第1章 順次実行と変数 例題と演習問題の解答例

## 例題1

|  |
| --- |
| print("こんにちは")  print("はじめての")  print("プログラミング") |

### 【解説】

print()関数を３回使っています。「こんにちは」「はじめての」「プログラミング」の順に表示するために、print("こんにちは")、print("はじめての")、print("プログラミング")の順に書きます。

## 例題2

|  |
| --- |
| a = "Hello"  a = "こんにちは"  print(a) |

### 【解説】

「こんにちは」と表示されます。

変数aに最初は"Hello"を代入します。次に"こんにちは"と代入した後、変数aを表示するので、表示は「こんにちは」となります。変数には、最後に代入した値だけが残ることを確認しましょう。

※後の章で出てくるリストでは、一つのリストに複数の値を代入することができます。

## 例題3

|  |
| --- |
| a = 34567  b = 123456  c = a + b  print("34567 + 123456 =", c) |

### 【解説】

変数aに34567、変数bに123456を代入します。aとbの足し算の結果をcに代入します。最後に、計算式の文字列と、変数cを表示します。関数print()には、複数の引数を渡すことができます。複数の引数を渡す場合、引数と引数の間に記号『,』（カンマ）を入れて区切るようにします。

## 例題4

|  |
| --- |
| a = input("数字を入力してください")  print(a) |

### 【解説】

関数input()を使って、ユーザーに入力をうながします。「数字を入力してください」と表示するには、引数として関数input()に渡します。

ユーザーが入力した値は、関数input()が受け取り、戻り値としてプログラムに返します。上の回答例では、戻り値を変数aに代入しています。

最後に関数print()に変数aを渡して、表示します。

## 問題集1

|  |
| --- |
| print("おはよう")  print("こんにちは")  print("こんばんは") |

### 【解説】

print()関数を３回使っています。「おはよう」「こんにちは」「こんばんは」の順に表示するために、print("おはよう")、print("こんにちは")、print("こんばんは")の順に書きます。

## 問題集2

|  |
| --- |
| myouji = "山田"  namae = "太郎"  print( myouji, namae ) |

### 【解説】

変数myoujiと、変数namaeにそれぞれ値を代入します。上の例では変数myoujiに"山田"、変数namaeに"太郎"を代入しています。

関数print()に、引数として変数myouji、namaeを渡します。複数の引数を渡すときには、『,』で区切ります。

## 問題集3

|  |
| --- |
| a = int( input("数字を入力してください") )  print("a+10:", a+10 ) |

### 【解説】

関数input()にユーザーに数字を入力するよう促すメッセージを渡します。関数input()が返す値を、関数int()に渡し、関数int()の戻り値を変数aに代入します。

関数print()の引数のカッコの中で、a+10の計算を行い、結果を表示させています。

第2章 条件による選択（分岐） 例題と演習問題の解答例

## 例題1

|  |
| --- |
| a = int( input("0か1を入力してください"))  if a == 1:  print("1が入力されました")  else:  print("1以外が入力されました") |

### 【解説】

関数input()を使い、ユーザーに０か１を入力するように促します。関数int()を使い、入力された値を整数に変換して、変数aに代入します。

キーワードifの後ろで、変数aと1を比較します。演算子『==』を使うことで、左辺と右辺が等しいかどうか調べます。等しければ真（true）、等しくなければ偽（false）が返されます。比較の式の後ろに記号『:』（コロン）をつけることを忘れないように注意してください。

真（true）が返されたら「1が入力されました」と表示します。キーワードelseを使って、偽（false）が返されたときには「1以外が入力されました」と表示するようにします。

なお、値を代入するときの記号『=』は一つです。一方、比較するときは『==』と、二つになることに注意してください。

## 例題2

|  |
| --- |
| import random  a = random.randint(1,6)  print("サイコロ:", a) |

### 【解説】

乱数を生成するための機能randomをインポートします。

関数random.randint()に、引数として1,6を指定します。これで、１以上６以下で乱数が生成されます。生成された乱数は戻り値として返されるので、変数aに代入します。

print()で変数aを表示します。

## 問題集1

|  |
| --- |
| a = int( input("数字を入力してください"))  if a % 2 == 0:  print("これは偶数です")  else:  print("これは奇数です") |

### 【解説】

関数input()を使い、数字を入力するように促します。関数int()を使い、入力された値を整数に変換して、変数aに代入します。

キーワードifの後ろで、変数aを2で割ったときの余りと、1を比較します。演算子『%』を使うことで、２で割った余りを求めることができます。演算子『==』を使うことで、左辺と右辺が等しいかどうか調べます。等しければ真（true）、等しくなければ偽（false）が返されます。

## 問題集2

|  |
| --- |
| import random  a = random.randint(1,6)  if a == 1:  print("一が出ました")  elif a == 2:  print("二が出ました")  elif a == 3:  print("三が出ました")  elif a == 4:  print("四が出ました")  elif a == 5:  print("五が出ました")  elif a == 6:  print("六が出ました") |

### 【解説】

乱数を生成するための機能randomをインポートします。

関数random.randint()に、引数として1,6を指定します。これで、１以上６以下で乱数が生成されます。生成された乱数は戻り値として返されるので、変数aに代入します。

キーワードifを使い、変数aと1が等しいか比較します。等しく無い場合、次のキーワードelifの比較が実行されます。２と等しいか、３と等しいか、…と、6までと等しいか比べます。

等しい値が見つかったら、関数print()を使って漢数字を表示します。

第3章 リスト 例題と演習問題の解答例

## 例題1

|  |
| --- |
| a = ["A","B","C","D"]  print( a[3] ) |

### 【解説】

“D”が表示されます。

リストに四つの要素を設定し、添え字（インデックス）を指定して、要素を表示します。添え字は０から始まることに注意しましょう。0番目が“A”、1番目が“B”、2番目が“C”、3番目が“D”です。

## 例題2

|  |
| --- |
| a = ["A","B","C"]  print( a ) |

【解説】

関数print()にリストaを渡すと、全ての要素を表示します。

リストの要素を指定するための記号『 [ ] 』や、添え字を指定する必要はありません。

## 例題3

|  |
| --- |
| a = ["A","B","C"]  a[0] = "a"  print( a ) |

### 【解説】

“a”、“B”、“C”と表示されます。先頭の要素（０番目の要素）が小文字の“a”になっていることを確認しましょう。

プログラムは、リストaに３つの要素を設定した後、添え字に０を指定して、値を代入しています。

## 例題4

|  |
| --- |
| g = []  g.append("おはよう")  g.append("こんにちは")  g.append("こんばんは")  print( g ) |

### 【解説】

空のリストgを用意しています。g.append()を3回呼び出して、「おはよう」「こんにちは」「こんばんは」をリストに追加しています。最後のprint( g ) で、g.append()が呼び出された順に要素が追加されていることが確認できます。

## 例題5

|  |
| --- |
| aisatsuList = []  aisatsuList.append("おはよう")  aisatsuList.append("こんにちは")  aisatsuList.append("こんばんは")  aisatsuList.pop(1)  print( aisatsuList ) |

### 【解説】

「おはよう」「こんばんは」と表示されます。

プログラムの冒頭では、空のリストaisatsuListを用意しています。aisatsuList.append()を3回呼び出して、「おはよう」「こんにちは」「こんばんは」をリストに追加しています。続いて、aisatsuList.pop(1)を実行し、1番目の要素を取り除きます。

最後のprint( g ) で、aisatsuList.append()が呼び出された順に要素が追加されたこと、（0番目でなく）1番目の要素「こんにちは」が削除されていることが確認できます。

## 問題集1

|  |
| --- |
| a = ["A","B"]  print( "a[1]:", a[1] )  print()  a.append("C")  print( "a[1]:", a[1] )  print( "a[2]:", a[2] )  print()  a.insert( 2, "b" )  print( "a[1]:", a[1] )  print( "a[2]:", a[2] )  print()  b = a.pop( 2 )  print( "a[1]:", a[1] )  print( "a[2]:", a[2] )  print( "b:", b ) |

### 【解説】

最初はa[1]: Bと表示されます。

続いて、a[1]: B、a[2]: C と表示されます。添え字は０から始まることと、“C”が追加されたことを確認しましょう。

次のa.insert(2, "b")は、“C”があった場所に“b”を挿入します。“C”は後ろにずれます。a[1]: B、a[2]: bと表示されます。

次のa.pop( 2 )で、“b”は削除され、前に詰められます。表示はa[1]: B、a[2]: Cに戻ります。

a.pop(2)は削除した値を戻り値として返すので、変数bには“b”が代入されます。これにより、最後のprint("b:", b)で “b”が表示されるわけです。

## 問題集2

|  |
| --- |
| b = []  b.append( 1 )  b.append( 2 )  b.append( 3 )  b.append( 4 )  b.append( 5 )  print( b.pop(0) )  print( b.pop(0) )  print( b.pop(0) )  print( b.pop(0) )  print( b.pop(0) )  print( b ) |

### 【解説】

空のリストbを用意し、b.append()を5回実行して、１から５まで要素を追加しています。続いて、b.pop( 0 )で要素を取り出し、取り出した要素をprint()で表示しています。b.pop( 0 )は、先頭の要素を削除しながら取り出すため、リストの要素は１つずつ前に詰められます。b.pop( 0 )を5回実行すると、リストから全ての要素が削除されることになります。空のリストbをprint()に渡すと、空をあらわす表示になります。

第4章 繰り返し 例題と問題集の解答例

## 例題1

|  |
| --- |
| h = [1,2]  for i in h:  print(i)  print("おわり") |

### 【解説】

１、２と表示した後、最後に「おわり」と表示します。元のプログラムでは繰り返しごとに「おわり」と表示していましたが、繰り返しが終わった後に表示することを確認してください。

## 例題2

|  |
| --- |
| h = [1,2,3,4]  for i in h:  print(i) |

【解説】

リストを用意し、要素として４つ（1,2,3,4）を代入します。

続いて、キーワードfor ... in ...を使い、繰り返しを行います。リストhの先頭から要素を取り出し、変数iに代入し、関数print()でiを表示します。

リストの全ての要素について繰り返すことで、１、２、３、４と順に表示できます。

## 例題3

|  |
| --- |
| h = [4,3,2,1]  for i in h:  print(i) |

### 【解説】

リストを用意し、要素として４つ（4,3,2,1）を代入します。

続いて、キーワードfor ... in ...を使い、繰り返しを行います。リストhの先頭から要素を取り出し、変数iに代入し、関数print()でiを表示します。

リストの全ての要素について繰り返すことで、４、３、２、１と順に表示できます。

## 例題4

|  |
| --- |
| h = range(15,36)  for i in h:  print(i) |

### 【解説】

関数range()を使い、15から35の範囲を表す値を作ります。その値（※上の例では変数hに代入してあります）をキーワードfor のinに渡します。リストの要素全てについて、「変数iに代入し、表示する」という処理を繰り返すことになるので、15から順に35まで表示するプログラムになります。

## 例題5

|  |
| --- |
| h = range(1,50,2)  for i in h:  print(i) |

### 【解説】

１からはじまり、２ずつ増やしながら、４９まで表示します。

プログラムの最初で、関数range()に３つの引数を渡し、範囲の値を作っています。最初の引数（例題では１）から、2番目の引数（例題では５０）までの範囲です。3番目の引数は、最初の値から「いくつずつ増やしていくか」を指定するために使っています。つまり、最初の１行目で、「１、３、５、７、…、４７、４９」という範囲を作っているのです。

この範囲と、キーワードforを用いて、繰り返し表示をしています。

## 例題6

|  |
| --- |
| h = range(100,1,-10)  for i in h:  print(i) |

### 【解説】

１００から、９０、８０…と１０ずつ減らしながら、１０まで表示します。

プログラムの最初で、関数range()に３つの引数を渡し、範囲の値を作っています。関数range()の最初の引数（例題では１００）から、2番目の引数（例題では１）まで、3番目の引数で指定した値ずつ変化させながら（例題ではマイナス１０なので、１０ずつ減らしながら）、範囲を作ります。

この範囲をキーワードfor ... in に渡して繰り返すことで、１００から１０までの値を表示します。

## 例題7

|  |
| --- |
| import random  a = random.randint(1,10)  count = 1  while a >= count:  print("a:", a)  count = count + 1 |

### 【解説】

最初に、乱数を生成する機能randomをインポートします。関数random.randint(1,10)で、１以上１０以下の乱数を生成します。生成した値は変数aに代入します。

次の行の、変数countは、繰り返し回数を記録するための変数です。１から数え始めたいので、１を代入しています。

キーワードwhileを使い、繰り返し処理を作ります。条件判定の式では、変数aと変数countの値を比較します。変数aの値が変数count以上かを調べています。

繰り返し条件を満たす間、関数print()を呼び出し、変数aの値を表示します。変数countに１を足した値を、変数countに代入して処理は終わりです。またキーワードwhileの行まで戻り、繰り返しを続けるかどうか判定を行います。変数aの値よりも、変数countの方が大きくなったら終了です。

## 例題8

|  |
| --- |
| import random  a = random.randint(1,10)  count = range( 0, a )  for i in count:  print("a:", a) |

### 【解説】

最初に、乱数を生成する機能randomをインポートします。関数random.randint(1,10)で、１以上１０以下の乱数を生成します。生成した値は変数aに代入します。

関数range()を使い、０から始めて変数aの値までの範囲（変数aの値が4だったら、0、１、２、３の４つの要素を持つ範囲）を作ります。これを使ってfor ... in の繰り返しを行わせることで、生成した乱数の回数だけ繰り返す処理を作れます。

## 問題集1

|  |
| --- |
| for i in range(11,21):  print(i) |

### 【解説】

関数range()に11、21と引数を渡し、範囲を作成します。その範囲の値から一つずつ変数iに代入し、関数print()で表示しています。11から20まで表示するために、関数range(11,21)と、二つ目の引数が21になっていることに注意してください。

## 問題集2

|  |
| --- |
| for i in range(1,101):  if i%3 == 0:  print(i) |

### 【解説】

関数range()に1、101と引数を渡し、範囲を作成します。その範囲の値から一つずつ変数iに代入します。if文のブロックでは、余りを求める演算子%を使って、３で割った余りを求め、余りが０と等しいか調べます。３の倍数は３で割った余りが０になるので、この条件判定の式が真（true）を返したら、関数print()を実行して、値を表示します。

|  |
| --- |
| for i in range(3,101,3):  print(i) |

関数range()の引数をrange(3,101,3)とすると、３からはじまり、３ずつ増やしながら、１０１より小さい範囲を作ります。３の倍数だけ含む範囲なので、if文の条件判定無しに、関数print()を呼び出すだけで３の倍数を表示できます。

## 問題集3

|  |
| --- |
| import random  a = random.randint(1,6)  b = 0  while a > 0:  b = b + a  a = a - 1  print(b) |

### 【解説】

最初に、乱数を生成する機能randomをインポートします。関数random.randint(1,6)で、１以上６以下の乱数を生成します。生成した値は変数aに代入します。

変数bに０を代入します。

キーワードwhileに続けて、変数aが０より大きい間、繰り返すような条件の式を書いてます。

whileの繰り返しの中で、b=b+a、a=a-1の計算を行います。インデント（字下げ）のレベルを揃えていることに注意してください。

whileの繰り返しの後に変数bの値を表示しています。インデント（字下げ）のレベルがwhileと揃っていることを確認してください。

第5章 関数の定義と利用 例題と問題集の解答例

## 例題1

|  |
| --- |
| aisatsu = input("なにか言葉を入力してください:")  print( aisatsu ) |

### 【解説】

関数input()を使い、「なにか言葉を入力してください:」と表示し、入力を求めます。ユーザーが入力した値は、関数input()の戻り値として返され、変数aisatsuに代入されます。

その変数aisatsuを関数print()で表示します。

これでユーザーからの入力を受け取り、受け取ったまま表示することができました。

## 例題2

|  |
| --- |
| from datetime import datetime  hiduke\_jikoku = datetime.now() # 現在の日付・時刻を取得する  nen = hiduke\_jikoku.year  tsuki = hiduke\_jikoku.month  hi = hiduke\_jikoku.day  print( nen, "年", tsuki, "月", hi, "日") |

### 【解説】

最初に、datetimeをインポートします。

続いて、datetime.now()で現在（※プログラムが実行された時点の現在）の日付・時刻を取得します。上の例では、変数hiduke\_jikokuの中に年（year）・月（month）・日（day）の値が含まれています。それぞれを取り出し、最後に関数print()で表示しています。

## 例題3

|  |
| --- |
| def aisatsu():  print("こんにちは")  aisatsu()  aisatsu() |

### 【解説】

キーワードdefを使って、関数aisatsu()を定義します。関数の定義の中での処理を決めるため、字下げ（インデント）してprint()を書きます。

定義した関数aisatsu()を2回呼び出すと、2回「こんにちは」と表示されます。1回も呼び出さなければ、「こんにちは」と表示はされません。関数の定義の中にprint()と書いてあっても、関数が呼び出されなければ、print()は実行されません。

## 例題4

|  |
| --- |
| def shouhizei(kingaku):  zeigaku = kingaku \* 0.1  return zeigaku  print("消費税額:", shouhizei( 12000 ), "円" ) |

## 【解説】

キーワードdefを使って、関数shouhizei()を定義します。関数を呼び出すときに、引数として金額を渡すため、defの後ろの小かっこの中にkingakuと書いて、引数を定義します。

関数の中で、そのkingakuを使った計算を行い、returnで計算結果を呼び出し元に返すようにしています。

実際に呼び出すときは、関数名(引数の値）とします。上の例では、「shouhizei( 12000 )」が関数を呼び出している部分です。

## 問題集1

|  |
| --- |
| aisatsu = ["おはよう","こんにちは","こんばんは"]  import random  rand = random.randint(0,2)  print( aisatsu[rand] ) |

### 【解説】

リストaisatsuに、三つのあいさつの言葉を代入しています。

機能randomをインポートし、関数random.randint(0,2)を呼び出します。０、１、２のいずれかの値がランダムに返されます。その値を変数randに代入し、リストaisatsuの添え字に使って、値を表示します。

## 問題集2

|  |
| --- |
| aisatsu = ["おはよう","こんにちは","こんばんは","まいど","おやすみなさい"]  print( len(aisatsu) ) |

### 【解説】

リストaisatsuに、五つのあいさつの言葉を代入しています。

関数len()にリストを渡すと、リストの中の要素数が返されます。このプログラムでは、関数print()を呼び出す際の、引数のかっこの中で関数len()を使っています。

## 問題集3

|  |
| --- |
| def zeigaku\_keisan(kingaku, zeiritsu):  zeikin = kingaku \* zeiritsu / 100  return zeikin  kekka = zeigaku\_keisan(1980,10)  print("税金:", kekka) |

### 【解説】

キーワードdefを使って、関数zeigaku\_keisan()を定義します。この関数では、金額と税率の二つの引数を取ることにするので、小かっこの中に二つ（kingaku, zeiritsu）引数を書いています。カンマ（ , ）で区切っていることも確認してください。

計算した結果をキーワードreturnで返すように定義しています。

呼び出す際にはzeigaku\_keisan(1980,10)として、計算させたい値を二つ渡しています。値を渡すときにも、カンマで区切っています。

関数から返された値を変数kekkaに代入し、関数print() で表示させています。

第6章 繰り返し（反復）と選択（分岐）の組み合わせ　 例題と問題集の解答例

## 例題1

|  |
| --- |
| for i in range(1, 31):  if i % 3 == 0:  print(i, "は3の倍数です") |

### 【解説】

関数 range()を使い、1から30の範囲を表す値を作ります。その値をもとにfor inで処理を繰り返します。繰り返しの中で、ifによる選択を行い、変数iを3で割り切れる場合は変数の値と文字列「”は3の倍数です”」を表示します。

## 例題2

|  |
| --- |
| aisatsu = ['おはよう', 'こんにちは', 'こんばんは']  for i in range(1, 6):  for kotoba in aisatsu:  print(kotoba) |

### 【解説】

あいさつリストを作ります。次に、関数 range()を使い、1から30の範囲を表す値を作ります。その値をもとにfor inで処理を繰り返します。繰り返しの中で、あいさつリストの値をもとに更に繰り返し処理を行います。リストの要素全てについて、「変数kotobaに代入し、表示する」という処理を繰り返すことになるので、3つのあいさつの言葉を5回繰り返すプログラムになります。

## 例題3

|  |
| --- |
| kotae = 0  for n in range(1, 10):  print(n, "の段")  for i in range(1, 10):  kotae = n \* i  print( kotae, end=' ')  if kotae > 25:  break  print() |

### 【解説】

もとのプログラムは二重ループの1段目でifを使い、もし変数kotaeが25より大きければループをbreakで終了するプログラムであった。具体的には3x9=27を計算した時点で25より大きいため全ての処理が終了する。今回のプログラムは二重ループの2段目でifを使っているため、変数kotaeが25より大きくなっても1段目のループは終了せずに継続する。そのため、3x9=27を計算した後も1段目の処理が継続し、九九の4の段以降も計算が行われる。

## 問題集1

|  |
| --- |
| aisatsu = ["おはよう", "こんにちは", "こんばんは"]  for i in range(0, len(aisatsu)):  if i % 2 == 0:  print(aisatsu[i]) |

### 【解説】

あいさつリストを作ります。次に、関数 range()を使い、0からリストの要素数（この場合は3）の範囲を表す値を作ります。その値をもとにfor inで処理を繰り返します。繰り返しの中で、ifによる選択を行い、変数iを2で割り切れる場合は偶数とみなし、i番目のリストの値を表示します。

## 問題集2

|  |
| --- |
| for i in range(9, 0, -1):  print(i, "の段")  for j in range(9, 0, -1):  print( i \* j, end = ' ' )  print() |

### 【解説】

関数 range()を使い、9から1の範囲を表す値を作ります。その値をもとにfor inで処理を繰り返します。繰り返しの中で変数iの段と表示します。更に関数 range()を使い、9から1の範囲を表す値を作り繰り返し処理を行います。変数iとjを掛け算した結果を表示することで、九九を後ろから表示するプログラムになります。

## 問題集3

|  |
| --- |
| def dan\_keisan(n):  for j in range(1, 10):  print( n \* j, end = ' ' )  for i in range(1, 10):  print(i, "の段")  dan\_keisan(i)  print() |

### 【解説】

関数 kan\_keisan(n)を定義します。関数を呼び出すときに段数を渡すため、引数nを定義します。関数の中で、関数 range()を使い、1から9の範囲を表す値を作ります。そしてfor inを使った繰り返し処理を行い、変数iと変数nを掛け算した結果を表示します。

実際に呼び出すときには、関数 range()を使い、1から9の範囲を表す値を作ります。そして繰り返しの中で『変数iの段』と表示し関数dan\_kansan()の呼び出しを行います。