

# pf-6. 条件分岐, ステップ実行

(Python 入門)

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

金子邦彦





①コンピュータでのプログラム実行は、通常実行が基本

②プログラムの流れの制御

# 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 **11**以下 → **500**  
**12**以上 → **1800**

条件式は「**age <= 11**」のようになる

# 条件分岐の Python プログラム



**age** の値が **11**以下 → **500**  
**12**以上 → **1800**

```
age = 18
if age <= 11:
    print(500)
else:
    print(1800)
```

条件式は「**age <= 11**」のようになる

- Trinket は**オンライン**の Python、HTML 等の**学習サイト**
- 有料の機能と無料の機能がある
- 自分が作成した Python プログラムを公開し、他の人に実行してもらうことが可能（そのとき、書き替えて実行も可能）
- Python の標準機能を登載、その他、次のモジュールやパッケージがインストール済み

math, matplotlib.pyplot, numpy, operator, processing, pygal, random, re, string, time, turtle, urllib.request

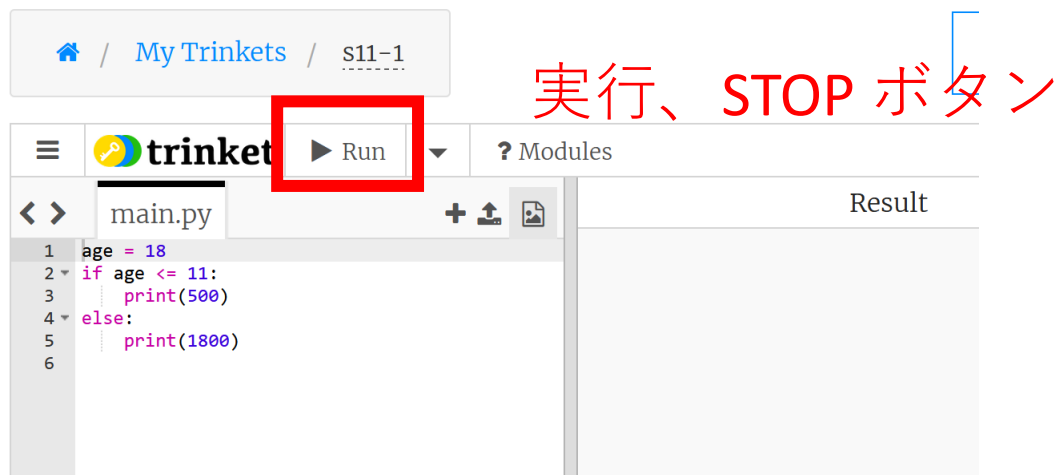
# trinket でのプログラム実行



- trinket は Python, HTML などのプログラムを書き実行できるサイト

- <https://trinket.io/python/0fd59392c8>

のように、違うプログラムには違う URL が割り当てられる



- 実行が開始しないときは、「**実行ボタン**」で**実行**
- ソースコードを書き替えて再度実行することも可能

# 演習

資料： 8, 9

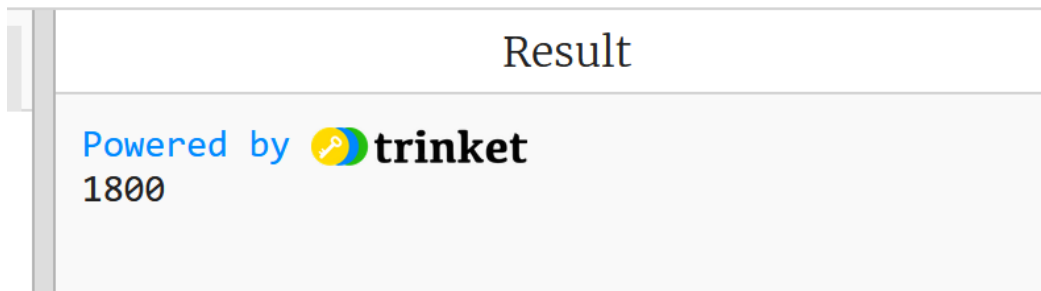
## 【トピックス】

- 条件分岐
- if
- else

① trinket の次のページを開く

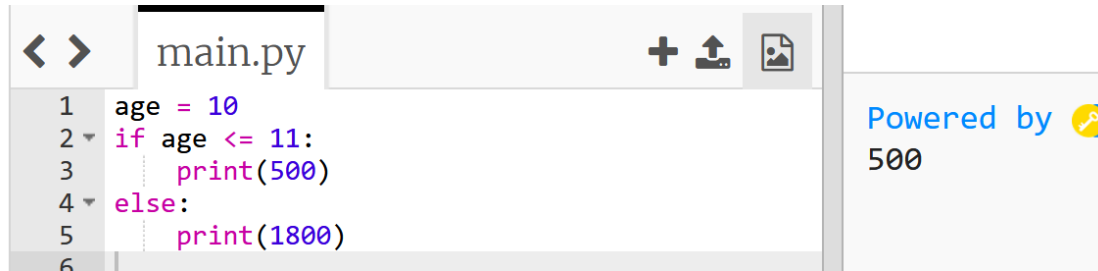
<https://trinket.io/python/0fd59392c8>

② 実行する。1800 が表示されることを確認







③ 「age = **18**」を「age = **10**」に書き替える




```
1 age = 10
2 if age <= 11:
3     print(500)
4 else:
5     print(1800)
6
```

Powered by  500

④ 実行する。500 が表示されることを確認



Result

Powered by  trinket

500

⑤ age の値が 8, 9, 10, 11 のときは 500 になり、  
12, 13, 14, 15 のときは 1800 になることを確認



```
age = 18  
if age <= 12:  
    print(500)  
else:
```

```
    print(1800)
```



1800

age の値が **13以上**の  
ときは、結果は **1800**



```
age = 10  
if age <= 12:  
    print(500)  
else:
```

```
    print(1800)
```

500

age の値が **12以下**の  
ときは、結果は **500**

## 演習

演習問題を解いてみる

資料：12, 13

### 【トピックス】

- 条件分岐
- if
- else

# 条件分岐



- 次のプログラムを作成

① **weight** と料金の関係は次の通り

**weight** の値が **100以下** → **0**

**100より大きい** → **1000**

② **weight = 80** に設定してテスト実行





# 正解の例







trinket のページ

<https://trinket.io/python/62f74d3bfc>

 / [My Trinkets](#) / s11-2

**trinket** Run ? Modules

 **main.py**   

```
1 weight = 80
2 if weight <= 100:
3     print(0)
4 else:
5     print(1000)
```

Powered by  
0

# 条件分岐 まとめ



- **条件分岐**は、特定の条件に基づいて、異なる結果を得ることを可能にする
- Python の**条件分岐**では、if, else などのキーワードを使用

ある映画館で、11歳以下のチケットと、12歳以上のチケットで値段の違いがあるとき、条件分岐を使用して、チケット料金を算出できる

```
age = 18
if age <= 11:
    print(500)
else:
    print(1800)
```

age <= 11 のときは、print(500) が実行される

そうでないときは、print(1800) が実行される

# ステップ実行



- **ステップ実行**では、**1行ずつの実行**が行われ、そのときの  
変数の値の変化などを**確認**できる
- **ステップ実行**により、**プログラムの動作を細かく追跡**でき、  
不具合が発生している箇所の特定、プログラムの学習に役  
立つ
- **通常実行**は、**プログラムを最初から最後まで一度に実行**す  
るもの（プログラム実行中の変数の値の変化を確認するな  
どは困難）。**ステップ実行**は、**プログラムを1行ずつ実行**  
し、**実行後にプログラムを一時停止**するもの。

# 条件分岐の Python プログラム



**age** の値が **11**以下 → **500**  
**12**以上 → **1800**

```
age = 18
if age <= 11:
    print(500)
else:
    print(1800)
```

条件式は「**age <= 11**」のようになる



# Python Tutor

- **Python Tutor** という**ウェブサ**  
**イト**を利用しよう

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

- **Web ブラウザ**を使ってアクセ  
スできる

- **PythonTutor** では, **Pythonだ**  
**けでなく**, Java, C,, C++,  
JavaScript, Ruby など, 多く  
のプログラミング言語を学ぶ  
ことができる.



Python 3.6  
[known limitations](#)

```
1 x = 100
2 if (x > 20):
3     print("big")
4 else:
5     print("small")
6 s = 0
7 for i in [1, 2, 3,
8     s = s + i
→ 9 print(s)
```

[Edit this code](#)

just executed  
to execute

<< First

< Prev

Next >

Done running (16 steps)

# Python Tutor の使用方法



- ① まず, **ウェブブラウザ**を開く
- ② **Python Tutor** を利用するために, 以下の URL にアクセス

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

- ③ 「**Python**」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:

# Python Tutor の編集画面



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

Write code in Python 3.6 「Python 3.6」になっている

1 |

**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

hide exited frames [default] ☐ inline primitives, don't nest objects [default] ☐

draw pointers as arrows [default] ☐

[Show code examples](#)

Generate permanent link

# Python Tutor でのプログラム実行



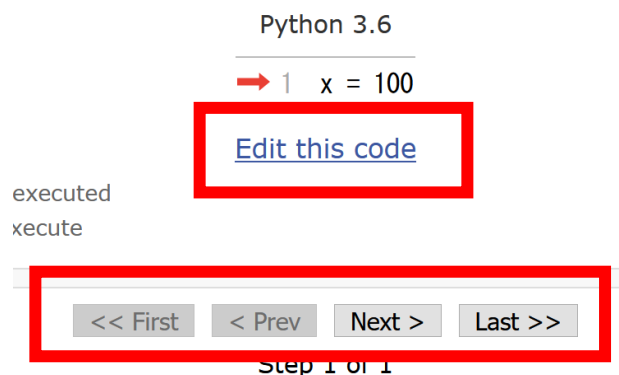
- Python Tutor は Python などのプログラムを書き実行できるサイト。ステップ実行、変数の値表示などの機能がある。
- Python Tutorのウェブサイトアクセス。「Python」を選択  
<https://www.pythontutor.com/>

メイン画面で、プログラムを書く



Visualize Execution ボタン

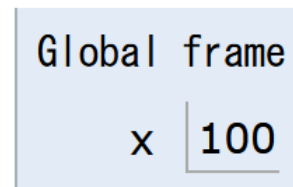
メイン画面に  
戻るには  
Edit this code



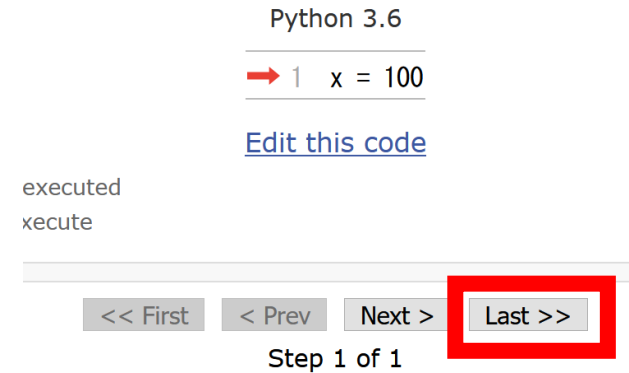
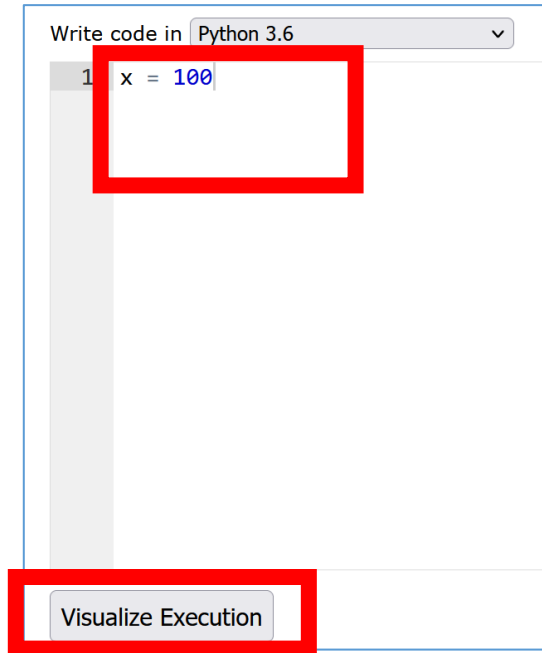
通常実行: Last  
ステップ実行: 他のボタン

変数の値を  
視覚的に  
確認できる

Frames

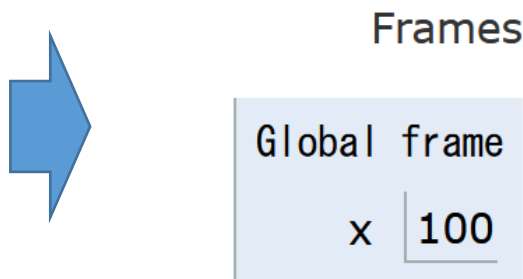


# Python Tutor でのプログラム実行手順



(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。



(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

# Python Tutor 使用上の注意点①



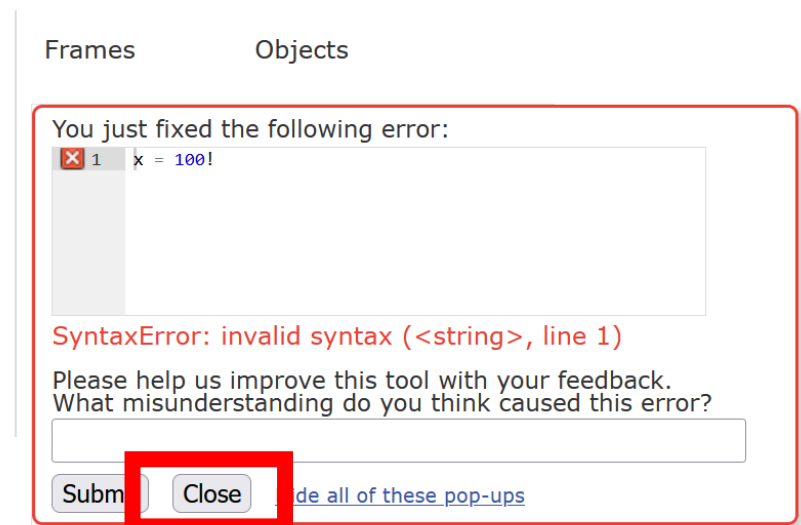
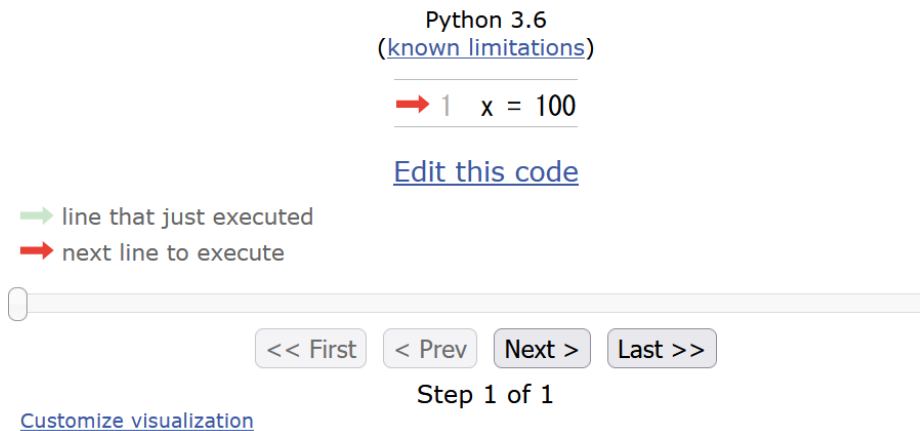
実行画面で、**赤いエラーメッセージ**が出ることがある

過去の文法ミスに関する確認表示.

基本的には、**無視**して問題ない

邪魔なときは「**Close**」

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



# Python Tutor 使用上の注意点②



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



- Python Tutor が混雑しているとき， 「Server Busy . . .」 と表示される場合がある。
- このメッセージは，サーバが混雑していることを示す。
- 数秒から数十秒待つと自動で処理が始まるはずですが（しかし，表示が変わらないときは，操作をもう一度試してください）

## 演習

資料 : 25 ~ 35

### 【トピックス】

- Python Tutor
- 字下げ
- :
- 条件分岐
- if
- else
- ステップ実行



# ステップ実行により確認できること



ステップ実行により，ジャンプの様子を観察

ジャンプの  
様子を示す  
矢印

```
Python 3.6
1 age = 30
2 if age <= 12:
3     print(500)
4 else:
5     print(1200)
```

[Edit this code](#)

Print output (drag lower right)

Frames

Global frame  
age 30

変数の値の  
確認もできる

uted  
e

<< First

< Prev

Next >

Last >>

Step 3 of 3

ステップ実行は、  
ボタンやスライダーでコントロール

# Python Tutor の起動



① ウェブブラウザを起動する

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

③ 「Python」をクリック ⇒ メイン画面が開く

## 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:

## ④ Python Tutor のエディタで次のプログラムを入れる

```
1 age = 18
2 if age <= 11:
3     print(500)
4 else:
5     print(1800)
```

if (age <= 11)の直後に「:」  
else の直後に「:」  
(どちらも、コロン)

字下げも正確に！

print の前に、「タブ (Tab)」を1つだけ

```
1 age = 18
2 if age <= 11:
3     print(500)
4     else:
5         print(1800)
```

正しくない字下げ

「delキー」などを使い  
ながら編集

```
1 age = 18
2 if age <= 11:
3     print(500)
4 else:
5     print(1800)
```

正しい字下げ

⑤ 通常実行するために, 「Visual Execution」 を  
クリック. そして「Last」 をクリック. 結果 1800  
を確認

Python 3.6  
[known limitations](#)

```
1 age = 18
2 if age <= 11:
3     print(500)
4 else:
→ 5     print(1800)
```

[Edit this code](#)

Executed  
Output

Print output (drag lower right corner to

1800

Frames

Objects

Global frame

age 18

結果の  
「1800」を確認

<< First

< Prev

Next >

Last >>

Done running (3 steps)



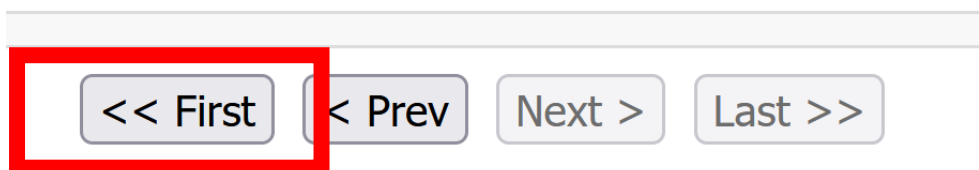
## ⑥プログラム実行を最初の行に戻す操作 「**First**」をクリックして、最初の行に戻す

Python 3.6  
[known limitations](#)

```
1 age = 18
2 if age <= 11:
3     print(500)
4 else:
5     print(1800)
```

[Edit this code](#)

executed  
xecute



Done running (3 steps)

- ⑦ 「**Step 1 of 3**」と表示されているので、  
全部で、**ステップ数**は**3**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**)

Python 3.6  
[known limitations](#)

```
→ 1 age = 18
   2 if age <= 11:
   3     print(500)
   4 else:
   5     print(1800)
```

[Edit this code](#)

ecuted  
cute

<< First

< Prev

Next >

Last >>

Step 1 of 3

⑧ **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」ボタンを何度かクリックし、それ以上進めなくなったら終了

見どころ  
2行目から 5行目へ  
**ジャンプ**するところ

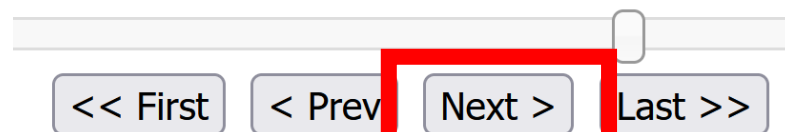


Python 3.6  
[known limitations](#)

```
1 age = 18  
→ 2 if age <= 11:  
3     print(500)  
4 else:  
→ 5     print(1800)
```

[Edit this code](#)

executed  
ecute



Step 3 of 3



## ⑨ First, Prev, Next, Last ボタンとスライダーによりプログラム実行を制御してみる

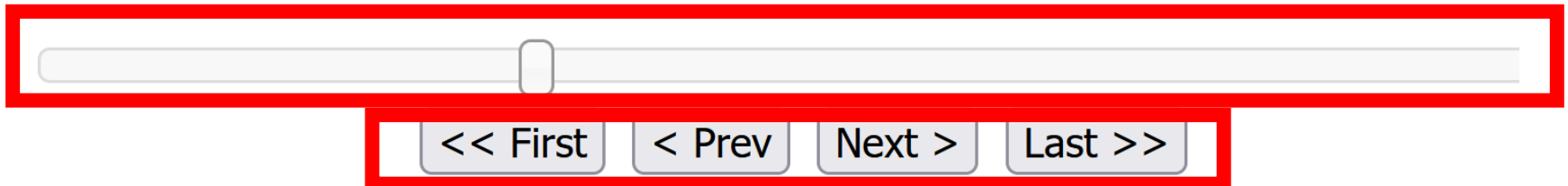
Python 3.6  
[known limitations](#)

```
→ 1 age = 18  
→ 2 if age <= 11:  
   3     print(500)  
   4 else:  
   5     print(1800)
```

[Edit this code](#)

→ line that just executed

→ next line to execute



Step 2 of 3

⑩ 1行目を，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

Python 3.6  
[known limitations](#)

```
1 age = 10
2 if age <= 11:
→ 3     print(500)
4 else:
5     print(1800)
```

[Edit this code](#)

→ line that just executed

→ next line to execute

<< First

< Prev

Next >

Last >>

Done running (3 steps)

Print output (drag

500

Frames

Global frame

age 10

# ⑪ メイン画面に戻るには、「Edit this code」をクリック



Python 3.6  
[known limitations](#)

```
1 age = 10
2 if age <= 11:
→ 3     print(500)
4 else:
5     print(1800)
```

[Edit this code](#)

→ line that just executed

→ next line to execute

Print output (drag)

500

Frames

Global frame

age 10

<< First

< Prev

Next >

Last >>

Done running (3 steps)

# ステップ実行 まとめ



- **通常実行**は、プログラムを最初から最後まで一度に実行する
- **ステップ実行**は、プログラムを1行ずつ実行し、実行後にプログラムを一時停止するもの
- **ステップ実行**により、プログラムの動作を細かく追跡でき、不具合が発生している箇所の特特定、プログラムの学習に役立つ

## ①コンピュータでのプログラム実行は、通常実行が基本

**通常実行**では、プログラムは、最初から最後まで一度に実行される。途中の変数の値を観察するには print やステップ実行を活用しよう。

## ②プログラムの流れの制御

**条件分岐**（if など）では、特定の部分のみ実行される。通常実行とは異なり、プログラムの流れが制御される。

# 全体まとめ



- **通常実行**は、プログラムを最初から最後まで一度に実行する
- **ステップ実行**は、プログラムを1行ずつ実行し、実行後にプログラムを一時停止する。プログラムの動作を細かく追跡できる。
- **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。**年齢 (age) が11以下**であれば500を、それ以上であれば1800を出力するといった場合、**条件式**は「**age <= 11**」となる。