¥Users¥user>java Main
big
15
```

Java プログラム実行のためのコマンドと実行結果

Java プログラムのソースコード

```
x = 100
if (x > 20):
    print("big")
else:
    print("small")
s = 0
for i in [1, 2, 3, 4, 5]:
    s = s + i
print(s)
```

Python

```
public class Main {
    public static void main(String[] args) throws Exception
    {
        int x = 100;
        if (x > 20) {
            System.out.printf("big%n");
        } else {
            System.out.printf("small%n");
        }
        int s = 0;
        for(int i = 1; i <= 5; i++) {
            s = s + i;
        }
        System.out.printf("%d%n", s);
    }
}
```

Java

```
#include <stdio.h>
int main(void){
    int x, s, i;
    x = 100;
    if (x > 20) {
        printf("big%n");
    } else {
        printf("small%n");
    }
    s = 0;
    for(i = 1; i <= 5; i++) {
        s = s + i;
    }
    printf("%d%n", s);
    return;
}
```

C

さまざまな
プログラミング言語



Python Tutor: Visualize code in Python, JavaScript, C, C++, a

Java 8 (known limitations)

```

1 public class Main {
2     public static void main(String[] args) throws Exception {
3         int x = 100;
4         if (x > 20) {
5             System.out.printf("big\n");
6         } else {
7             System.out.printf("small\n");
8         }
9         int s = 0;
10        for(int i = 1; i <= 5; i++) {
11            s = s + i;
12        }
13        System.out.printf("%d\n", s);
14    }
15 }

```

Print output (drag lower right corner to resize)

```
big
15
```

Frames

```
main:14
x 100
s 15
Return value void
```

Objects

Done running (24 steps)

Main.java

```

1 public class Main {
2     public static void main(String[] args) throws Exception {
3         int x = 100;
4         if (x > 20) {
5             System.out.printf("big\n");
6         } else {
7             System.out.printf("small\n");
8         }
9         int s = 0;
10        for(int i = 1; i <= 5; i++) {
11            s = s + i;
12        }
13        System.out.printf("%d\n", s);
14    }
15 }

```

input

```
big
15
```

...Program finished with exit code 0
Press ENTER to exit console.

オンラインでの Java プログラム
実行 (Java Tutor を使用)

オンラインでの Java プログラム
実行 (GDB online を使用)

コンピュータは便利なものであるが、コンピュータを使うから
といって、**計算が完璧に正確**というわけではない

```

1 public class YourClassNameHere {
2     public static void main(String[] args) {
3         System.out.printf("%f\n", 1.0/3.0);
4     }
5 }

```



Print output (drag |)

```
0.333333
```

プログラム

実行結果

番号	項目
1-1	プログラミング
1-2	Java プログラムの実行方法
1-3	オンライン開発環境
1-4	Java Tutor での Java プログラム実行
1-5	GDB online での Java プログラム実行
1-6	計算誤差
1-7	さまざまなプログラミング言語
1-8	この授業の全体計画

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

1-1. プログラミング

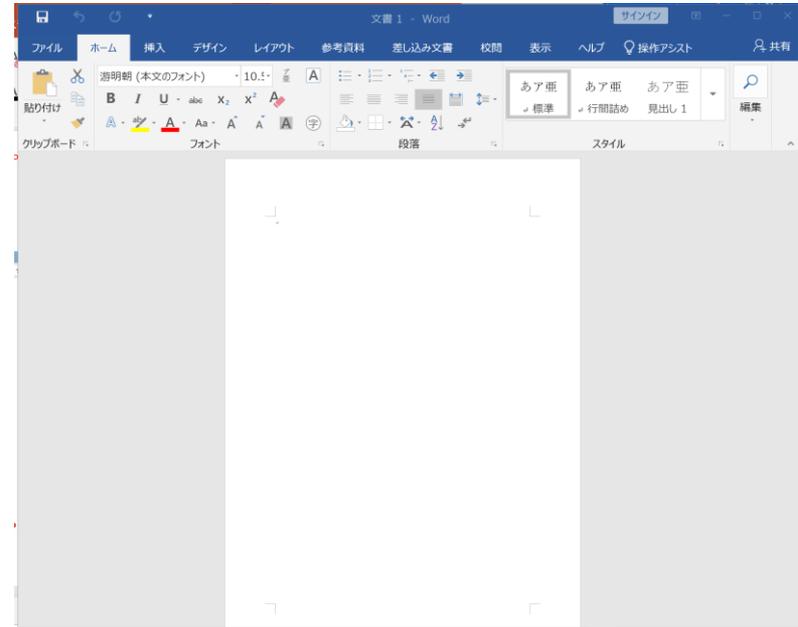
プログラム

- **コンピュータ**は, **プログラム**で動く
- プログラムを設計, 制作することはクリエイティブである

① さまざまなアプリ



Web ブラウザ



ワープロ
(マイクロソフト・ワード)

アプリでは、**プログラム**が動いている

② コンピュータを細かくコントロール



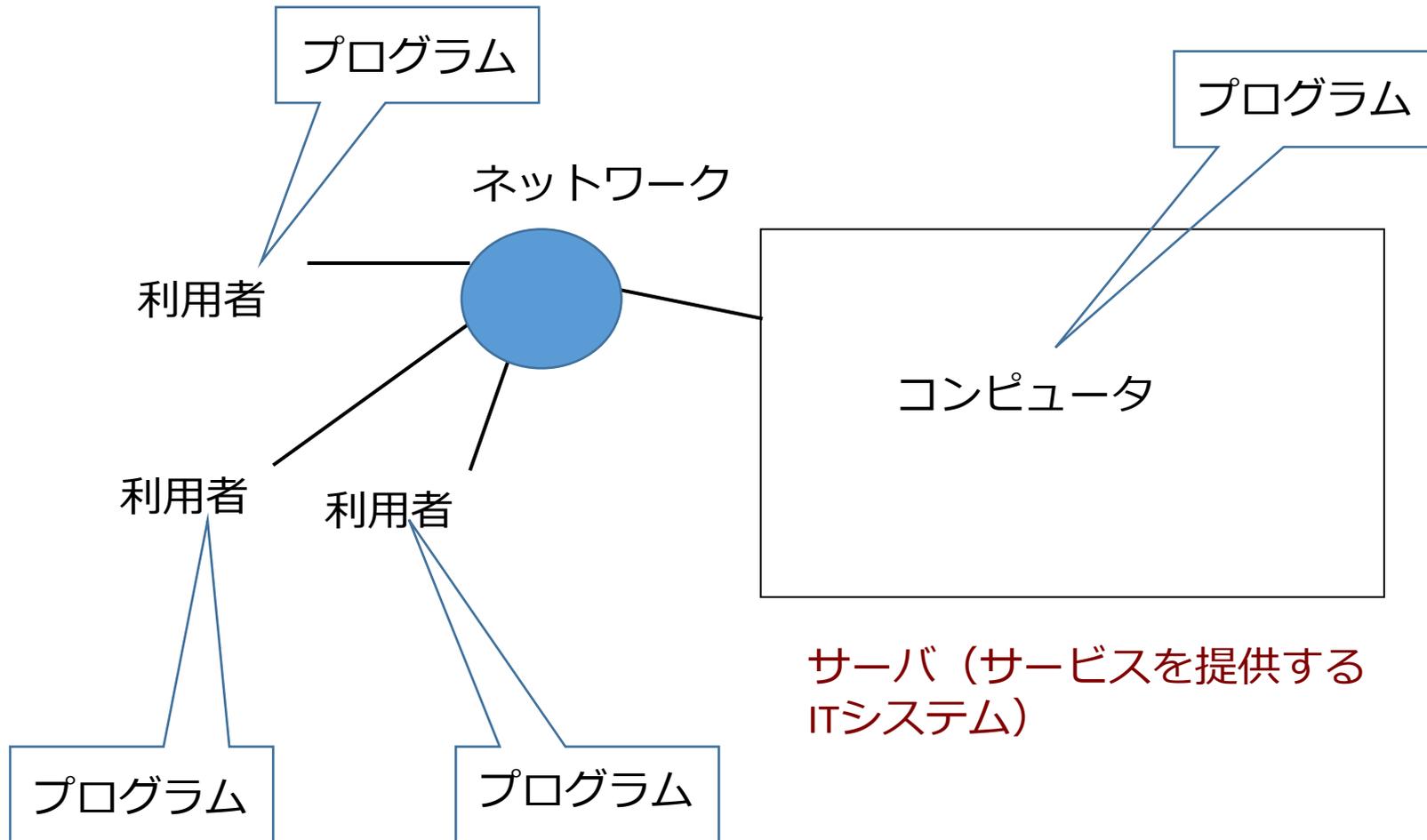
```
In [7]: from keras.models import Sequential
...: model = Sequential()
...: from keras.layers import Dense, Activation
...:
...: model.add(Dense(units=64, input_dim=len(x_train[0])))
...: model.add(Activation('relu'))
...: model.add(Dense(units=max(set(y_train)) - min(set(y_train)) + 1))
...: model.add(Activation('softmax'))
...: model.compile(loss='sparse_categorical_crossentropy',
...:               optimizer='sgd',
...:               metrics=['accuracy'])
...: model.fit(x_train, y_train, epochs=200)
...: score=model.evaluate(x_test, y_test, batch_size=1)
...: print(score)
...: model.predict(x_test)
...: model.summary()

Epoch 1/200
3/3 [=====] - 0s 5ms/step - loss: 1.0583 - accuracy:
0.3200
Epoch 2/200
3/3 [=====] - 0s 0s/step - loss: 1.0530 - accuracy:
0.3200
Epoch 3/200
3/3 [=====] - 0s 0s/step - loss: 1.0485 - accuracy:
0.3200
```

人工知能のプログラム
(Python 言語)

ニューラルネットワークを
作成している

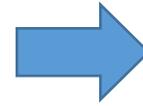
③ コンピュータどうしがつながるときも プログラムが必要



プログラミング (programming)

- コンピュータは、プログラムで動く
- プログラミングは、プログラムを設計、製作すること
- 何らかの作業を、コンピュータで実行させるために行う

```
public class YourClassNameHere {  
    public static void main(String[] args) {  
        int x = 100;  
        int y = 200;  
        System.out.printf("%d", x * y);  
    }  
}
```



20000

プログラムの
ソースコード
(Java 言語)

プログラムの
実行結果

ソースコード (source code)

- **プログラム**を, 何らかの**プログラミング言語**で書いたもの
- 「**ソフトウェアの設計図**」ということも.

人間も読み書き, 編集できる

```
public class YourClassNameHere {  
    public static void main(String[] args) {  
        int x = 100;  
        int y = 200;  
        System.out.println(x + y);  
    }  
}
```

100 × 200 を計算する Java 言語プログラム

プログラムが役に立つ理由



- ① プログラム次第で，様々な処理が可能.
- ② プログラムは，コンピュータでの様々な処理を自動化する
- ③ プログラムのソースコードは，作業記録としても使うことができる．いつでも再現できる．
- ④ プログラム中の値などを変えて再実行も簡単

プログラミングで気を付けること



- ① コンピュータにも、**できないことがある**
- ② コンピュータを使うからといって、**計算が完璧に正確**というわけでは**ない**
- ③ **人間**がプログラムを作るとき、書き間違い、勘違い、思い込みなどによる**ミスがありえる**。
- ④ 「プログラムが期待通りに動いているか」の**テストが重要**
- ⑤ **ミスを減らす**ためにも、「やりたいこと」を**1回書いて済ませる**ことが大切。次のようなさまざまな手段がある
 - 抽象化
 - 標準ライブラリ
 - クラス階層
- ⑥ **問題**をコンピュータで解くとき、解くべき**問題を深く理解**した上で、必要に応じて、**算法（アルゴリズム）**を活用する

1-2. Java プログラムの実行方法

プラットフォームとは



- もともとは、**大地**、**乗り降り場**等の意味
- IT では、ソフトウェア等を**動作させるのに必要な****機器**や**ソフトウェア**のこと

Windows 10 + パソコン

Max OS X + パソコン

Linux + サーバコンピュータ

Android + スマホ

Java 言語の良さ



- さまざまなプラットフォームで、同じプログラムが動く (プラットフォーム非依存)

私の見解

- Java の登場前は、「違うプラットフォームで動かすときは、プログラムの書き替えが必要」なのが常識
- Java の登場により、これが変化. 人気の理由
- インターネットの普及により、Windows, Linux 等がミックスして動く IT システムが当たり前. Java は便利に利用できる.

Java 言語の特徴



1. さまざまな**プラットフォーム**で、**同じプログラムが動く**（プラットフォーム非依存）
2. **オブジェクト指向のプログラミング言語**である
3. 標準ライブラリ（標準機能として備わっているライブラリ）が充実している
4. C++言語と書き方が類似
Java のことを「C++ の改良」という人も

Java のプログラムを動かすには

Java のソースコード



```
public class Main {  
    public static void main(String[] args) throws Exception {  
        int x = 100;  
        if (x > 20) {  
            System.out.printf("big%n");  
        } else {  
            System.out.printf("small%n");  
        }  
        int s = 0;  
        for(int i = 1; i <= 5; i++) {  
            s = s + i;  
        }  
        System.out.printf("%d%n", s);  
    }  
}
```

ファイル名: Main.java

コンパイル
(ビルド)

バイトコード

自動結合

Java 仮想マシン

標準ライブラリ

全部がそろって、
1つのアプリケーション

これで、プラットフォーム
非依存を達成

コンパイル (ビルド) は、
ソースコードをバイトコードに
変換する操作

Java のプログラムのコンパイル（ビルド）と実行



Java のソースコード

```
public class Main {
    public static void main(String[] args) throws Exception {
        int x = 100;
        if (x > 20) {
            System.out.printf("big%n");
        } else {
            System.out.printf("small%n");
        }
        int s = 0;
        for(int i = 1; i <= 5; i++) {
            s = s + i;
        }
        System.out.printf("%d%n", s);
    }
}
```

ファイル名: Main.java

Java のルール

Java のアプリケーション
を起動すると, **main メソッド**が実行される

```
kaneko@www:~$ javac Main.java
kaneko@www:~$ java Main
big
15
kaneko@www:~$
```

javac は、**コンパイル（ビルド）** を行うコマンド

java は**アプリケーションの起動**を行うコマンド

1-3. オンライン開発環境

プログラム開発環境



プログラム開発環境は、**プログラミング**におけるさまざまなことを支援する機能をもった**プログラム**

- プログラムの作成, 編集 (**エディタ**)
- プログラム中の誤り (**バグ**) の発見やテストの支援 (**デバッガ**)
- プログラムの実行
- マニュアルの表示
- プログラムが扱うファイルのブラウズ
- プログラムの配布 (**パッケージ機能**など) , 共有, 共同編集
- バックアップ, バージョン管理

これらが簡単に行えるようになる

オンラインのプログラム開発環境



- **プログラム開発環境**の操作は，ウェブブラウザでできる
- 自分のパソコンに，特別なソフトをインストールする必要がない
- 機能制限がある場合が多い
- 利用登録の有無と内容，利用条件，料金については，利用者で確認のこと

プログラム作成ができるウェブサービス (オンラインの開発環境) の例 ①



Google Colaboratory

Python の開発環境

多数のパッケージがインストール済み
ノートブックにより、記録が簡単に残せる。
ビジュアルな表示も簡単に可能
プログラムの共有も簡単

<https://colab.research.google.com/>

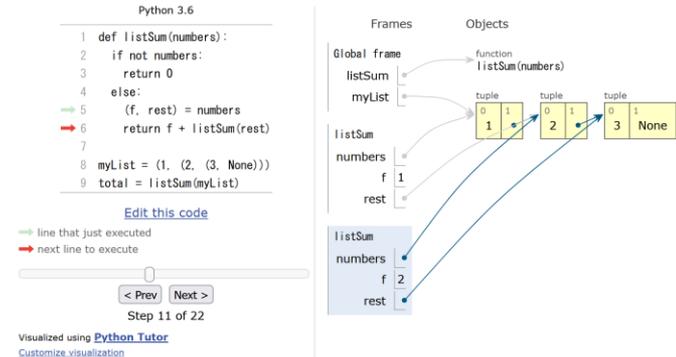
Learn Python, JavaScript, C, C++, and Java

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

Python, JavaScript, C, C++, Java
ステップ実行、オブジェクト
の表示がビジュアルに

<https://pythontutor.com/>

プログラム作成ができるウェブサービス (オンラインの開発環境) の例 ②



```
main.py
1 '''
2
3 Welcome to GDB Online.
4 GDB online is an online compiler and debugger tool for C, C
5 C#, OCaml, VB, Swift, Pascal, Fortran, Haskell, Objective-C
6 Code, Compile, Run and Debug online from anywhere in world.
7
8 '''
9 print ('Hello World')
```

```
--Return--
> /home/main.py(9)<module>()->None
-> print ('Hello World')
(Pdb) until
Hello World
--Return--
> <string>(1)<module>()->None
(Pdb) □
```

GDB online

C, C++, Java, Python, PHP, C#, OCaml, VB, HTML, Ruby, Perl, Pascal, R, Fortran, Haskell, アセンブリ, Objective C, SQLite, Javascript, Prolog, Swift, Rust, Go, Bash
デバッガの機能あり

<https://www.onlinegdb.com/>

```
main.py
1 x = [5, 4, 1, 3, 2]
2 for i in x:
3     print(i * 120)
4 |
```

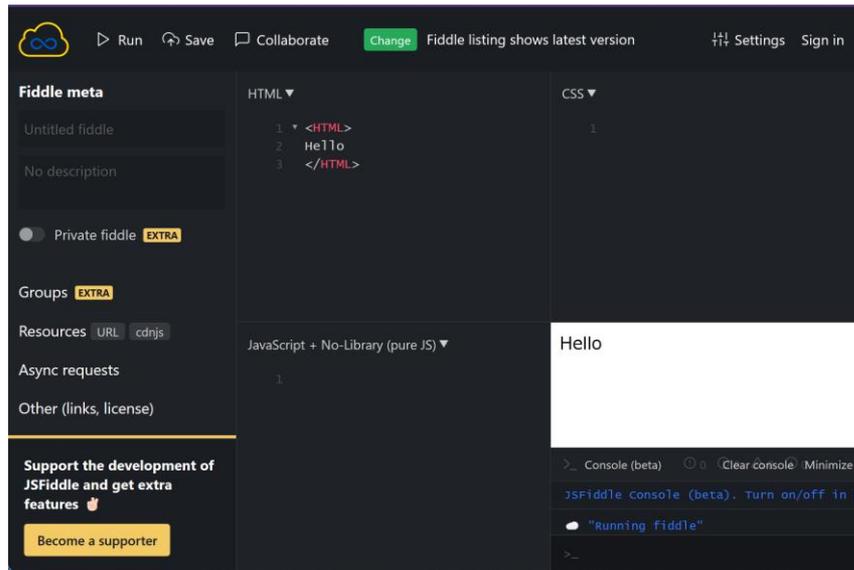
```
$python3 main.py
600
480
120
360
240
```

Coding Ground

Python, C, Java, JavaScript, R, Octave/MATLAB, SQL, bash, アセンブリ, MySQL, SQLite, その他多数
ファイル作成, ファイル読み書き, 複数プログラムファイルの組み合わせ可能

<https://www.tutorialspoint.com/codingground.htm>

プログラム作成ができるウェブサービス (オンラインの開発環境) の例 ③



JSFiddle

HTML, CSS, JavaScript
見た目をオンラインで確認

<https://jsfiddle.net/>



Paiza.IO

Python, C, Java, JavaScript, R, MySQL
など多数

表示は日本語.

一定の条件下でファイル操作も可能

<https://paiza.io/>

無料のオンラインサービス



- 一定の条件下で無料で使える.
- 活用によって, ICTはより便利になる.

【マナー】

- 作者が定める**利用条件**を確認
- **著作権**を尊重
- 安全意識：秘密にしたいデータをアップロードしない
- 広告等が表示される場合がある

1-4. Java Tutor での Java プ ログラム実行

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)

[customize visualization](#) (NEW!)

Print output (drag lower right corner to resize)

300

Frames

Objects

main:6

x 100

y 200

Return
value void

unsupported features

Java などのプログラミング言語の体験, 演習ができるオンラインサービス

Java Tutor

<http://www.pythontutor.com/>

オンラインなので、「秘密にしたいプログラム」を扱うには十分な注意が必要

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 の編集画面



Java debugger and visualizer - Java Tutor - Learn Java programming by visualizing code (also debug [Python](#), [JavaScript](#), [C](#), and [C++](#) code)

Write code in **Java 8** 「Java 8」になっている

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

最初から main メソッドの
ひな形が入っている

エディタ
(プログラムを書き換えることができる)

Visualize Execution

実行のためのボタン

inline primitives, don't nest objects [default] v draw pointers as arrows [default] v

[Show code examples](#)

Generate permanent link

Java Tutor でのプログラム実行手順



```
Write code in Java 8
```

```
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**が出ることがある →
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）

1-5. GDB online での Java プ ログラム実行



```
Main.java
1 public class Main
2 {
3     public static void main(String[] args) {
4         int x = 100;
5         int y = 200;
6         System.out.printf("%d\n", x + y);
7     }
8 }
```

input

```
300
...Program finished with exit code 0
Press ENTER to exit console.
```

Java などのプログラミング言語の体験, 演習ができるオンラインサービス

GDB online

<http://www.pythontutor.com/>

オンラインなので、「秘密にしたいプログラム」を扱うには十分な注意が必要

GDB online で Java を動かす手順



① ウェブブラウザを起動する

② 次の URL を開く

<https://www.onlinegdb.com>

A screenshot of a search bar with a magnifying glass icon on the left and the text "https://www.onlinegdb.com" entered inside. The search bar is set against a light gray background.

🔍 <https://www.onlinegdb.com>

③ 「Language」 のところで, 「Java」 を選ぶ

SPONSOR Slack — Bring your team together with Slack, the collaboration hub for work.

Run Debug Stop Share Save { } Beautify Language -- select --

```
1 /*****  
2  
3 Welcome to GDB Online.  
4 GDB online is an online compiler and debugger tool for C, C++, Python  
5 C#, VB, Perl, Swift, Prolog, Javascript, Pascal, HTML, CSS, JS  
6 Code, Compile, Run and Debug online from anywhere in world.  
7  
8 *****/  
9 #include <stdio.h>  
10  
11 int main()  
12 {  
13     printf("Hello World");  
14  
15     return 0;  
16 }  
17
```

Language dropdown menu options:

- select --
- C
- C++
- C++ 14
- C++ 17
- Java**
- Python 3
- PHP
- C#
- VB
- HTML,JS,CSS
- Ruby
- Perl
- Pascal
- R
- Fortran
- Haskell
- Assembly(GCC)
- Objective C
- SQLite



実行ボタン



▶ Run

⊞ Debug

■ Stop

↻ Share

💾 Save

{ } Beautify

Main.java

エディタ画面

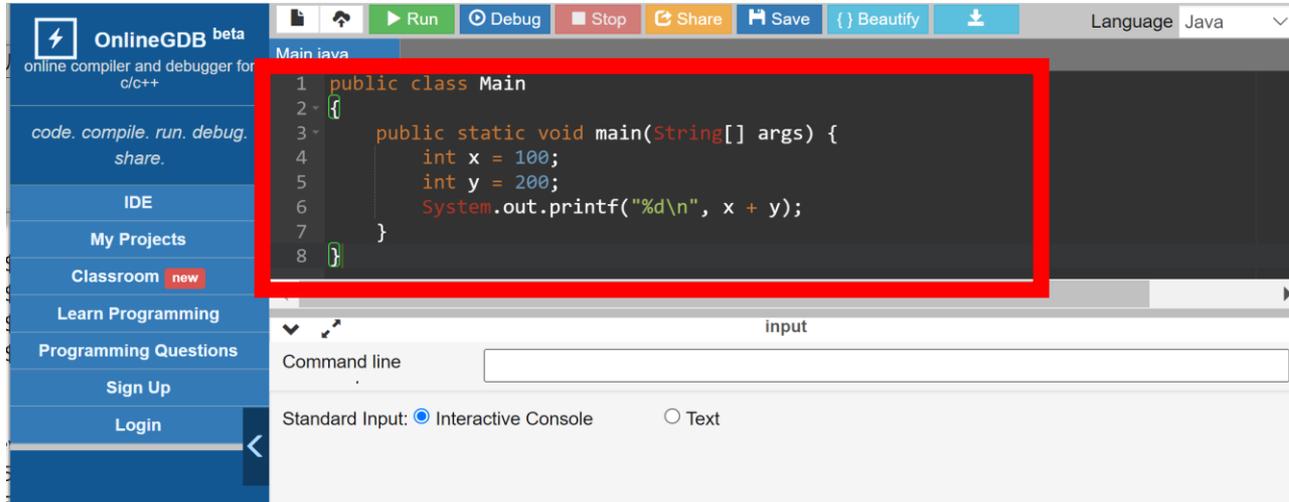
```
1 /*****  
2  
3 Welcome to GDB Online.  
4 GDB online is an online compiler and debugger t  
5 C#, VB, Swift, Pascal, Fortran, Haskell, Object  
6 Code, Compile, Run and Debug online from anywhe  
7  
8 *****/  
9 public class Main  
10 {  
11     public static void main(String[] args) {  
12         System.out.println("Hello World");  
13     }  
14 }  
15
```

プログラムを
書き換えること
ができる

input

```
Hello World  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```

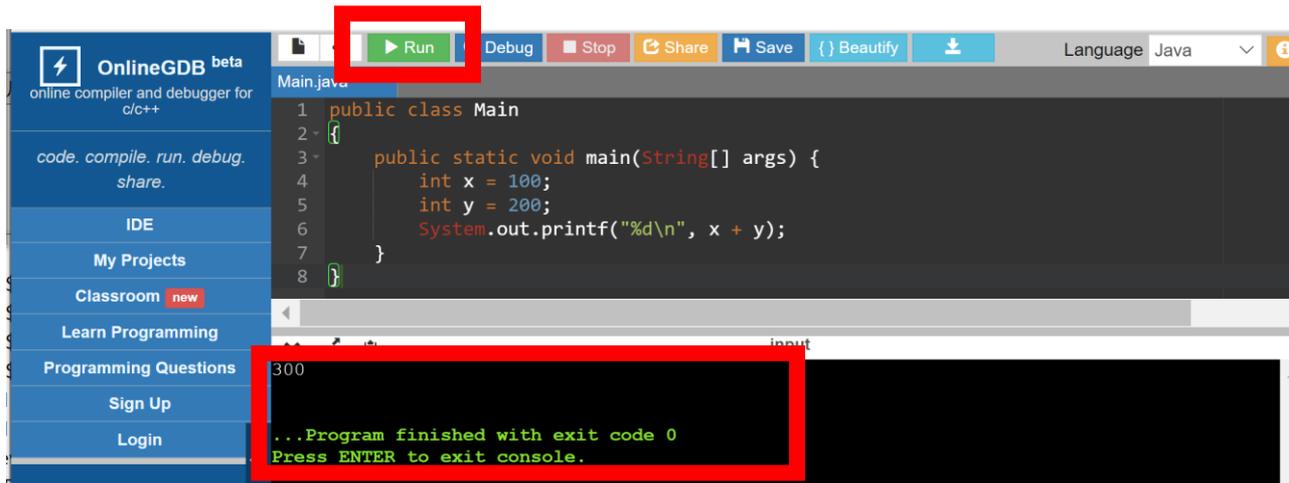
④ ソースコードを入れる



```
1 public class Main
2 {
3     public static void main(String[] args) {
4         int x = 100;
5         int y = 200;
6         System.out.printf("%d\n", x + y);
7     }
8 }
```

⑤ 実行. 実行結果を確認

「Run」をクリック.



```
300
...Program finished with exit code 0
Press ENTER to exit console.
```

1-5. 計算誤差

コンピュータで「 $1 \div 3$ 」を求めると どうなると思いますか



1. 0.333 と無限に表示される
2. 計算できない
3. 正確な値が表示されない（誤差を含む）

演習

資料：43 ~ 45

【トピックス】

- ・ 計算誤差

① Java Tutor のエディタで次のプログラムを入れる

```
public class YourClassNameHere {  
    public static void main(String[] args) {  
        System.out.println(1.0/3.0);  
    }  
}
```



② 実行するために, 「**Visual Execution**」 をクリック. そして 「**Last**」 をクリック. 結果を確認

結果を確認

```
Java 8  
(known limitations)  
1 public class YourClassNameHere {  
2     public static void main(String[] args) {  
3         System.out.println(1.0/3.0);  
→ 4     }  
5 }
```

[Edit this code](#)

at just executed
ne to execute

<< First < Prev Next > Last >>

Done running (3 steps)

Print output (drag lower right corner to expand)

0.3333333333333333

Frames

main:4
Return value void

誤差がある

③ 「**Edit this code**」 をクリックして, エディタの画面に戻る

④ Java Tutor のエディタで次のプログラムを入れる

```
public class YourClassNameHere {  
    public static void main(String[] args) {  
        System.out.println(6 * 1.1);  
    }  
}
```



⑤ 実行するために、「**Visual Execution**」をクリック。そして「**Last**」をクリック。結果を確認

結果を確認

Java 8
([known limitations](#))

```
1 public class YourClassNameHere {
2     public static void main(String[] args) {
3         System.out.println(6 * 1.1);
4     }
5 }
```

[Edit this code](#)

that just executed
line to execute

<< First < Prev Next > Last >>

Done running (3 steps)

Print output (drag lower right)

6.6000000000000005

誤差がある

Frames

main:4
Return value void

⑥ 「**Edit this code**」をクリックして、エディタの画面に戻る

⑦ Java Tutor のエディタで次のプログラムを入れる

```
public class YourClassNameHere {  
    public static void main(String[] args) {  
        System.out.println(3 * 1.1);  
    }  
}
```



⑧ 実行するために、「**Visual Execution**」をクリック。そして「**Last**」をクリック。結果を確認

結果を確認

誤差がある

Java 8
([known limitations](#))

```
1 public class YourClassNameHere {
2     public static void main(String[] args) {
3         System.out.println(3 * 1.1);
4     }
5 }
```

[Edit this code](#)

What just executed
Line to execute

<< First < Prev Next > Last >>

Done running (3 steps)

Print output (drag lower right)

3.3000000000000003

Frames

main:4

Return value void

- コンピュータだから「計算が完璧に正確」という
思い込みはしないこと
 - 1 ÷ 3 を計算して表示させると、
正確な値が表示されない（誤差を含む）
- 誤差があっても、十分に役に立つ
- 誤差を許しているから、計算が効率的に済むという
考え方もある

1-6. さまざまなプログラミング言語

プログラミングを学ぶときに気を付けること



- **プログラミング言語**には、種類が数多くある

- **基礎**となる知識が大事.

一度、あるプログラミング言語で**基礎をマスター**しておけば、**他のプログラミング言語**でも**応用が利く**、という考え方も



- 複数のプログラミング言語を学ぶことは大事.

賛成できますか？

プログラミング
言語は複数ある

- 「1つを知っていれば、どの言語も大体似ているので、応用が利く」という考え方もある.
- 「やりたいこと、学びたいことに向いた言語を、そのときどきで選ぶのが、一番良い」とも.
- 人によって「好きな言語が違う」ということも

さまざまなプログラミング言語



- Python
 - C
 - Java
 - JavaScript
 - R
 - Octave
 - Scheme
- など

ここで行う作業

1. 20 より大きければ「big」、
さもなければ「small」と表示
2. $0 + 1 + 2 + 3 + 4 + 5$ を求める

国家資格取得にも関係する
(Java, Python, C/C++)

なぜプログラミング言語は たくさんあるのでしょうか？



それぞれ
特徴があ
る



Java

どのコン
ピュータ
でも同じ
プログラ
ムが動く。

普及度は
トップレ
ベル。



Python

初心者向
け。その
おかげで、
多数の拡
張機能も。



C / C++

コン
ピュータ
の性能を
最大限引
き出す。



R

「データ
処理」に
特化した
コマンド
言語



SQL

「データ
ベース」
に特化し
たコマン
ド言語



MATLAB /
Octave

「数値計
算」，
「信号処
理」など
に特化し
たコマン
ド言語

Python プログラム見本



```
x = 100
if (x > 20):
    print("big")
else:
    print("small")
s = 0
for i in [1, 2, 3, 4, 5]:
    s = s + i
print(s)
```

- すぐに実行できる
- さまざまな「パッケージ」で機能を拡張できる
- Windows でも Linux でも、ほぼ同じプログラムで動く

Java プログラム見本



```
public class Main {
    public static void main(String[] args) throws Exception {
        int x = 100;
        if (x > 20) {
            System.out.printf("big%n");
        } else {
            System.out.printf("small%n");
        }
        int s = 0;
        for(int i = 1; i <= 5; i++) {
            s = s + i;
        }
        System.out.printf("%d%n", s);
    }
}
```

- Windows でも Linux でも Android アプリでも, 同じプログラムで動く

C プログラム見本



```
#include <stdio.h>
int main(void){
    int x, s, i;
    x = 100;
    if (x > 20) {
        printf("big¥n");
    } else {
        printf("small¥n");
    }
    s = 0;
    for(i = 1; i <= 5; i++) {
        s = s + i;
    }
    printf("%d¥n", s);
    return;
}
```

- ・コンピュータの決め細かなコントロール
- ・高速実行できるチューニング

JavaScript プログラム見本



Webアプリに向く

```
process.stdin.resume();
process.stdin.setEncoding('utf8');
var util = require('util');
var x = 100;
if (x > 20) {
    process.stdout.write('big¥n');
} else {
    process.stdout.write('small¥n')
}
var s = 0;
for(var i = 1; i <= 5; i++) {
    s = s + i;
}
process.stdout.write(util.format('%d¥n', s));
```

R プログラム見本



データ専門家向け

```
x <- 100
if (x > 20) {
  print("big")
} else {
  print("small")
}
s <- 0
for (i in c(1,2,3,4,5)) {
  s <- s + i
}
print(s)
```

Octave プログラム見本



```
x = 100
if (x > 20)
    printf("big¥n")
else
    printf("small¥n")
endif
s = 0
for i = [1 2 3 4 5]
    s = s + i
endfor
printf("%d", s)
```

行列計算, 信号処理など
に向く

Scheme プログラム見本



関数型言語

```
(define (decide x)
```

```
  (cond
```

```
    ((> x 20) "big")
```

```
    (else "small")))
```

```
(define (sum n)
```

```
  (cond
```

```
    ((= n 0) 0)
```

```
    (else (+ (sum (- n 1)) n))))
```

```
(begin
```

```
  (print (decide 100))
```

```
  (print (sum 5)))
```

1-7 この授業の全体計画

この授業

- **プログラム**に上達するとよいことがたくさんある
- この授業では、**プログラミング**に関する基礎、大切なことを学ぶ。
- **プログラミング**の初心者を対象

入門

- オブジェクト
- メソッド
- データの種類
- クラス
- 配列
- 条件分岐, 繰り返し

この授業の 主な内容

発展

- **バグのないプログラム**を作成するのに役立つ実践
- **プログラムの設計法**
- プログラムでできる**種々の機能** (タイマー, グラフィックスなど)

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>