

Variables en



En Python los tipos de datos básicos son:

- Números (enteros, flotantes, Long, complejos)
- Cadenas de texto
- Listas
- Tuplas (secuencias)
- Diccionario

Declarar y asignar valores a una variable

Para realizar la declaración de la variable lo hacemos de la siguiente manera.

```
numero = 17      # numero entero  
decimal = 0.5    # punto flotante  
nombre = "Jorge" # cadena de texto
```

En Python las variables son contenedores que reconocen el tipo de dato que se le ingresa.

Utilizar nombres descriptivos y en minúsculas. Para nombres compuestos, separar las palabras por guiones bajos. Antes y después del signo =, debe haber uno (y solo un) espacio en blanco.

Declarar y asignar valores a una variable

Tupla

Una tupla es una variable que permite almacenar varios datos inmutables (no pueden ser modificados una vez creados) de tipos diferentes:

```
mi_tupla = ('cadena de texto', 15, 2.8, 'otro dato', 25)
```

Se puede acceder a cada uno de los datos mediante su índice correspondiente, siendo 0 (cero), el índice del primer elemento:

```
print mi_tupla[1] # Salida: 15
```

Declarar y asignar valores a una variable

Listas:

Una lista es similar a una tupla con la diferencia fundamental de que permite modificar los datos una vez creados:

```
mi_lista = ['cadena de texto', 15, 2.8, 'otro dato', 25]
```

A las listas se accede igual que a las tuplas, por su número de índice:

```
print mi_lista[1] # Salida: 15  
print mi_lista[1:4] # Devuelve: [15, 2.8, 'otro dato']  
print mi_lista[-2] # Salida: otro dato
```

Las lista NO son inmutables: permiten modificar los datos una vez creados:

```
mi_lista[2] = 3.8 # el tercer elemento ahora es 3.8
```

Las listas, a diferencia de las tuplas, permiten agregar nuevos valores:

```
mi_lista.append('Nuevo Dato')
```

Declarar y asignar valores a una variable

Diccionarios

Mientras que a las listas y tuplas se accede solo y únicamente por un número de índice, los diccionarios permiten utilizar una clave para declarar y acceder a un valor:

```
mi_diccionario = {'clave_1': valor_1, 'clave_2': valor_2, 'clave_7': valor_7}  
print mi_diccionario['clave_2'] # Salida: valor_2
```

Un diccionario permite eliminar cualquier entrada:

```
del(mi_diccionario['clave_2'])
```

Al igual que las listas, el diccionario permite modificar los valores

```
mi_diccionario['clave_1'] = 'Nuevo Valor'
```

GRACIAS POR SU ATENCIÓN

Sandra Lorena Hoyos
Cead- Palmira