

Clase 13

Diego Armando Santillán Arriaga

23/01/19

1 Funciones recursivas:

Compuestas por 2 partes: base de recursividad (condicionales iniciales) y regla de recursividad que va a generar los términos sucesivos. Se utiliza la función para definirla (se define recursivamente).

Ejemplo: programa para calcular los términos de la sucesión de Fibonacci.

```
def fib(n):
    """ Calcula el nesimo término
    de la sucesión de fibonacci con n natural
    """
    if n>2:
        return fib(n-1)+fib(n-2)
    else:
        return 1
```

Se introduce la función dentro de la función de forma que eventualmente se llegue a las condiciones iniciales. Así se hacen varias iteraciones similares a las de los ciclos for y while.

2 Ámbitos de validez de las variables

Una variable hace referencia a un objeto. Los ámbitos de validez de una variable son: **local o global**. Las **variables locales** son aquellas que pertenecen a una función, son las variables de la función. Las **variables globales** no están vinculadas a ninguna función ni ciclo, se declaran solas en el shell. Una variable local, por ejemplo de una función puede ocultar a una variable global. Las variables globales son las que no tienen sangría, las variables locales, son locales a un bloque de una función. Una variable global está disponible para todas las funciones y con todas ellas conserva su valor. Ejemplo:

```
>>>T=167 # Variable global
>>>def mult(n):    # n es una variable local
    return 2*n
```

Si queremos utilizar una variable global dentro de una función se escribe **global "variable"**. Esto indica que la variable que estamos utilizando en

la función es la global. Las variables locales solamente existen durante la ejecución de un programa (dentro de la función).

Si de forma explícita se introducen argumentos en una función es posible introducir valores en cualquier orden. Esto significa que en los argumentos se escriba "variable"="valor". También se pueden declarar variables globales y luego utilizarlas en una función, solo basta escribirlas como argumentos.