- 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 = int(input("enter value : "))
if guess_me == 7 :
    print('Just Right')
elif guess_me > 7 :
    print("Too High")
else:
    print('Too Low')
```

```
enter value : 8
Too High
```

- 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!
```

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

In [8]:
```
lst = [3,2,1,0]
for i in lst:
    print(i)
```

```
3
2
1
0
```

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

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

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

- 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]:
```
squares = {num: num * num for num in range(10)}
squares
```

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

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

In [7]:
```
odd = {item for item in range(10) if item%2==1}
odd
```

Out[7]:
```
{1, 3, 5, 7, 9}
```

- 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 [9]:
```
string_generator = ('Got ' + str(num) for num in range(10))
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
```

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

In [11]:
```
def good():
    return ['Harry', 'Ron', 'Hermione']
good()
```

Out[11]:
```
['Harry', 'Ron', 'Hermione']
```

- 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 [16]:
```
def get_odds():
    for number in range(1,10,2):
        yield number

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
```

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

In [17]:
```
class OopsException(Exception):
    pass

def raiseException(num):
    if num < 0:
        raise OopsException(num)

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

```
Caught an oops
```

- 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 [20]:
```
titles = ['Creature of Habit', 'Crewel Fate']
plots = ['A nun turns into a monster', 'A haunted yarn shop']
dict(zip(titles,plots))
```

Out[20]:
```
{'Creature of Habit': 'A nun turns into a monster',
 'Crewel Fate': 'A haunted yarn shop'}
```

In [ ]:
```

```