プログラミング教室のテクノロ

Pythonの道

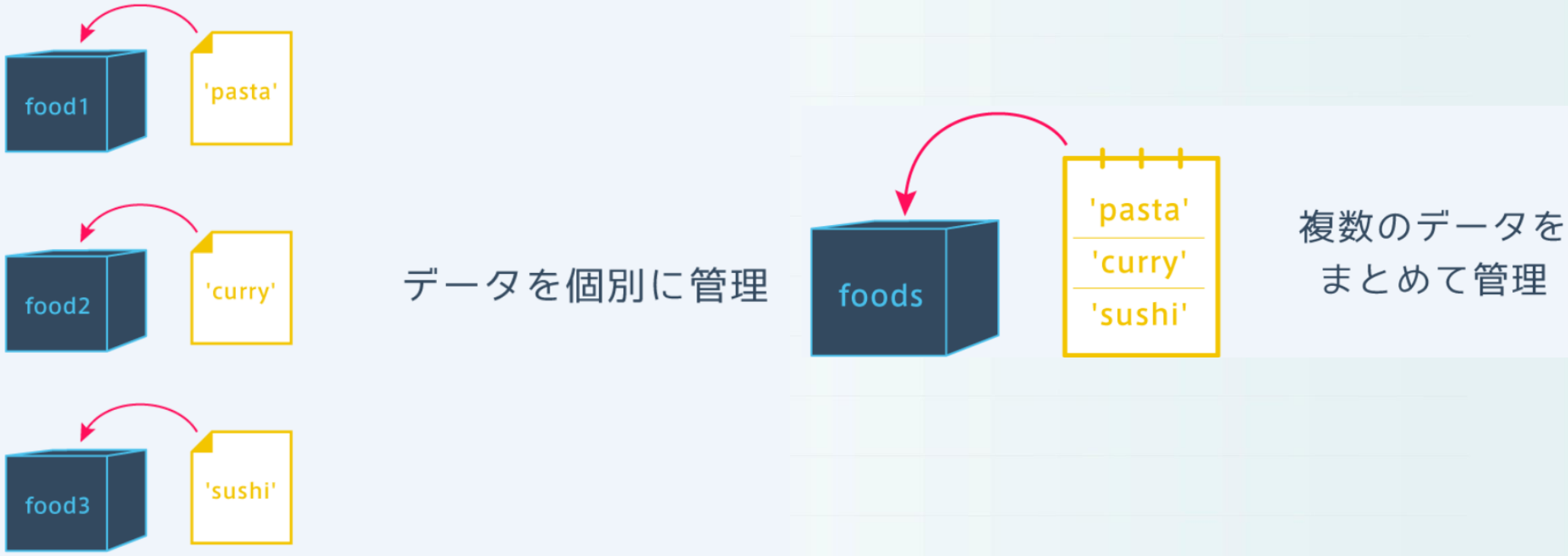
トレーニングドリル③

もくじ

・複数のデータを扱う（リスト、辞書）
・お買い物プログラムを作ろう

複数のデータを扱う

複数のデータをまとめて扱う方法を学ぼう。
例えば、食べ物の名前が複数ある時、それぞれを個別に変数として定義しておくより「食べ物の名前一覧」といったように、関連するデータをまとめて管理する方が便利。



リスト

複数のデータをまとめて管理するにはリストを使う。
リストは[要素1，要素2，...]のように作る。リストに入っているそれぞれの値のことを要素と呼ぶ。リストを使うと、複数の文字列や複数の数値を1つのものとして管理することができる。


```
＃文字列のリスト
[‘ pasta’ , ’ curry’ , ’ sushi’ ]

＃数値のリスト
[1, 2, 3, 5, 8, 13, 21]

＃文字列と数値のリスト
[‘ apple’ , ’ banana’ , 200, 300]
```

リストを変数に代入しよう

リストも1つの値なので変数に代入することができる。
このとき、リストを代入する変数名は慣習上複数形にすることが多いので、覚えておこう。

```
foods = [‘ pasta’ , ‘ curry’ , ‘ sushi’ ]
print(foods)
```

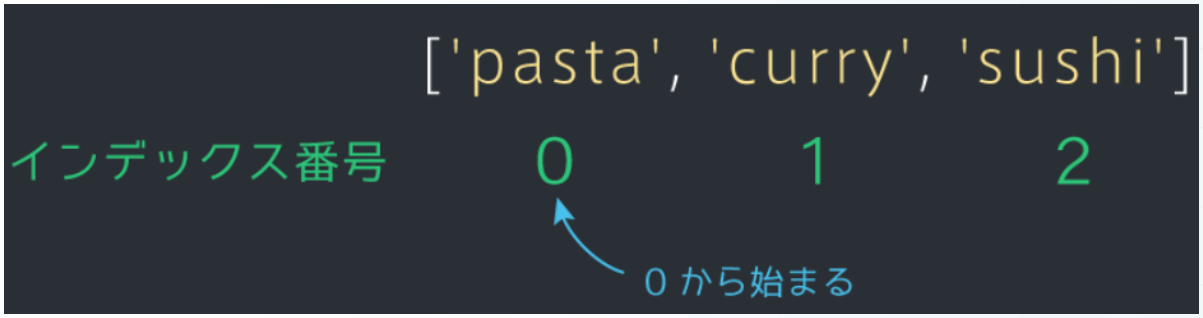
変数名

```
['pasta', 'curry', 'sushi']
```

リストがそのまま出力される

リストの要素を取得しよう

リストの要素には、前から順に「0, 1, 2,・・・」と数字が割り振られている。これをインデックス番号という。インデックス番号は0から始まる点に注意しよう。
リストの各要素は、リスト[インデックス番号]とすることで取得することができる。




```
foods = [' pasta', ' curry', ' sushi' ]
print(' 好きな食べ物は' + foods[2] + ' です' )
```

インデックス番号を用いて値を取り出す

```
好きな食べ物は sushi です
```

インデックス番号が 2 の要素が出力されている

練習

問題を読んでコードを書いてみよう。

【問題】

変数fruitsに、複数の文字列を要素に持つリストを代入してください

インデックス番号が0の要素を出力してください

インデックス番号が2の要素を文字列と連結して出力してください

練習

【答え】

```
# 変数fruitsに、複数の文字列を要素に持つリストを代入してください
fruits = ["apple","banana","orange"]

# インデックス番号が0の要素を出力してください
print(fruits[0])

# インデックス番号が2の要素を文字列と連結して出力してください
print("好きな果物は"+fruits[2]+"です")
```

リストの要素を更新しよう

リストの要素を更新してみよう。
「リスト[インデックス番号] = 値」とすることで、リストの指定したインデックス番号の要素を更新することができる。

```
foods = [' pasta' , ' curry' , ' sushi' ]
foods[1] = ' pizza'
print(foods)
```

インデックス番号1の要素を更新

```
['pasta', 'pizza', 'sushi']
```

要素が更新されている

リストに要素を追加しよう

リストには、新しく要素を追加することもできる。
「リスト.append(値)」とすることで、すでに定義されているリストの末尾に新たな要素を追加することができる。

```
foods = [‘ pasta’ , ’ curry’ , ’ sushi’ ]
foods.append( ‘ pizza’ )

print(foods)
```

リストの末尾に要素を追加

```
['pasta', 'curry', 'sushi', 'pizza']
```

要素が追加されている

練習

問題を読んでコードを書いてみよう。

【問題】

```
fruits = ['apple', 'banana', 'orange' ]
# リストの末尾に文字列「grape」を追加してください

# 変数fruitsに代入したリストを出力してください

# インデックス番号が0の要素を文字列「cherry」に更新してください

# インデックス番号が0の要素を出力してください
```

練習

【答え】

```
fruits = ['apple', 'banana', 'orange']

# リストの末尾に文字列「grape」を追加してください
fruits.append('grape')

# 変数fruitsに代入したリストを出力してください
print(fruits)

# インデックス番号が0の要素を文字列「cherry」に更新してください
fruits[0] = 'cherry'

# インデックス番号が0の要素を出力してください
print(fruits[0])
```

リストの要素を全て取得しよう

リストの要素を全て出力したいような時、次のように書くのは非常に面倒だ。for文というものを使うと、これを簡単にできるようになる。

```
foods = [‘ pasta’ , ’ curry’ , ’ sushi’ ]
print(‘ 好きな食べ物は’ + foods[0] + ‘ です’ )
print(‘ 好きな食べ物は’ + foods[1] + ‘ です’ )
print(‘ 好きな食べ物は’ + foods[2] + ‘ です’ )
```

要素を１つずつ出力するのは面倒。。。。。

```
好きな食べ物は pasta です
好きな食べ物は curry です
好きな食べ物は sushi です
```

for文の書き方

for文を使うと、リストの要素を順に取り出して処理を行うことができる。次のように「for 変数名 in リスト:」と書くことで、リストの要素の数だけ、処理を繰り返すことができる。

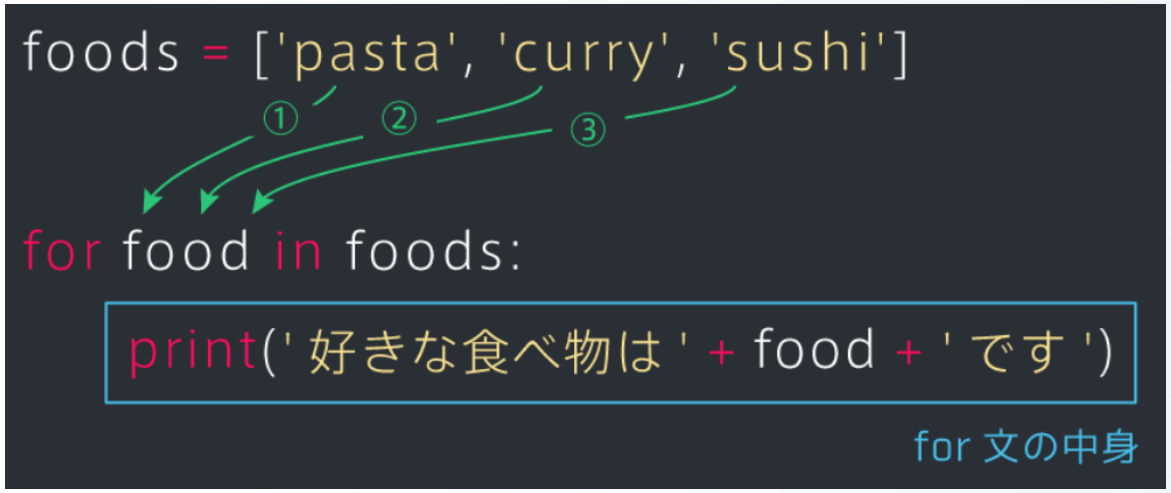
```
foods = ['pasta', 'curry', 'sushi']
for food in foods:
    print('好きな食べ物は' + food + 'です')
```

変数名　リスト　行末にコロン！　インデント！

```
好きな食べ物は pasta です
好きな食べ物は curry です
好きな食べ物は sushi です
```

for文の処理の流れ

「for 変数名 in リスト:」とすると変数には、リストの要素が先頭から順に1つずつ入っていき、その上でfor文の中の処理が実行される。処理はリストの要素の数だけ繰り返し行われる（繰り返し処理）。変数名は自由だが、リスト名の単数形にすることが多い。



① 変数 food に 'pasta' が代入される
for 文の中身の処理が実行される

② 変数 food に 'curry' が代入される
for 文の中身の処理が実行される

③ 変数 food に 'sushi' が代入される
for 文の中身の処理が実行される

練習

問題を読んでコードを書いてみよう。

【問題】

```
fruits = ['apple', 'banana', 'orange']

# for文を用いてリストの要素を1つずつ取り出し、「好きな果物は○○です」と出力してください
```

練習

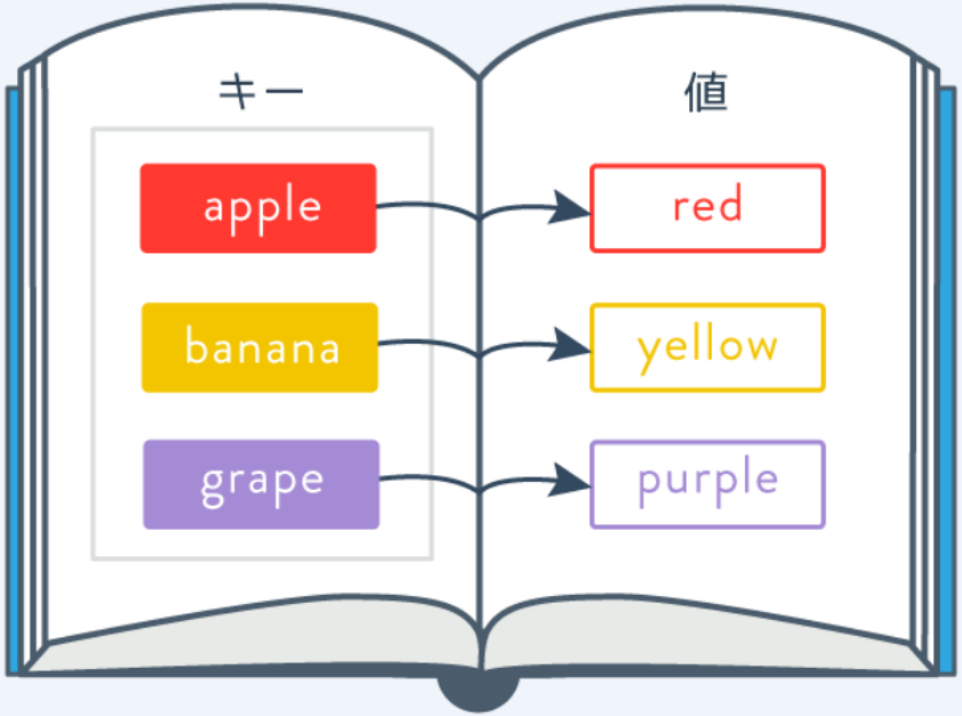
【答え】

```
fruits = ['apple', 'banana', 'orange']

# for文を用いてリストの要素を1つずつ取り出し、「好きな果物は○○です」と出力してください

for fruit in fruits:
    print('好きな果物は' + fruit + 'です')
```

辞書とは

辞書は、リストと同じように複数のデータをまとめて管理するのに用いられる。リストとの違いは、個々の要素をインデックス番号ではなくキーと呼ばれる名前を付けて管理する点にある。辞書ではキーと値のペアが1つの要素となる。

辞書のイメージ



辞書の作り方

辞書は{キー1: 値1, キー2: 値2, …}のように作る。
ほとんどの場合、キーには文字列が使われる。
リストは要素を[]で囲むが、辞書は{}で囲む。キーと値の間はコロン（:）、要素同士の間はコンマ（,）で区切る。

```
fruits = {'apple':'red','banana':'yellow','grape':'purple'}
```

変数名　キー　値　コンマで要素を区切る

辞書の要素の取り出し方

辞書の値を取り出すには、取り出したい値に対応する「キー」を用いて辞書名[キー]のように書く。

```
fruits = {‘ apple’ :’ red’ , ‘ banana’ :’ yellow’ , ‘ grape’ :’ purple’ }

print(‘ appleの色は’  + fruits[‘ apple’ ] + ‘ です’ )
```

キー名を用いて値を取り出す

```
appleの色は red です
```

練習

問題を読んでコードを書いてみよう。

【問題】

変数fruitsに辞書を代入してください

辞書fruitsのキー「banana」に対応する値を出力してください

辞書fruitsを用いて、「appleは○○という意味です」となるように出力してください

練習

【答え】

```
# 変数fruitsに辞書を代入してください
fruits = {'apple': 'りんご', 'banana': 'バナナ'}

# 辞書fruitsのキー「banana」に対応する値を出力してください
print(fruits['banana'])

# 辞書fruitsを用いて、「appleは○○という意味です」となるように出力してください
print('appleは' + fruits['apple'] + 'という意味です')
```

辞書の要素を更新しよう

辞書はリストと同じように要素の更新と追加をすることができる。まずは要素の変更をしてみよう。
辞書名[キー名] = 値と書くことで要素の更新をすることができる。

```
fruits = {'apple':'red','banana':'yellow','grape':'purple'}

fruits['apple'] = 'green'
print('appleの色は' + fruits['apple'] + 'です')
```

←キー名がappleである要素の値を更新

キー名を指定

```
appleの色はgreenです
```

値が更新されている

辞書に要素を追加しよう

次に要素の追加をしてみよう。
図のように「辞書名[新しいキー名] = 値」と書くことで辞書に新しい要素を追加することができる。ただし、辞書にすでにあるキー名を指定すると、値の追加ではなく更新になってしまうので注意する。

```
fruits = {'apple':'red','banana':'yellow'}

fruits['peach'] = 'pink'   ←キー名がpeachである要素の値を追加

print(fruits)
```

キー名を指定

```
{'peach': 'pink', 'banana': 'yellow', 'apple': 'red'}
```

要素が追加されている

練習

問題を読んでコードを書いてみよう。

【問題】

```
fruits = {'apple': 100, 'banana': 200, 'orange': 400}

# キー「banana」の値を数値「300」に更新してください

# キーが「grape」、値が数値の「500」の要素を辞書fruitsに追加してください

# fruitsの値を出力してください
```

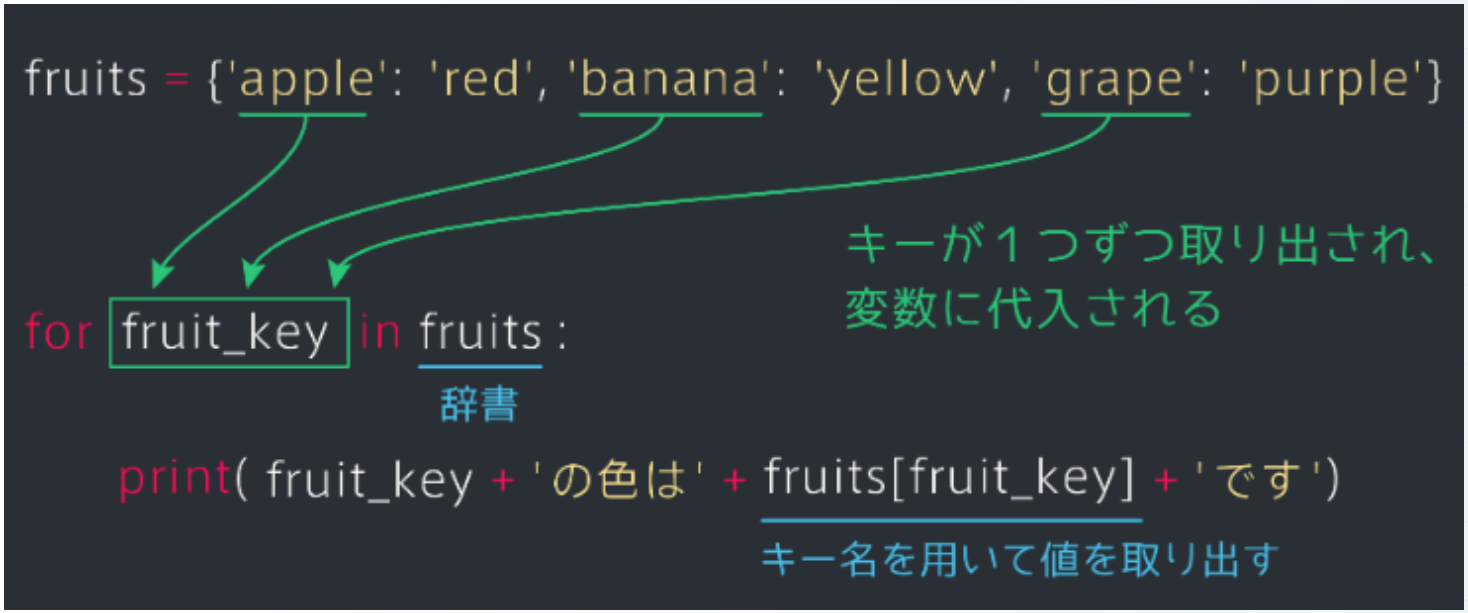
練習

【答え】

```
fruits = {'apple': 100, 'banana': 200, 'orange': 400}

# キー「banana」の値を数値「300」に更新してください
fruits['banana'] = 300

# キーが「grape」、値が数値の「500」の要素を辞書fruitsに追加してください
fruits['grape'] = 500

# fruitsの値を出力してください
print(fruits)
```

辞書の要素を全て取得しよう

リストと同じように、辞書もfor文を用いて要素を1つずつ取り出し、処理を行うことができる。「for 変数名 in 辞書:」と書くことで繰り返し処理をすることができる。ここで定義された変数にそれぞれの要素のキーが1つずつ代入される。要素の値は図のようにキーが代入された変数を用いて取り出せる。




```
appleの色はredです
bananaの色はyellowです
grapeの色はpurpleです
```

練習

問題を読んでコードを書いてみよう。

【問題】

```
fruits = {'apple': 'りんご', 'banana': 'バナナ', 'grape': 'ぶどう'}

# for文を用いて、辞書のキーを1つずつ取り出し、繰り返しの中で「〇〇は△△という意味です」と出力させてください
```

練習

【答え】

```
fruits = {'apple': 'りんご', 'banana': 'バナナ', 'grape': 'ぶどう'}

# for文を用いて、辞書のキーを1つずつ取り出し、繰り返しの中で「〇〇は△△という意味です」と出力させてください

for fruit_key in fruits:
    print(fruit_key + 'は' + fruits[fruit_key] + 'という意味です')
```

while文

繰り返し処理にはfor文以外にもwhile文というものがある。
while文を用いると、「ある条件に当てはまる間、処理を繰り返す」といったことが可能になる。

1～100までの数字を出力する例

```
X = 1
while X <= 100:
    print(X)
    X = X + 1
```

```
1
2
3
4
:
99
100
```

練習

問題を読んでコードを書いてみよう。

【問題】

```
x ＝ 10

# while文を用いて、「変数xが0より大きい」間、繰り返される繰り返し処理を作ってください

    # 変数xを出力してください

    # 変数xから1引いてください
```

練習

【答え】

```
x = 10

# while文を用いて、「変数xが0より大きい」間、繰り返される繰り返し処理を作ってください

while x > 0:
    # 変数xを出力してください
    print(x)
    # 変数xから1引いてください
    x = x-1
```

break

ここでは繰り返し処理を繰り返しの途中で強制的に終了するための方法を学ぼう。
breakを用いると繰り返し処理を終了することができる。if文などの条件分岐と組み合わせて使う。while文でも同じように使うことができることを知っておこう。

```
numbers = [1,2,3,4,5,6]
for number in numbers:
    print(number)
    if number == 3:
        break
```

```
1
2
3
>>>
```

練習

問題を読んでコードを書いてみよう。

【問題】

```
numbers = [765, 921, 777, 256]
for number in numbers:
    print(number)

    # 変数numberが777のとき「777が見つかったので処理を
      終了します」と出力した後、処理を終了させてください
```

練習

【答え】

```
numbers = [765, 921, 777, 256]
for number in numbers:
    print(number)

    # 変数numberが777のとき「777が見つかったので処理を
      終了します」と出力した後、処理を終了させてください

    if number == 777:
        print('777が見つかったので処理を終了します')
        break
```

continue

繰り返し処理を終了するbreakと違い、continueはその周の処理だけをスキップすることができる。continueもif文などと組み合わせて利用する。while文でも同じように使うことができる。

```
numbers = [1,2,3,4,5,6]
for number in numbers:
    if number %2 == 0:
        continue
    print(number)
```

```
1
3
5
>>>
```

練習

問題を読んでコードを書いてみよう。

【問題】

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
for number in numbers:
    # 変数numberの値が3の倍数のとき、繰り返し処理をスキップしてください

    print(number)
```

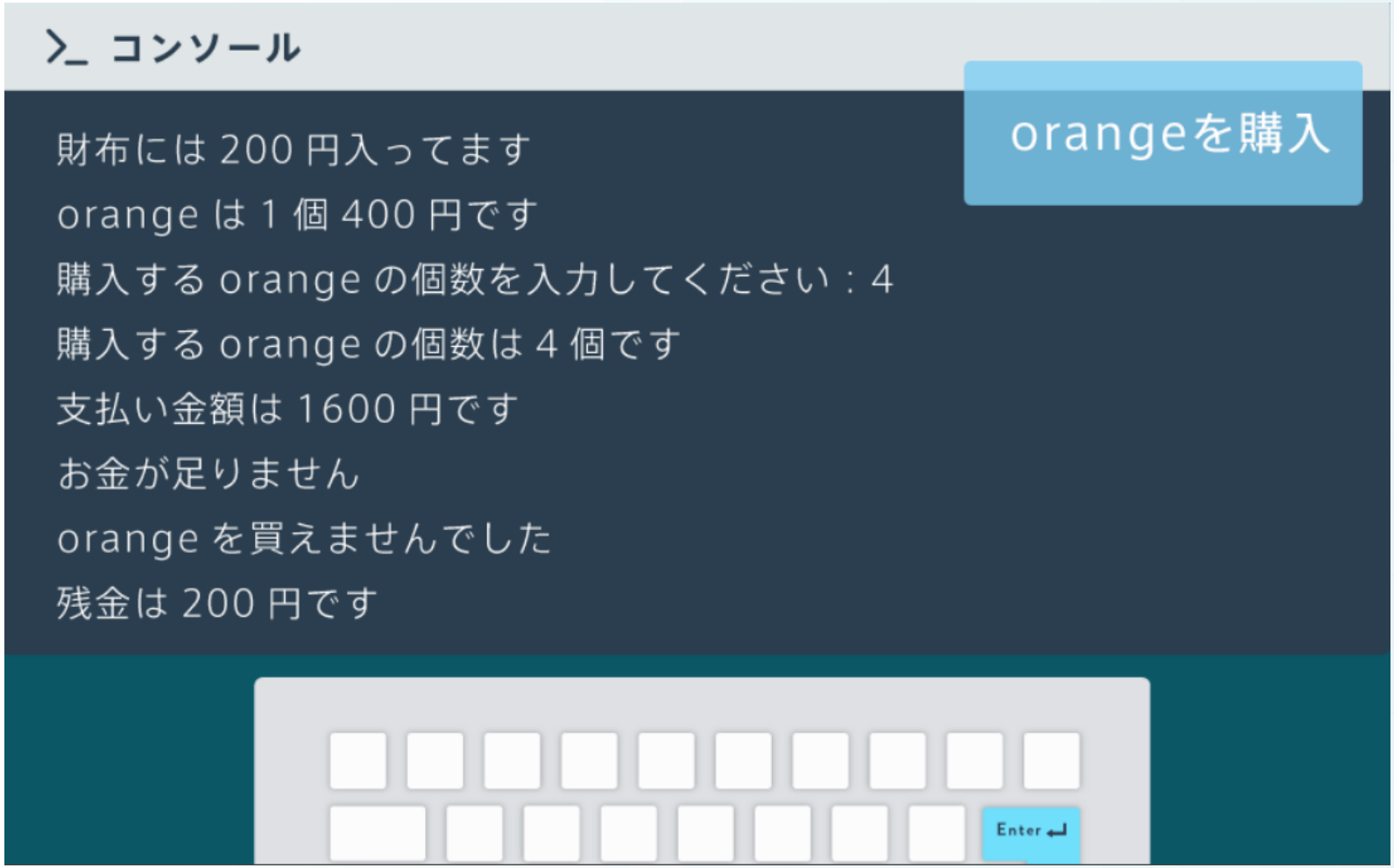
練習

【答え】

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
for number in numbers:
    # 変数numberの値が3の倍数のとき、繰り返し処理をスキップしてください

    if number % 3 == 0:
        continue
    print(number)
```

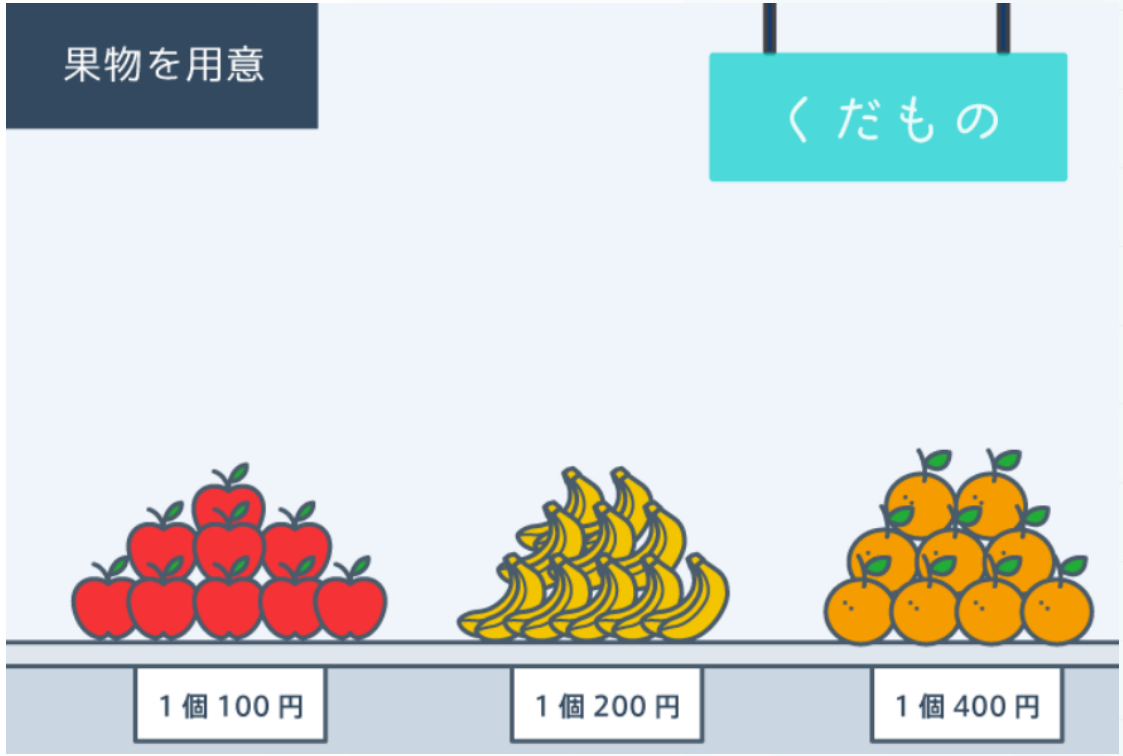
お買い物プログラムを作ろう

いままで学んだことを組み合わせて、お買い物代金を計算するプログラムを作ろう。コンソールに果物を購入する個数を入力し、個数に応じた処理内容を出力していく。



商品を用意しよう

まずは購入することができる果物とそれらの値段が要素として入った辞書を用意し、繰り返し処理を用いてそれぞれの値段を出力してみよう。




```
apple は 1 個 100 円です
banana は 1 個 200 円です
orange は 1 個 400 円です
```

練習

問題を読んでコードを書いてみよう。

【問題】

文字列のキーと数値の値を持つ辞書を作って、変数itemsに代入してください

for文を用いて、辞書itemsのキーを1つずつ取り出していく繰り返し処理を作成してください

「----------」を出力してください

「○○は1個△△円です」となるように出力してください

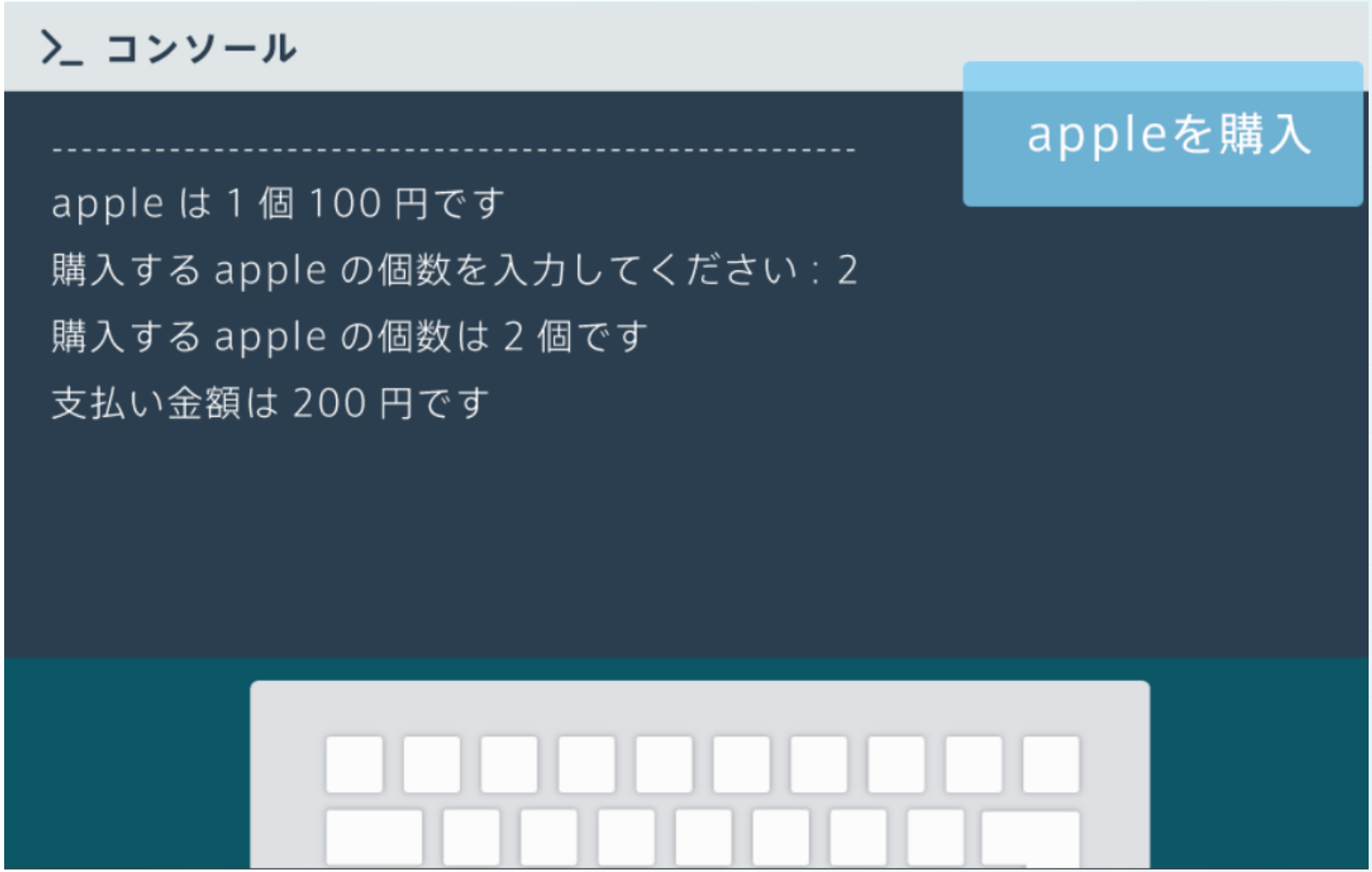
練習

【答え】

```
# 文字列のキーと数値の値を持つ辞書を作って、変数itemsに代入してください
items = {'apple': 100, 'banana': 200, 'orange': 400}

# for文を用いて、辞書itemsのキーを1つずつ取り出していく繰り返し処理を作成してください
for item_name in items:
    # 「----------」を出力してください
    print('----------')
    # 「○○は1個△△円です」となるように出力してください
    print(item_name + 'は1個' + str(items[item_name]) + '円です')
```

商品を購入しよう

繰り返し処理を用いて果物を順番に購入できるようにする。
購入する個数の入力は、inputを用いる。入力した個数と、支払い金額を順に出力してみよう。



練習

問題を読んでコードを書いてみよう。

【問題】

```
items = {'apple': 100, 'banana': 200, 'orange': 400}
for item_name in items:
    print('-----------------------------------------------')
    print(item_name + 'は1個' + str(items[item_name]) + '円です')

    # inputを用いて入力を受け取り、変数input_countに代入
      してください


    # キーと変数input_countを用いて「購入する○○の個数
      は△△個です」となるように出力してください


    # input_countを数値として変数countに代入してください
```

練習

変数total_priceに果物1個の値段と変数countを掛けた値を代入してください

変数total_priceと型変換を用いて、「支払い金額は〇〇円です」となるように出力してください

練習

【答え】

```
items = {'apple': 100, 'banana': 200, 'orange': 400}
for item_name in items:
    print('-------------------------------------------------')
    print(item_name + 'は1個' + str(items[item_name]) + '円です')

    # inputを用いて入力を受け取り、変数input_countに代入
      してください
    input_count = input('購入する' + item_name + 'の個数を入力してください：')

    # キーと変数input_countを用いて、「購入する○○の個数は△△個です」とな
      るように出力してください
    print('購入する' + item_name + 'の個数は' + input_count + '個です' )

    # input_countを数値として変数countに代入してください
    count = int(input_count)

    # 変数total_priceに果物1個の値段と変数countを掛けた値を代入してください
    total_price = items[item_name] * count
    # 変数total_priceと型変換を用いて、「支払い金額は○○円です」となるよう
    に出力してください
    print('支払い金額は' + str(total_price) + '円です')
```

条件分岐をしよう

果物を購入できるようになったので次は、所持金と購入代金を用いて条件分岐を作っていこう。

練習

【問題】

```
# 変数moneyに数値1000を代入してください

items = {'apple': 100, 'banana': 200, 'orange': 400}
for item_name in items:
    print('-------------------------------------------------')
    # 変数moneyを用いて「財布には○○円入っています」の
      ように出力してください

    print(item_name + 'は1個' + str(items[item_name]) + '円です' )

    input_count = input('購入する' + item_name + 'の個数を入力してください：' )

    print('購入する' + item_name + 'の個数は' + input_count + '個です')

    count = int(input_count)

    total_price = items[item_name] * count

    print('支払い金額は' + str(total_price) + '円です')

    # moneyとtotal_priceの比較結果によって条件を分岐してください
```

練習

【答え】

```
# 変数moneyに数値1000を代入してください
money = 1000

items = {'apple': 100, 'banana': 200, 'orange': 400}
for item_name in items:
    print('----------------------------------------------------')
    # 変数moneyを用いて「財布には○○円入っています」のように出力してください

    print('財布には' + str(money) + '円入っています')
    print(item_name + 'は1個' + str(items[item_name]) + '円です')

    input_count = input('購入する' + item_name + 'の個数を入力してください：' )
    print('購入する' + item_name + 'の個数は' + input_count + '個です')

    count = int(input_count)
    total_price = items[item_name] * count
    print('支払い金額は' + str(total_price) + '円です')

    # moneyとtotal_priceの比較結果によって条件を分岐してください
    if money >= total_price:
        print(item_name + 'を' + input_count + '個買いました')
        money = money - total_price
    else:
        print('お金が足りません')
        print(item_name + 'を買えませんでした')
```

残金を計算しよう

最後に、お買い物が終わったあとの処理を書いてみよう。
お買い物が終了したら残金を出力するようにする。また、途中で所持金がなくなったらお買い物（繰り返し処理）を終了させるようにしよう。

練習

【問題】

```
money = 1000
items = {'apple': 100, 'banana': 200, 'orange': 400}
for item_name in items:
    print('------------------------------------------------')
    print('財布には' + str(money) + '円入っています')
    print(item_name + 'は1個' + str(items[item_name]) + '円です')

    input_count = input('購入する' + item_name + 'の個数を入力してください：')
    print('購入する' + item_name + 'の個数は' + input_count + '個です')

    count = int(input_count)
    total_price = items[item_name] * count
    print('支払い金額は' + str(total_price) + '円です')

    if money >= total_price:
        print(item_name + 'を' + input_count + '個買いました')
        money = money - total_price
        # if文を用いて、moneyの値が0のときの条件を分岐してください

    else:
        print('お金が足りません')
        print(item_name + 'を買えませんでした')
# 変数moneyと型変換を用いて、「残金は○○円です」となるように出力してください
```

練習

【答え】

```
money = 1000
items = {'apple': 100, 'banana': 200, 'orange': 400}
for item_name in items:
    print('-------------------------------------------------')
    print('財布には' + str(money) + '円入っています')
    print(item_name + 'は1個' + str(items[item_name]) + '円です')

    input_count = input('購入する' + item_name + 'の個数を入力してください：')
    print('購入する' + item_name + 'の個数は' + input_count + '個です')

    count = int(input_count)
    total_price = items[item_name] * count
    print('支払い金額は' + str(total_price) + '円です')

    if money >= total_price:
        print(item_name + 'を' + input_count + '個買いました')
        money = money - total_price
         # if文を用いて、moneyの値が0のときの条件を分岐してください
        if money == 0:
            print('財布が空になりました')
            break
    else:
        print('お金が足りません')
        print(item_name + 'を買えませんでした')
# 変数moneyと型変換を用いて、「残金は○○円です」となるように出力してください
print('残金は' + str(money) + '円です')
```

復習＆チャレンジ

ここまで習ったことをScratchでもできるかチャレンジしてみよう。
その過程でScratchでできること、Pythonでないとできないことを整理してみよう。
例えば、データの型という概念はscratchにはあっただろうか？

メモ

プログラミング教室のテクノロ

なまえ：