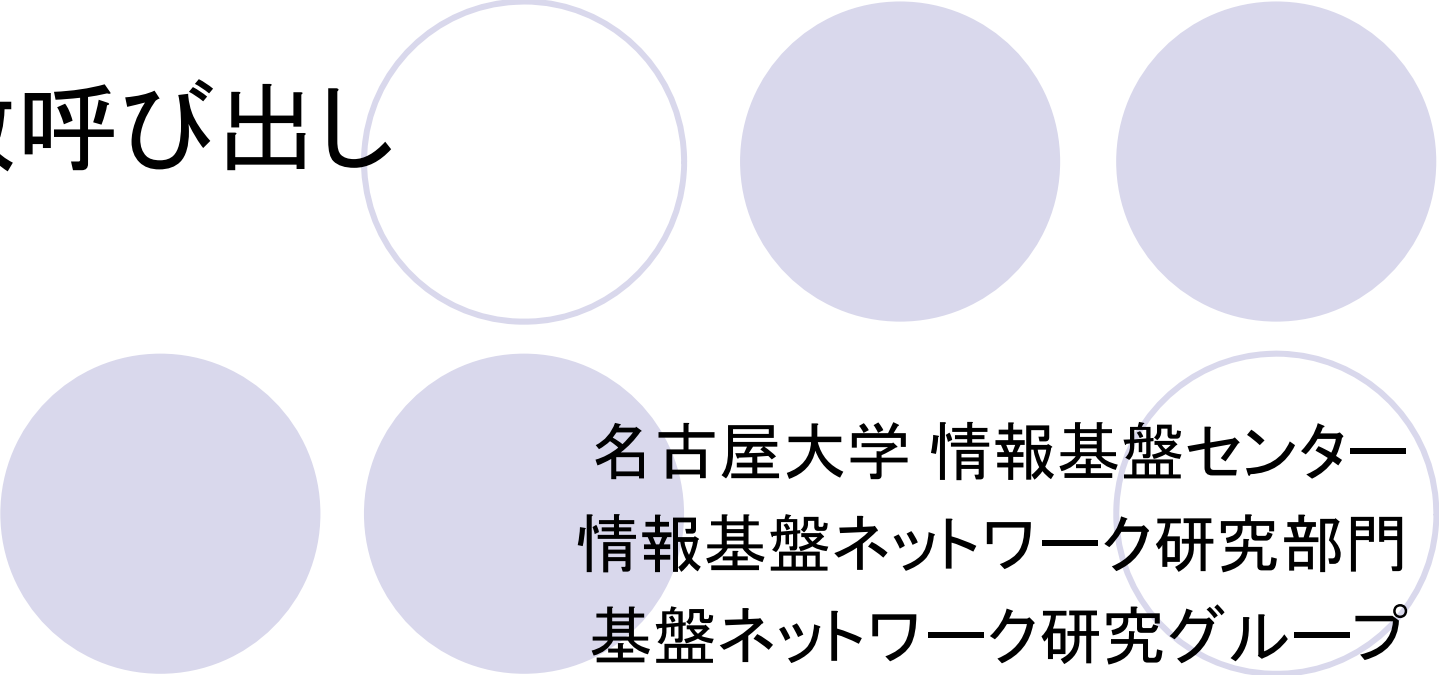


関数呼び出し



名古屋大学 情報基盤センター
情報基盤ネットワーク研究部門
基盤ネットワーク研究グループ

嶋田 創

ポイント

- 同じ処理は関数として定義して、必要な所で呼び出せるようになる
 - ライブラリの関数などもこの形で実装されている
- 関数に渡す値(引数)、関数からの戻り値の利用を理解する
- 関数の中の変数はローカル変数になり、関数外からは見れない
- ある関数の中からその関数自身を呼び出す「再帰呼び出し」をうまく使うと、プログラムがすごく単純になることがある

関数定義

- 関数の定義とその構造

```
def 関数名(引数1, 引数2, ...):
```

 関数の中の処理

```
return 戻り値とする値or変数
```

- 引数とは呼び出し元から関数に渡す値
- 戻り値とは、関数から呼び出し元に戻す値
 - 戻り値は関数内の変数でも固定値でも良い
- 必ず、呼びされる場所より手前(上)の行で定義すること
 - プログラムは上の行から順番に解釈されるため、呼び出した場所より後ろで定義されていると「そんな関数はない!」とエラーが出る
- 関数内で定義された変数は関数内でのみ使える(ローカル変数)
- 関数の呼び出し: 関数名(引数1, 引数2, ...)

関数の引数と戻り値

- 引数とは「関数呼び出し元から、関数に渡す値」
 - 引数無し関数というのもあり
- 引数として値を記せば値が、変数を記せば「変数の中の値」が渡される
 - 「変数自体」が渡されない点に注意 → 関数内で引数の値を変更しても、呼び出し元の変数の値は変わらない
- 引数は関数の宣言の時の呼び出し時の括弧内に記された順番で渡される(指定の無い限り)
- 戻り値はカンマで区切って複数書くこともできる
 - 例: `return pi, 'Success'`
 - 複数の戻り値はタプルの形になる(整数、文字列などを混在可能)
 - 他の言語だと複数の戻り値を設定できないことが多いので注意

関数と呼び出しの例

```
#!/bin/env python3
```

```
def volume_cone (radius, height) :  
    pi = 3.14  
    return radius * radius * pi * height / 3
```

} この部分が関数定義
(必ず呼び出される 前の場所で定義)

```
print(volume_cone(3, 4))  
# print(pi)
```

- 参考: 末尾にprint(pi)と書いて実行してみると、「Error: name 'pi' is not defined」が出ます
 - piはローカル変数なので、関数の外では参照できない

ローカル変数とグローバル変数

- 関数の中の変数はローカル変数となる
 - 関数内でのみ有効な変数
 - 変数名がかぶっても大丈夫なありがたい仕様(特にライブラリ)
 - 引数はローカル変数となる
 - 関数内で新たに定義した変数はローカル変数となる
- 逆に、関数外の変数はグローバル変数となる
 - 同じPythonプログラムファイル内ならどこでも利用可能
 - 「本当の意味でのグローバル変数」はimportなどを行った別ファイルからでも利用可能なので、本当の意味でのグローバル変数ではない
 - Pythonは本来の意味でのグローバル変数を(まず)必要としない設計になっているので、特に困ることは無いはず
 - 関数内でグローバル変数を使いたい場合は、関数定義の前にグローバル変数を定義すること

ローカル変数とグローバル変数の仕様の確認(1/2)

```
#!/bin/env python3
```

```
pi = 3.14
```

```
def volume_cone (radius, height) :  
    return radius * radius * pi * height / 3
```

```
print(volume_cone(3, 4))
```

```
print(pi)
```

- グローバル変数として定義されたpiは関数内でも関数外でも利用可能

ローカル変数とグローバル変数の仕様の確認(2/2)

```
pi = 3.14
```

```
def volume_cone (radius, height) :  
    return radius * radius * pi * height / 3
```

```
def volume_cone_fine_grain (radius, height) :  
    pi = 3.1415926535  
    return radius * radius * pi * height / 3
```

```
print(volume_cone(3, 4))  
print(volume_cone_fine_grain(3, 4))  
print(pi)
```

- 関数volume_coneではグローバル変数のpi = 3.14が使われる
- 関数volume_cone_fine_grainではローカル変数のpi = 3.1415926535が使われる

関数の再帰呼び出し

- 「関数aの中から関数aを呼び出す」ことを関数の再帰呼び出しと呼ぶ

```
def recur (x, n)
```

```
    y = ....
```

```
    if n >= 0 :
```

```
        y = y + recur(x, n-1)
```

```
    return y
```

- 一部のアルゴリズムでは、関数の再帰呼び出しを使うと実装がすごく簡単になることがある
 - 一部のソートアルゴリズム(クイックソートとか)の関数化
 - 漸化式の計算の関数化

関数のTips

- 引数にデフォルト値を指定する: `def 関数名 (引数1 = デフォルト値1, 引数2 = デフォルト値2, ...)`
 - デフォルト(default): 初期値とか既定値とか(債務不履行ではない)
 - 一部の引数だけデフォルト値を設定もOK
- 呼び出し時に引数名を指定することで、引数の順番を無視可能

例: `関数名(引数名1 = 定数or変数, 引数名2 = 定数or変数, ...)`

 - デフォルト値の設定と同時に使うと便利