

```
dir(str.format)
print help(str.format)
print str.format.__doc__
```

```
#####
```

```
# Numeros #
```

```
#####
```

```
n = 12 #entero
n = 3.8 #decimal
n = 0xF #hexadecimal
n = 0b11000101 #binario
n = 010 #octal
n = 3e2 #potencia en base 10
n = 2+4j #complejos
```

```
#####
```

```
# Colecciones #
```

```
#####
```

```
a = (1,2,3) #Tupla
a = (1,3) + ("a","c","d") #Suma de tuplas concatena al final
a = [1,3,5] #Lista
a.sort()
a.sort(reverse=True)
a = {"a":12,"b":43} #Diccionario
a["c"] = (78,"76")
a = set([1,2,2,2,3]) #Conjunto
a = [x for x in range(101) if x % 2 == 0] #lista por comprension
b = [x for x in range(51) if x % 2 == 0]
c = [x for x in a if x not in b]
a = [(x,y) for x in a for y in a] #lista de tuplas
```

```
#####
```

```
# Funciones #
```

```
#####
```

```
def foo(a,b,arg1=True,arg2=2):
    return a+arg2
```

```
def foo(*args): #funcion con multiples parametros
    return args[0]+args[1]
```

```
def foo(**args): #funcion con multiples parametros con nombre
    c = args["a"] + args["b"]
    return c*2
```

```
a = foo
foo(a=3,b=4)
```

```
def mix(a,b,c,*arg,**argd): #mix de multiples parametros posicionales y con nombre
    return a + arg[0] + argd["hola"]
```

```

f = lambda x : x*2 #Funciones lamda

#####
# Clases #
#####
class Usuario(object):
    def __init__(self,nombre): #constructor
        self.__nombre = nombre

    def GetNombre(self): #get al estilo java
        return self.__nombre

    def _setNombre(self,v): #set usando convencion para que no usen el metodo
        self.__nombre = v

nombre = property(fget=GetNombre, fset=_setNombre) #get y set como propiedad

def Locura(self):
    print "Locura %s" % self.__nombre
    self.Loco = True #agregando otro atributo

def __str__(self): #ToString del usuario
    return self.__nombre

us1 = Usuario("Carla")
print us1.__Usuario__nombre #Acceder al atributo "privado". Poco feliz
print us1.nombre #accedo a la propiedad

users =[Usuario("Pepe"), Usuario("Juan"),Usuario("Maria")]
for u in users:
    print u #se invoca el __str__ del Usuario

#####
# Herencia #
#####
class UsuarioHombre(Usuario): #Hereda de usuario
    pass #palabra clave para no definir nada

h1 = UsuarioHombre("Jose")
print h1 #Se invoca el __str__ del Usuario ya que UsuarioHombre hereda de Usuario

class Mujer:
    pass

class UsuarioMujer(Usuario,Mujer): #Herencia multiple
    def __init__(self, nombre):
        Usuario.__init__(self,nombre) #llamando al constructor del padre
        self.Sexo = "F"

```

```

m1 = UsuarioMujer("Caro")
print m1.Sexo
#####
# Archivos #
#####
f = open("C:\\pepe.txt") # se debe poner \\ debido a que \ es un caracter de escape
f = open(r"C:\pepe.txt") # r para String en crudo, no se intepretan caracteres de escape

try: # atrapo errores
    l = f.readline() # lee una linea sola
    f.seek(0) # vuelve al principio
    lineas = f.readlines() # lee todas las lineas
    for l in lineas:
        print l

    todojunto = "".join(lineas) #junto todas las lineas
    aux = todojunto.split("\n") #separo por salto de linea

    aux = [x[:-1] for x in lineas] #quito el ultimo caracter de cada linea
    print aux
finally:
    f.close() #cierro archivo

f = open(r"C:\pepe.txt", "r+") # "r+" modo lectura escritura, "r" modo lectura, "w" modo escritura,
    # "a" append

```