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 whileloop that compares start with guess_me. Print too low if start is less than guess me. If start equalsguess_me, print 'found it!'and exit the loop. If start is greater than guess_me, print 'oops'and exitthe loop. Increment start at the end of the loop.

In [3]:

```
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 [4]:

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

```
3
2
1
0
```

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

In [6]:

```
print([i for i in range(2,10+1) if i%2==0])
```

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

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

In [7]:

```
print({i:i**2 for i in range(1,11)})
```

```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100}
```

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

In [9]:

```
print({i for i in range(1,10) if i%2==0})
```

```
{8, 2, 4, 6}
```

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 [10]:

```
def gene():
    for i in range(1,11):
        yield "Got",i


k=gene()

for i in range(10):
    print(next(k))


```

```
('Got', 1)
('Got', 2)
('Got', 3)
('Got', 4)
('Got', 5)
('Got', 6)
('Got', 7)
('Got', 8)
('Got', 9)
('Got', 10)
```

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

In [11]:

```
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 [17]:

```
def gene():
    for i in range(1,10):
        if i%2!=0:
            yield i

gen=gene()
for i in range(3):
    d=next(gen)
    if i==2:
        print(d)


```

```
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 [19]:

```
class OopsException(Exception):
    pass
try:
    raise OOpsException()
except Exception as e:
    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 [22]:

```
titles = ['Creature of Habit','Crewel Fate']
plots = ['A nun turns into a monster', 'A haunted yarn shop']
l=list(zip(titles,plots))
k={}
for i in range(len(l)):
    k[l[i][0]]=l[i][1]

print(k)
```

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