FUNCTION ARGUMENTS

Default arguments

```
def inp(s,b=10):
   . . .
```

Default value is provided in function definition If parameter is omitted, default value is used

Default value must be available at definition time

Default arguments

```
def f(a,b,c=15,d=32):
   . . .
```

`f(13,12)` is interpreted as `f(13,12,15,32)`

`f(13,12,26)` is interpreted as `f(13,12,26,32)`

`f(15,56,45,12) a=15 ,b=56 ,c=45,d=12`

Parameter Order is important

Its values are identified by position,

Default values must come at the end

Function definitions

`def` associates a function body with a name

Flexible, like other value assignments to name

Definition can be declared inside conditional

Statements

```
if condition:
  def u(a,b,c):
    . . .
else:
  def u(a,b,c):
    . . .
```

Function definitions

Can assign a function to a new name

```
def f(a,b,c):
   . . .

g = f
```

Now `g` is another name for `f`

Can pass functions

Implement `f` to `x` `n` times

```
def imp(f,x,n):
    r = x
  for i in range(n):
    r = f(r)
  return(r)
```

```
def sq(x):
  return(x*x)

imp(sq,5,2)

sq(sq(5))
```