

# プログラミング基礎

プログラミング言語 Java (1) 概要

# 今日の学習目標



- Javaに関連した用語・概念について知る
- Javaでのプログラムの書き方を体験してみる

# プログラミング言語って何？

- プログラム＝命令書
- プログラミング＝命令書を作成すること
- プログラミング言語＝人間用言語
- コンパイル＝人間用言語 → コンピュータ言語
- コンパイラ＝翻訳者
- インタプリタ＝通訳者

# Javaって何だろう？

- (一応)コンパイラ型のプログラミング言語
- オブジェクト指向プログラミング言語
  - ▣ オブジェクト(部品)の組み合わせでプログラムを作成
  - ▣ オブジェクト=Scratchで言うところのスプライト

# (Javaにおける)オブジェクト指向

- パッケージ=オブジェクトの属する業界
- オブジェクト
  - ▣ クラス(1個しかない)
  - ▣ インスタンス(幾つでも作られる)

# オブジェクトの中身



- そのオブジェクトの変数(メンバ変数)
- そのオブジェクトが行える仕事(メソッド)

# 猫のオブジェクトを作ってみる

(Windowsの場合)

- 以下のプログラムをメモ帳で入力する
  - ※ コンパイラは細かい間違いに厳しい
    - 大文字小文字なども間違いの無いように注意！
- 入力したら「Cat.java」という名前で保存
  - ※ ファイルの種類を「すべて」にする

```
class Cat {  
    void cry() {  
        System.out.println("Mew!!");  
    }  
}
```

# 猫のオブジェクトを作ってみる

(Macの場合)

- 以下のプログラムをテキストエディットで入力する

※「フォーマット」メニューの「標準テキストにする」で標準テキストにして作成

- 入力したら「Cat.java」という名前で保存

※ 保存するときに拡張子を .java にする

```
class Cat {  
    void cry() {  
        System.out.println("Mew!!");  
    }  
}
```

コンパイラは細かいところに厳しい！  
大文字・小文字の区別も間違いないように入力しよう



# Javaのルール

---

- 文は ; (セミコロン) で終わる
- 範囲は { ~ } (中カッコ) で囲む

# コンパイルしてみる

(Windowsの場合)

- コマンドプロンプトを起動
- 先ほど保存した Cat.java の場所に移動
  - ※ dir と cd で頑張って移動しよう(授業中に説明する)
- 以下のコマンドを実行してコンパイルする

```
javac Cat.java
```

- dir コマンドを実行
  - ▣ Cat.class というファイルができていることを確認
  - ▣ これがコンピュータの言語に翻訳されたプログラム

# コンパイルしてみる

(Macの場合)

- ターミナルを起動
- 先ほど保存した Cat.java の場所に移動
  - ※ ls と cd で頑張って移動しよう(授業中に説明する)
- 以下のコマンドを実行してコンパイルする

```
javac Cat.java
```

- ls コマンドを実行
  - ▣ Cat.class というファイルができていることを確認
  - ▣ これがコンピュータの言語に翻訳されたプログラム

# Javaで作成したプログラムの実行

- main という決まった名前のメソッドから実行開始
- 基本的には順番に命令が実行される
  - ▣ 「繰り返し」や「条件分岐」があると順番が変更される

# 実行開始用オブジェクトを作る

(Windowsの場合)

- 以下のプログラムをメモ帳で入力する
  - ※ コンパイラは細かい間違いに厳しい
    - 大文字小文字なども間違いの無いように注意！
- 入力したら「Main.java」という名前で保存
  - ※ ファイルの種類を「すべて」にする

```
class Main {  
    public static void main(String[] args) {  
        Cat mike;  
        mike = new Cat();  
        mike.cry();  
    }  
}
```

# 実行開始用オブジェクトを作る

(Macの場合)

## □ 以下のプログラムをテキストエディットで入力する

※ コンパイラは細かい間違いに厳しい

→ 大文字小文字なども間違いの無いように注意！

※ 「フォーマット」メニューの「標準テキストにする」で標準テキストにして作成

## □ 入力したら「Main.java」という名前で保存

※ 保存するときにファイルの拡張子を「.java」にする

```
class Main {  
    public static void main(String[] args) {  
        Cat mike;  
        mike = new Cat();  
        mike.cry();  
    }  
}
```

# コンパイル＋実行

- 先ほどと同じように Main.java をコンパイル
- 以下のように入力してプログラムを実行させる

```
java Main
```

- 画面に「Mew!!」と表示されたらOK

# オブジェクトの作成

- クラス＝プログラムの実行時に自動的に作成
  - ※ 1つしか無いからそれでいい
- インスタンス＝プログラム中で必要に応じて作成
  - ▣ new 命令で新しいインスタンスを作る

```
mike = new Cat();
```
  - ▣ 作ったインスタンスを変数(mike)に入れておく
    - インスタンスの区別をするため