


cs-8.式と変数, 条件分岐, リストと その操作, 繰り返し処理

(コンピューターサイエンス)

URL:<https://www.kkaneko.jp/cc/cs/index.html>

金子邦彦



- 
- ① **式と変数**：変数は、値を保存し参照する仕組み。計算に利用。
 - ② **条件分岐 (if else)**：条件に応じて処理を変える制御構造。
 - ③ **リスト**：複数の要素を順序付けて格納するデータ型。
 - ④ **繰り返し処理**：同じ処理を複数回実行する制御構造

プログラミングの基本になる変数，条件分岐，リスト，繰り返しの理解。オンラインツール (trinket) を活用した実践的な演習

アウトライン

1. 式と計算
2. 条件分岐
3. 演習問題
4. リストと繰り返し

- **プログラミング**は**人間の力を増幅**し、私たちができることを大幅に広げる
- **プログラミング**はさまざまな分野で活用されている
 - シミュレーション：複雑な現象をモデル化し、予測や分析を行える
 - 大量データ処理：データの収集、加工、分析
 - AI（人工知能）システムの開発
 - Webサイト、アプリケーションなどのソフトウェア
- **プログラミングはクリエイティブな行為**
- さまざまな**作業を自動化**したいとき、**問題解決**したいときにも役立つ

プログラミングの楽しさと達成感



• 楽しさ

- **未来の技術を学ぶことは楽しい。**
- **プログラミングは自分のアイデアを形にできるクリエイティブな行為。**
- **視覚的なプログラムを書くことで、ゲーム感覚をもって楽しみながら学習することも可能 | 。**

• 達成感

- **プログラミングを通じて問題解決のスキルを身につけることは、大きな達成感につながる**
- **新しいソフトウェアや技術を生み出す。**

- Trinket (Python) などを実行できるオンライン | 学習環境) 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

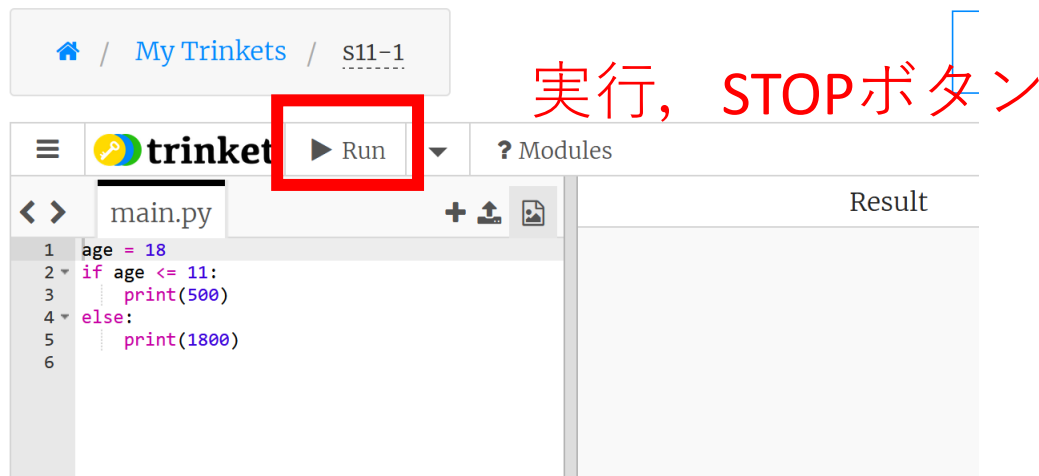
math,matplotlib.pyplot,numpy, operator, processing,pygal,
random, re, string, time, turtle,urlib.request



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



ソースコードの
メイン画面

実行結果

- 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

8-1.式と計算

変数, 代入

- **変数** : プログラム内で名前を付けて利用する オブジェクト で, 値を保存 し, 後から 参照 できる
- **代入** : 「 $x = 100$ 」のように書くことで, x という名前の変数に, 値100が保存 される

例

```
x = 100
```

式の実行結果



式の実行結果として、値が得られる

A screenshot of a code editor interface. The top bar contains icons for a menu, a key, a play button, a dropdown arrow, and a question mark. Below the bar, the editor shows a file named "main.py" with a single line of code: "1 print(100 * 200)". To the right of the editor, the execution result is displayed as "Powered by 20000" with a key icon.

プログラム

実行結果

変数への代入



プログラムで、「**x = 100**」のように書くと、**xの値が100に変化**する

The screenshot shows a code editor interface. At the top, there is a toolbar with icons for a menu, a key, a play button, a dropdown arrow, and a question mark. Below the toolbar, the file name 'main.py' is displayed. The code editor contains two lines of Python code:

```
1 x = 100
2 print("x = ", x)
```

To the right of the code editor, the execution result is shown. It includes the text 'Powered by' followed by a key icon and the word 'tri'. Below this, the output of the print statement is displayed: ('x = ', 100).

プログラム

実行結果

式の中に変数を含めることができる

A screenshot of a code editor interface. The editor title bar shows "main.py" and icons for adding, uploading, and inserting files. The code is as follows:

```
1 x = 100
2 y = 200
3 print(x + y)
```

On the right side of the editor, the execution result is displayed: "Powerexec" in blue text and "300" in black text below it.

プログラム

実行結果

式と変数のPython) プログラム



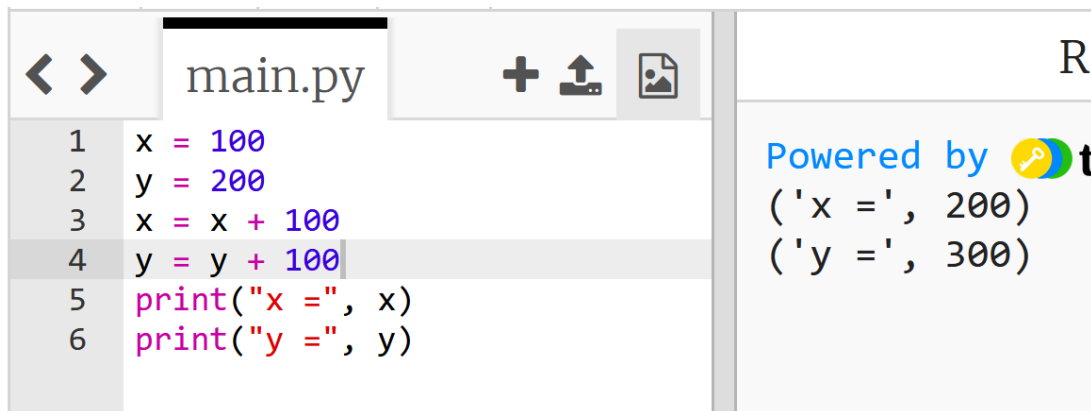
変数x, yに100を足して表示

```
x = 100
y = 200
x = x + 100
y = y + 100
print("x =", x)
print("y =", y)
```

xに100を足す

yに100を足す

printは、メッセージ（文字列）や変数の値を表示



```
< > main.py + [upload] [image] Re
1 x = 100
2 y = 200
3 x = x + 100
4 y = y + 100
5 print("x =", x)
6 print("y =", y)

Powered by [Python logo] t
('x =', 200)
('y =', 300)
```

実行結果

演習 1

変数 x , y を使って計算

ページ 15 ~ 18

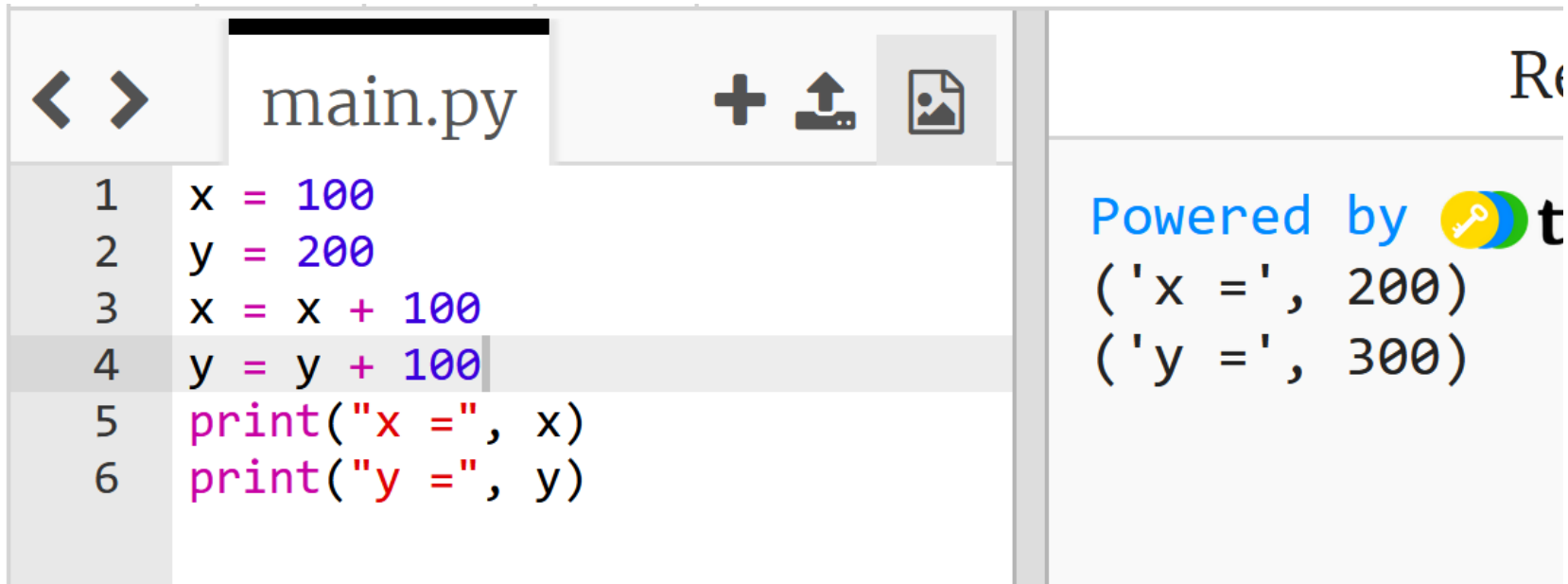
【トピックス】

- trinketの利用
- 式と変数
- 確認クイズに自主的に挑戦

① trinketの次のページを開く

[https://trinket.io/library/trinkets/1e41 | 4fec80](https://trinket.io/library/trinkets/1e41|4fec80)

② 実行結果が、次のように表示されることを確認



```
< > main.py + ⬆️ 📄  
1 x = 100  
2 y = 200  
3 x = x + 100  
4 y = y + 100  
5 print("x =", x)  
6 print("y =", y)  
Re  
Powered by 🔑 t  
( 'x =', 200 )  
( 'y =', 300 )
```

③ 確認クイズ

x, yを**2倍する**プログラムはどちらが正しいであるか？

なお、掛け算には「*」を使う。

正解だと思う方を、自分でコンピュータに入れて、正しいかを確認させてよ

```
x = 100
y = 200
x = x * 2
y = y * 2
print("x =", x)
print("y =", y)
```

```
x = 100
y = 200
2 * x = x
2 * y = y
print("x =", x)
print("y =", y)
```

④ 確認クイズ

x, yの値を**半分にする**プログラムはどちらが正しいであるか？

なお、割り算には「/」を使う。

正解だと思ふ方を、自分でコンピュータに入れて、正しいかを確認させてよ

```
x = 100
y = 200
x = 2
y = 2
print("x =", x)
print("y =", y)
```

```
x = 100
y = 200
x = x / 2
y = y / 2
print("x =", x)
print("y =", y)
```

⑤ 確認クイズ

xとyを足した値を新しいxの値にするプログラムはどちらが正しいであるか？

正解だと思う方を，自分でコンピュータに入れて，正しいか確かめてせよ

```
x = 100  
y = 200  
x = x + y  
print("x =", x)  
print("y =", y)
```

```
x = 100  
y = 200  
x + y  
print("x =", x)  
print("y =", y)
```

答え合わせ



③

```
main.py  
1 x = 100  
2 y = 200  
3 x = x * 2  
4 y = y * 2  
5 print("x =", x)  
6 print("y =", y)
```

Powered by triton
('x =', 200)
('y =', 400)

④

```
main.py  
1 x = 100  
2 y = 200  
3 x = x / 2  
4 y = y / 2  
5 print("x =", x)  
6 print("y =", y)
```

Powered by triton
('x =', 50.0)
('y =', 100.0)

⑤

```
main.py  
1 x = 100  
2 y = 200  
3 x = x + y  
4 print("x =", x)  
5 print("y =", y)
```

Powered by triton
('x =', 300)
('y =', 200)

8-2.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**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

演習 2

条件分岐, if, else

資料 : 21~23

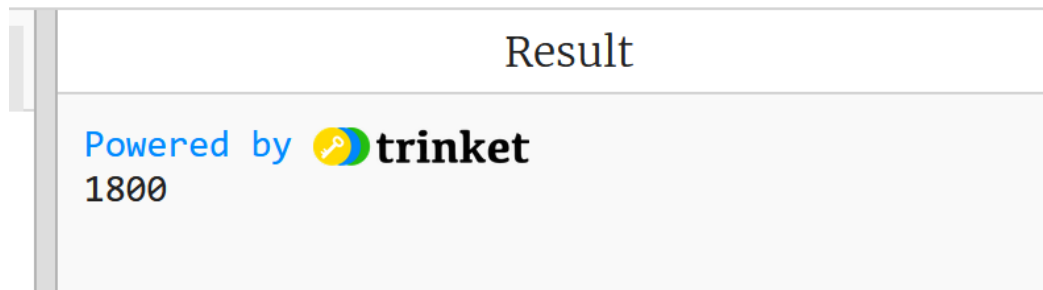
【トピックス】

- 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

```
main.py
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**

```
uics
```

```
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**

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

8-3. 演習問題

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



Powered by

0

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

8-4. リストと繰り返し

Python) のリスト



- リストは、**複数の要素を順序付けて保持**できる
- リストの要素には順序を持ち、**順序の番号は0から開始**する
- リストの要素は**変更可能**（**新しい要素の挿入**，**既存の要素の削除**が可能）

4を末尾に挿入

list


0	1	2	3	4
15	8	6	32	23



list

0	1	2	3	4	5
15	8	6	32	23	4

8の削除



list

0	1	2	3	4
15	6	32	23	4

31

リストと繰り返し

- リストと繰り返しは密接に関連
- リストの各要素に対して同じ操作を行う場合、繰り返し処理を使用すると効率的

```
x = [8, 6, 4, 2, 3]
y = [0, 0, 0, 0, 0]
for i in [0, 1, 2, 3, 4]:
    y[i] = x[i] * 1.1
    print(y[i])
```

演習 3. リストと繰り返し し, for

資料 : 34~41

【トピックス】

- リスト
- 繰り返し
- for

1. 月の日数

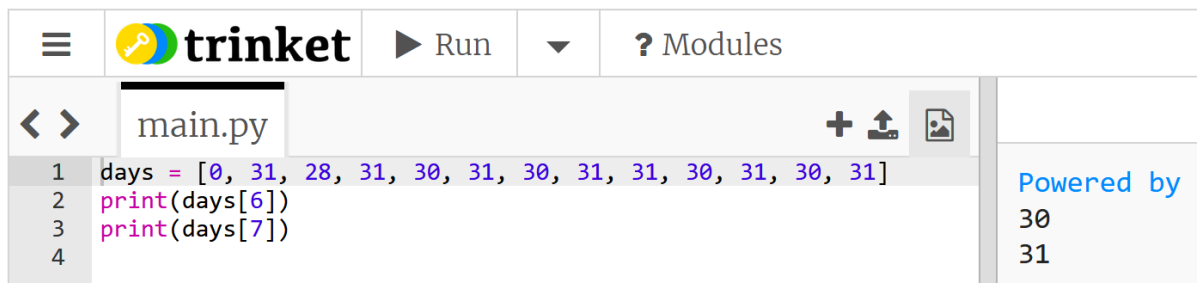
6月は**30**日までである。7月は**31**日までである。

※うるう年のことは考えないことにする

①trinketの次のページを開く

<https://trinket.io/python/88a728c3cb>

② 実行する。 **30, 31**が表示されることを確認



```
main.py
1 days = [0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
2 print(days[6])
3 print(days[7])
4
```

Powered by
30
31

③8月や9月について表示できるように、プログラムを変更し実行してみる

trinket Run Modules

```
main.py  
1 days = [0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]  
2 print(days[6])  
3 print(days[7])  
4
```

Powered by
30
31

リストの
組み立て

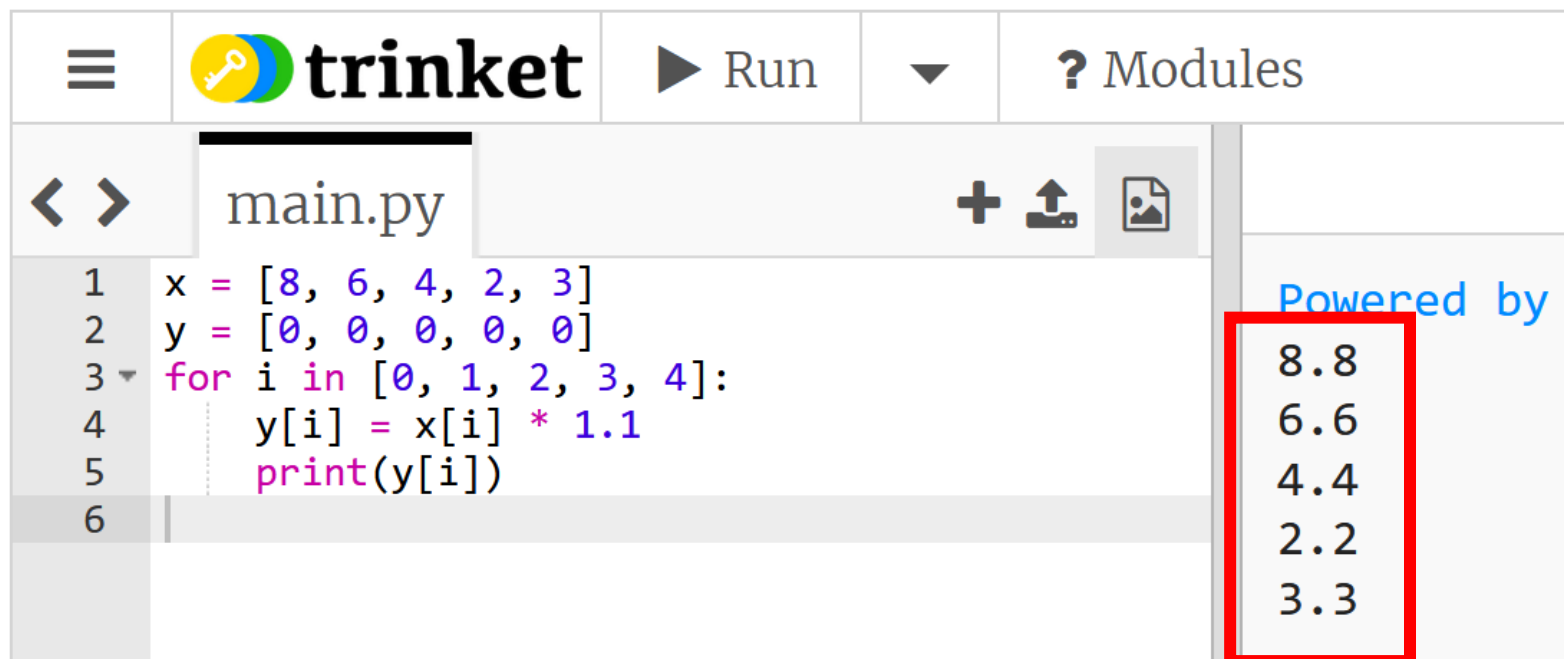
6番の要素, 7番の要素の表示

2. 計算の繰り返し

① trinketの次のページを開く

<https://trinket.io/python/cc2c13d793>

② 実行する。結果が5つ表示されることを確認



```
main.py
1 x = [8, 6, 4, 2, 3]
2 y = [0, 0, 0, 0, 0]
3 for i in [0, 1, 2, 3, 4]:
4     y[i] = x[i] * 1.1
5     print(y[i])
6
```

Powered by

8.8
6.6
4.4
2.2
3.3

リストの 組み立て

main.py

```
1 x = [8, 6, 4, 2, 3]
2 y = [0, 0, 0, 0, 0]
3 for i in [0, 1, 2, 3, 4]:
4     y[i] = x[i] * 1.1
5     print(y[i])
6
```

Powerrec

8.8
6.6
4.4
2.2
3.3

「 $y[i] = x[i] * 1.1$ 」を
 i の値を変えながら
5回繰り返す

3. 重力と落下距離

物体を落とすと $9.8 \times (\text{時間})^2 \div 2$ の分, 落ちていく。
空気抵抗は無視する。

(この式は, 等加速度運動の式)


① trinketの次のページを開く

<https://trinket.io/python/e27702ef75>

② 実行する。結果が表示されることを確認

```
main.py
1 x = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
2 for t in x:
3     print((9.8 / 2) * t * t)
4
```

Rest

Powered by  trinket

0.0
4.9
19.6
44.1
78.4
122.5
176.4
240.1
313.6
396.9
490.0

4. リストの組み立て, 要素の挿入と削除



```
x = [15, 8, 6, 32 ,23]
print(x)
x.append(4)
print(x)
x.remove(8)
print(x)
```

① trinketの次のページを開く

<https://trinket.io/python/f33df42c8d>

② 実行する。結果が表示されることを確認

```
main.py
1 x = [15, 8, 6, 32 ,23]
2 print(x)
3 x.append(4)
4 print(x)
5 x.remove(8)
6 print(x)
```

Powered by trinket

```
[15, 8, 6, 32, 23]
[15, 8, 6, 32, 23, 4]
[15, 6, 32, 23, 4]
```

リストのプログラム例



≡  trinket ▶ Run ▼ ? Modules

< > main.py




Powered by
June

```
1 # リストの作成
2 months = ["", "January", "February", "March", "April", "May", "June",
3 "July", "August", "September", "October", "November", "December"]
4 print(months[6])
```

forによる繰り返し



```
main.py
1 for i in range(10):
2     print(i)
3
4
5
```

Powered by  tri

0
1
2
3
4
5
6
7
8
9

プログラム

実行結果

- Python) のforループは、指定された範囲に対して繰り返し処理を行う
- 「range(10)」は、数の並び0, 1, 2, 3, 4, 5, 6, 7, 8, 9の意味
- 「for i in range(10)」の場合、iの値は、0, 1, 2, 3, 4, 5, 6, 7, 8, 9と変化する。ループは、それぞれに対して1回ずつ実行される

演習 4 . 繰り返しでの変数値の変化

ページ43～47

【トピックス】


- forによる繰り返し
- foriin range(5)でのiの値の変化
- 確認クイズに自主的に挑戦
(アクティブラーニング)

① trinketの次のページを開く

<https://trinket.io/python/27f6ebe1da>

② 実行結果が、次のように表示されることを確認

```
1 for i in range(5):  
2     print(i * '&')  
3  
4  
5
```

Powered by 

&
&&
&&&
&&&&

Python) で「**i * '&'**」は、**&をi個**の意味になる

③ 確認クイズ

&を9個まで表示したい（下図のように）ときは、
どのようにプログラムを書き換えるか？

自分でチャレンジしてせよ

```
1 for i in range(5):  
2     print(i * '&')
```

このプログラムを書き換える

```
&  
&&  
&&&  
&&&&  
&&&&&  
&&&&&&  
&&&&&&&  
&&&&&&&&  
&&&&&&&&&  
&&&&&&&&&&
```

このような結果を得る

1 × 2 × 3 × 4 × 5 の計算



④ trinketの次のページを開く

<https://trinket.io/python/b7eb532453>

⑤ 実行する。 1 × 2 × 3 × 4 × 5 の計算が行われる。

1 2 0 が表示される

A screenshot of the Trinket Python IDE interface. The top bar contains a menu icon, a key icon, a play button, a dropdown arrow, and a question mark. Below this, the file name "main.py" is shown with navigation arrows, a plus sign, an upload icon, and a file icon. The code editor shows the following Python code:

```
1 s = 1
2 for i in range(5):
3     s = s * (i + 1)
4
5 print(s)
6
7
8
```

The right side of the interface is labeled "Result" and is currently empty. At the bottom of the interface, it says "Powered by trinket" with the Trinket logo and the number "120" below it.


⑥ 確認クイズ

1 × 2 × 3 × 4 × 5 × 6 × 7 × 8を表示したい（下図のように）ときは、どのようにプログラムを書き換えるか？

自分でチャレンジしてせよ

```
1 s = 1
2 for i in range(5):
3     s = s * (i + 1)
4
5 print(s)
```

このプログラムを書き換える

Powered by  **trinket**
40320

このような結果を得る

答え合わせ



③

```
main.py
1 for i in range(10):
2     print(i * '&')
3
```

Powered by t

&
&&
&&&
&&&&
&&&&&
&&&&&&
&&&&&&&
&&&&&&&&
&&&&&&&&&

⑥

```
1 s = 1
2 for i in range(8):
3     s = s * (i + 1)
4
5 print(s)
6
7
8
```

Powered by t

40320

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。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。
- Python) の**リスト**は、**複数の要素を一度に保持**できる。リストの要素は順序を持ち、**番号は0から始まる**。

```
months = ["", "January", "February", "March", "April",  
"May", "June", "July", "August", "September",  
"October", "November", "December"]  
print(months[6])
```
- **繰り返し**:繰り返し計算では、特定の計算を**何度も行う**。

今回の授業の学ぶ意義と満足感



- ① **変数, 条件分岐, リスト, 繰り返し**を理解. Python) 以外のプログラミング言語でも通用する知識の習得. 論理的思考力を向上. 達成感を獲得
- ② **オンラインツールtrinket**を活用した実践的なスキル. 自分でプログラムを書き換えて, 間違いがあったときは自分で対処する. 自主性向上, プログラミングスキル向上
- ③ コンピューターの動作原理やアプリケーションの仕組みを理解. **コンピュータとプログラミングの幅広い可能性**に気づく
- ④ **デジタル時代に有用なスキル**を獲得. 将来は, データ分析, アプリ開発, AI (人工知能) 等の先端技術学習への足がかりを獲得