In [ ]:
```
# while loop
```

In [3]:
```
n=int(input('enter an input'))
i=0
while i<=n:
    print(i,end=' ')
    i=i+1
```

```
enter an input10
0 1 2 3 4 5 6 7 8 9 10
```

In [ ]:
```
# reverse of anumber 123- 321
n=int(input('enter value'))
rev=0
while n>0:
    #r=n%10
    #rev=rev*10+r
    rev= rev*10+n%10
    n=n//10

print(rev)
```

```
enter value123
```

In [12]:
```
123%10
```

Out[12]: 3

In [13]:
```
12%10
```

Out[13]: 2

In [ ]:
```
# palindrome
n=int(input('enter value'))
m=n
rev=0
while n>0:
    #r=n%10
    #rev=rev*10+r
    rev= rev*10+n%10
    n=n//10
if m==rev:
    print(m,'is palindrome')
else:
    print('not palindrome')
```

function

- it is a group of statements to do a specific task
- funcyion breaks a code into small modules to look more organise
- code re-useability
- types of functions
  - built in functions
  - user defined functions

In [15]:
```
# list of built in functions or (predefined functions)
dir(__builtins__)
```

Out[15]:
```
['ArithmeticError',
 'AssertionError',
 'AttributeError',
 'BaseException',
 'BlockingIOError',
 'BrokenPipeError',
 'BufferError',
 'BytesWarning',
 'ChildProcessError',
 'ConnectionAbortedError',
 'ConnectionError',
 'ConnectionRefusedError',
 'ConnectionResetError',
 'DeprecationWarning',
 'EOFError',
 'Ellipsis',
 'EnvironmentError',
 'Exception',
 'False',
 'FileExistsError',
 'FileNotFoundError',
 'FloatingPointError',
 'FutureWarning',
 'GeneratorExit',
 'IOError',
 'ImportError',
 'ImportWarning',
 'IndentationError',
 'IndexError',
 'InterruptedError',
 'IsADirectoryError',
 'KeyError',
 'KeyboardInterrupt',
 'LookupError',
 'MemoryError',
 'ModuleNotFoundError',
 'NameError',
 'None',
 'NotADirectoryError',
 'NotImplemented',
 'NotImplementedError',
 'OSError',
 'OverflowError',
 'PendingDeprecationWarning',
 'PermissionError',
 'ProcessLookupError',
 'RecursionError',
 'ReferenceError',
 'ResourceWarning',
 'RuntimeError',
 'RuntimeWarning',
 'StopAsyncIteration',
 'StopIteration',
 'SyntaxError',
 'SyntaxWarning',
 'SystemError',
 'SystemExit',
```

```
 'TabError',
 'TimeoutError',
 'True',
 'TypeError',
 'UnboundLocalError',
 'UnicodeDecodeError',
 'UnicodeEncodeError',
 'UnicodeError',
 'UnicodeTranslateError',
 'UnicodeWarning',
 'UserWarning',
 'ValueError',
 'Warning',
 'WindowsError',
 'ZeroDivisionError',
 '__IPYTHON__',
 '__build_class__',
 '__debug__',
 '__doc__',
 '__import__',
 '__loader__',
 '__name__',
 '__package__',
 '__spec__',
 'abs',
 'all',
 'any',
 'ascii',
 'bin',
 'bool',
 'breakpoint',
 'bytearray',
 'bytes',
 'callable',
 'chr',
 'classmethod',
 'compile',
 'complex',
 'copyright',
 'credits',
 'delattr',
 'dict',
 'dir',
 'display',
 'divmod',
 'enumerate',
 'eval',
 'exec',
 'filter',
 'float',
 'format',
 'frozenset',
 'get_ipython',
 'getattr',
 'globals',
 'hasattr',
 'hash',
```

```
 'help',
 'hex',
 'id',
 'input',
 'int',
 'isinstance',
 'issubclass',
 'iter',
 'len',
 'license',
 'list',
 'locals',
 'map',
 'max',
 'memoryview',
 'min',
 'next',
 'object',
 'oct',
 'open',
 'ord',
 'pow',
 'print',
 'property',
 'range',
 'repr',
 'reversed',
 'round',
 'set',
 'setattr',
 'slice',
 'sorted',
 'staticmethod',
 'str',
 'sum',
 'super',
 'tuple',
 'type',
 'vars',
 'zip']
```

In [32]:
```
# user defined functoins
### syntax in c
```
function fname(){
    condition/stmts to execute
}```
syntax in python
```
def fname:
    cond/stmts
    return
fname()
```
- advantages
  - making large into small modules
  - reuse of code in a function by calling its fname
- types of argumented in functions
 - required arguments
 - keyword
 -default
 -variable
```

```
  File "<ipython-input-32-acf6061ba15b>", line 3
    ```
    ^
SyntaxError: invalid syntax
```

In [27]:
```
a=4
b=10
sum([a,b])
```

Out[27]: 14

In [24]:
```
a=[1,2,3,4,5]
max(a)
```

Out[24]: 5

In [25]:
```
len(a)
```

Out[25]: 5

In [26]:
```
min(a)
```

Out[26]: 1

In [31]:
```
a=('ganesh')
len(a)
```

Out[31]: 6

In [29]:
```
len(a)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-29-1a2e6ec5f1e3> in <module>
----> 1 len(a)

TypeError: object of type 'int' has no len()
```

In [15]:
```
def add(a,b):
    c=a+b
    return c
a=int(input('enter a value'))
b=int(input('enter b value'))
add(a,b)
```

```
enter a value5
enter b value7
```

Out[15]: 14

In [ ]:
```
# keyword arguments
def key(str):
    print(str)
key(str=123)
```

In [17]:
```
def keyw(name,clz):
    print('name:',name)
    print(college:,'clz')
keyw(name='abc',college='aits')
```

```
  File "<ipython-input-17-0d3001c514bb>", line 3
    print(college:,'clz')
                  ^
SyntaxError: invalid syntax
```

In [11]:
```
### default arguments
def default(a=5,b=7):
    print(a,b)
default(a,b)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-11-c64f1deab278> in <module>
      2 def default(a=5,b=7):
      3     print(a,b)
----> 4 default(a,b)

NameError: name 'a' is not defined
```

In [19]:
```
# default arguments
def default(1,r=1):
    print(1,r)
default(1='11',r='a')
default(1='13')
```

```
  File "<ipython-input-19-cae723142003>", line 2
    def default(1,r=1):
                ^
SyntaxError: invalid syntax
```

In [ ]:
```
# task
- print n natural numbers using funccheckction
-
```

In [21]:
```
# n odd numbers using functions
n=int(input('enter the value'))
def odd(n):
    for i in range(1,n+1):
        if i%2 !=0:
            print(i,end=' ')
    return
odd(n)
```

```
enter the value50
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
```

In [32]:
```
n=int(input('enter a number'))
def prime(n):
    c=0
    for i in range(1,n+1):
        if n%i==0:
            c=c+1
    if c==2:
        print(n,'is prime')

    else:
        print('not a prime')
prime(n)
```

```
enter a number2
2 is prime
```

In [ ]:

In [ ]: