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]:
```
guess_me = 7
if guess_me < 7:
    print("too low")
elif guess_me > 7:
    print("too high")
else:
    print("just right")
```

```
just right
```

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("oops")
        break
    else:
        print("found it!")
        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]:
```
ls = [3, 2, 1, 0]
ss = {0 : 'First', 1 : 'Second', 2 : 'Third', 3 : 'Fourth'}
for idx,i in enumerate(ls):
    print("The " + ss[idx] + " element of the list is :", i)
```

```
The First element of the list is : 3
The Second element of the list is : 2
The Third element of the list is : 1
The Fourth element of the list is : 0
```

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

In [4]:
```
lst_even = [x for x in range(1,11) if x%2 == 0]
```

In [5]:
```
lst_even
```

Out[5]:
```
[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 [6]:
```
dict_square = {x : x * x for x in range(10)}
dict_square
```

Out[6]:
```
{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 [7]:
```
set_odd = {x for x in range(10) if x%2 !=0}
print(set_odd)
print(type(set_odd))
```

```
{1, 3, 5, 7, 9}
<class 'set'>
```

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 [8]:
```
limit = 10
string_generator = ('Got ' + str(num) for num in range(limit))
for item in string_generator:
    print(item)
```

```
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 [9]:
```
def good():
    return ['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 [10]:
```
def get_odds():
    for i in range(10):
        if i%2 !=0:
            yield i

count = 1
for number in get_odds():
    if count == 3:
        print("The third odd number is", number)
        break
    count+=1
```

```
The third odd number is 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 [11]:
```
class OopsException(Exception):
    pass


def with_exception(a):
    if a < 0:
        raise OopsException(a)


try:
    with_exception(-1)
except OopsException as err:
    print('Caught an oops')
```

```
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'].

In [12]:
```
titles = ['Creature of Habit', 'Crewel Fate']
plots = ['A nun turns into a monster', 'A haunted yarn shop']

movies = {}
for title, plot in zip(titles, plots):
    movies[title] = plot
# or movies = dict(zip(titles, plots))
print(movies)
```

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

In [ ]: