1. Assign the value 7 to the variable guess\_me. Then, write the conditional tests (if, else, and elif) to print the string 'too low' if guess\_me is less than 7, 'too high' if greater than 7, and 'just right' if equal to 7.

2. Assign the value 7 to the variable guess\_me and the value 1 to the variable start. Write a while loop that compares start with guess\_me. Print too low if start is less than guess me. If start equals guess\_me, print 'found it!' and exit the loop. If start is greater than guess\_me, print 'oops' and exit the loop. Increment start at the end of the loop.

3. Print the following values of the list [3, 2, 1, 0] using a for loop.

4. Use a list comprehension to make a list of the even numbers in range(10)

5. Use a dictionary comprehension to create the dictionary squares. Use range(10) to return the keys, and use the square of each key as its value.

6. Construct the set odd from the odd numbers in the range using a set comprehension (10).

7. Use a generator comprehension to return the string 'Got ' and a number for the numbers in range(10). Iterate through this by using a for loop.

8. Define a function called good that returns the list ['Harry', 'Ron', 'Hermione'].

9. Define a generator function called get\_odds that returns the odd numbers from range(10). Use a for loop to find and print the third value returned.

10. Define an exception called OopsException. Raise this exception to see what happens. Then write the code to catch this exception and print 'Caught an oops'.

11. Use zip() to make a dictionary called movies that pairs these lists: titles = ['Creature of Habit', 'Crewel Fate'] and plots = ['A nun turns into a monster', 'A haunted yarn shop'].

Solution

#### 1. Assign the value 7 to the variable guess\_me. Then, write the conditional tests (if, else, and elif) to print the string 'too low' if guess\_me is less than 7, 'too high' if greater than 7, and 'just right' if equal to 7.

In [1]:

**def** guess\_me(guess\_me):

**if** guess\_me **<** 7:

print('too Low')

**elif** guess\_me **>** 7:

print('too High')

**else**:

print('just Right')

guess\_me(guess\_me**=**7)

guess\_me(guess\_me**=**5)

guess\_me(guess\_me**=**15)

just Right

too Low

too High

#### 2. Assign the value 7 to the variable guess\_me and the value 1 to the variable start. Write a while loop that compares start with guess\_me. Print too low if start is less than guess me. If start equals guess\_me, print 'found it!' and exit the loop. If start is greater than guess\_me, print 'oops' and exit the loop. Increment start at the end of the loop

In [2]:

guess\_me **=** 7

start **=** 1

**while** **True**:

**if** start **<** guess\_me:

print('too low')

**elif** start **==** guess\_me:

print('found it')

**break**

**else**:

print('oops')

**break**

start **+=** 1

too low

too low

too low

too low

too low

too low

found it

#### 3. Print the following values of the list [3, 2, 1, 0] using a for loop.

In [3]:

in\_list **=** [3,2,1,0]

**for** ele **in** in\_list:

print(ele)

3

2

1

0

#### 4. Use a list comprehension to make a list of the even numbers in range(10)

In [4]:

print([x **for** x **in** range(10**+**1) **if** x**%2**==0 ])

[0, 2, 4, 6, 8, 10]

#### 5. Use a dictionary comprehension to create the dictionary squares. Use range(10) to return the keys, and use the square of each key as its value.

In [5]:

*# Method 1*

print(dict([(x,pow(x,2)) **for** x **in** range(10)]))

*# Method 2*

print({x:x**\*\***2 **for** x **in** range(10)})

{0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81}

{0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81}

#### 6. Construct the set odd from the odd numbers in the range using a set comprehension (10).

In [6]:

print({x **for** x **in** range(10) **if** x**%2** !=0})

{1, 3, 5, 7, 9}

#### 7. Use a generator comprehension to return the string 'Got ' and a number for the numbers in range(10). Iterate through this by using a for loop

In [7]:

gen\_com **=** ('Got\_'**+**str(x) **for** x **in** range(10))

**for** ele **in** gen\_com:

print(ele, end**=**' ')

Got\_0 Got\_1 Got\_2 Got\_3 Got\_4 Got\_5 Got\_6 Got\_7 Got\_8 Got\_9

#### 8. Define a function called good that returns the list ['Harry', 'Ron', 'Hermione'].

In [8]:

**def** good():

x **=** ['Harry', 'Ron', 'Hermione']

**return** x

print(good())

['Harry', 'Ron', 'Hermione']

#### 9. Define a generator function called get\_odds that returns the odd numbers from range(10). Use a for loop to find and print the third value returned.

In [9]:

**def** get\_odds():

output **=** []

**for** ele **in** range(10):

**if** ele**%2** != 0:

output**.**append(ele)

**yield** output

next(get\_odds())[2]

Out[9]:

5

#### 10. Define an exception called OopsException. Raise this exception to see what happens. Then write the code to catch this exception and print 'Caught an oops'.

In [10]:

**class** OopsException(Exception):

**pass**

**def** test(input):

**if** input **<**0:

**raise** OopsException(a)

**try**:

test(**-**100)

**except** Exception **as** e:

print('Caught in Oops ->',e)

Caught in Oops -> name 'a' is not defined

#### 11. Use zip() to make a dictionary called movies that pairs these lists: titles = ['Creature of Habit', 'Crewel Fate'] and plots = ['A nun turns into a monster', 'A haunted yarn shop'].

In [11]:

titles **=** ['Creature of Habit', 'Crewel Fate']

plots **=** ['A nun turns into a monster', 'A haunted yarn shop']

output **=** dict(zip(titles,plots))

print(output)

{'Creature of Habit': 'A nun turns into a monster', 'Crewel Fate': 'A haunted yarn shop'}