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.

Answer:

In [29]:

```
value=int(input())
guess_me=7
if value<guess_me:
    print('too low')
elif value> guess_me:
    print('too hight')
else:
    print('just right')

```

```
7
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.

Answer:

In [2]:

```
guess_me=7
start=1
while True:
    if start<guess_me:
        print('too low')
    elif start> guess_me:
        print('Oops')
    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.

Answer:

```
In [3]:
for value in [3,2,1,0]:
    print(value)
```

```
3
2
1
0
```

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

Answer:

```
In [4]:
[ num for num in range(10) if num%2==0]
```

```
Out[4]: [0, 2, 4, 6, 8]
```

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.

Answer:

```
In [5]:
squares={key : key*key for key in range(10)}
```

```
In [6]:
squares
```

```
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).

Answer:

```
In [7]:
odd={odd for odd in range(10)if odd%2!=0}
```

```
In [8]:
odd
```

```
Out[8]: {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.

Answer:

```
In [9]:
1 for fun in ('Got %s' % number for number in range(10)):
2     print(fun)
```

```
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'].

Answer:

```
In [10]:
1 def good():
2         return ['Harry', 'Ron', 'Hermione']
```

```
In [11]:
1 good()
```

```
Out[11]: ['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.

Answer:

In [12]:
```
def get_odds():
    for num in range(1,10,2):
        yield num

count=1
for num in get_odds():
    if count ==3:
        print('the third odd number is ', num)
        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'.

Answer:

In [13]:
```
class OopsException(Exception):
        pass
```

In [14]:
```
raise OopsException()
```

```
---------------------------------------------------------------------------
OopsException                             Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_20952/2009335150.py in <module>
----> 1 raise OopsException()

OopsException:
```

In [19]:
```
try:
    raise OopsException
except OopsException:
    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'].

Answer:

```
In [15]: 1 titles=['Creature of Habit','Crewel Fate']

In [16]: 1 plots=['A num turns into a monster', 'A haunted yarn shop']

In [17]: 1 movies=dict(zip(titles,plots))

In [18]: 1 movies

Out[18]: {'Creature of Habit': 'A num turns into a monster',
          'Crewel Fate': 'A haunted yarn shop'}
```