```
# Q1. Which keyword is used to create a function? Create a function to return a list of odd numbers in the range of 1 to 25.
Ans def key word is used for create a function
```

In [1]:
```
def odd_list(l):
    l=[]
    for i in range(1,25):
        if i%2!=0:
            l.append(i)
    return l
```

In [ ]:
```
# Q2. Why *args and **kwargs is used in some functions? Create a function each for *args and **kwargs
# to demonstrate their use.

*args-*args allows us to pass a variable number of non-keyword arguments to a
Python function.


**kwargs-**kwargs allows us to pass a variable number of keyword arguments to a
Python function in key value pair
```

In [1]:
```
# *args
def check(*args):
    return args
```

In [2]:
```
check(1,2,3,4,5)
```

Out[2]: (1, 2, 3, 4, 5)

In [ ]:
```
# for **kwargs
def check1(**kwargs):
    return kwargs
```

In [5]:
```
check1(a=[1,2,3,4],b='ash')
```

Out[5]: {'a': [1, 2, 3, 4], 'b': 'ash'}

In [13]:
```
"""Que 3-What is an iterator in python? Name the method used to initialise the iterator object and the method
used for iteration. Use these methods to print the first five elements of the given list [2, 4, 6, 8, 10, 12, 14,
16, 18, 20]."""


"""Ans 3-

iterator-Repetitive execution of the same block of code over and over is referred to as iteration

Method-iter() are used for the initialise the iterator object and the method

Method -next() are used for iteration """

a='ashish'
b=iter(a)
print(next(b))
print(next(b))
```

```
a
s
```

In [8]:
```
"""Q4. What is a generator function in python? Why yield keyword is used? Give
an example of a generator function."""

"""Ans- they allow you to define an iterative algorithm by writing a single
function whose execution is not continuous"""

def new_fib(n):
    a,b=0,1
    for i in range(n):
        yield a
        a,b=b,a+b
```

In [10]:
```
for i in new_fib(10):
    print(i)
```

```
0
1
1
2
3
5
8
13
21
34
```

In [4]:
```
"""Que 5 Q5. Create a generator function for prime numbers less than 1000. Use
the next() method to print the first 20 prime numbers."""

def isPrime(n):
    if n <= 1 :
        yield False
    for i in range(2, n):
        if n % i == 0:
            yield False

    yield True

def a_Prime(n):
    for i in range(2, n + 1):
        if isPrime(i):
            print(i, end = " ")


n=100
print(a_Prime(n))
```

```
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 8
9 90 91 92 93 94 95 96 97 98 99 100 None
```

In [ ]:
```
""" Q6. Write a python program to print the first 10 Fibonacci numbers using
 a while loop."""

n=int(input('hi user please provide the no'))
a,b=0,1
c=0
while c<n:
    print(a)
    c1=a+b
    a=b
    b=c1
    c=c+1
```

In [5]:
```
"""Q7. Write a List Comprehension to iterate through the given string: ‘pwskills’.
Expected output: ['p', 'w', 's', 'k', 'i', 'l', 'l', 's']"""

s='pwskills'
[i for i in s]
```

Out[5]:
```
['p', 'w', 's', 'k', 'i', 'l', 'l', 's']
```

In [7]:
```
"""Q8. Write a python program to check whether a given number is Palindrome or
not using a while loop."""

n=int(input("Enter number:"))
temp=n
rev=0
while(n>0):
    dig=n%10
    rev=rev*10+dig
    n=n//10
if(temp==rev):
    print("The number is a palindrome!")
else:
    print("The number isn't a palindrome!")
```

```
Enter number:20
The number isn't a palindrome!
```

Type *Markdown* and LaTeX: α^2

In [9]:
```
"""Q9. Write a code to print odd numbers from 1 to 100 using list comprehension."""
[i for i in range(100) if i%2!=0  ]
```

Out[9]:
```
[1,
 3,
 5,
 7,
 9,
 11,
 13,
 15,
 17,
 19,
 21,
 23,
 25,
 27,
 29,
 31,
 33,
 35,
 37,
 39,
 41,
 43,
 45,
 47,
 49,
 51,
 53,
 55,
 57,
 59,
 61,
 63,
 65,
 67,
 69,
 71,
 73,
 75,
 77,
 79,
 81,
 83,
 85,
 87,
 89,
 91,
 93,
 95,
 97,
 99]
```

In [ ]: