

Datos

Tipos de datos

Números

En Python como en cualquier lenguaje de programación se hace uso de los números pero estos números están clasificados en diferentes secciones:

- **Enteros**

Para números sin decimales se usan para generar constantes y operaciones básicas o no tan complejas para llevar decimales. Estos se declaran con la sintaxis `int` a diferencia de otros lenguajes de programación Python no tiene límite de número es decir puedes escribir tantos números como quieras no como en C++ que este tiene un límite de

`2147483647`

ejemplo:

```
numero = int(20) #La variable tiene el numero 20
```

Que pasa si no pongo el tipo de dato?

hago énfasis en esto porque Python deja introducir los datos aun sin declarar el tipo de dato pues lo tomara bien esto siendo solo si es declarado constante es decir que no se modificara luego

ejemplo:

```
numero = 20 #esto esta bien pues lo declara como entero  
texto = hola como estas #Esto igual es correcto pues lo declara como string
```

el código declara automáticamente si es directa del código y no ingresado desde el teclado es:

```
numero = input("escribe el numero:") #Esto esta incorrecto pues lo toma como  
cadena
```

aquí el número está siendo ingresado por teclado pero el que se supone que tiene que ser número está siendo ingresado como cadena de caracteres(string) siendo que si lo sumamos saldría algo así:

```
numero1 = input(20)
numero2 = input(10)
print(numero1 + numero2)
```

Esto nos imprimirá 2010 lo tomara como un string y simplemente unirá los dos "textos"

Reales

Para declarar que una variable numérica es de tipo real se usa `float` al igual que los enteros no tiene limite.

ejemplo:

```
numero = float(11.5)
numero2 = 11.5
```

Complejos

los datos complejos son aquellos que presentan una parte real y una parte imaginaria y se declara con la función `complex` esta devuelve un número imaginario a partir de los valores cedidos como argumentos para la parte real e imaginaria del mismo, o a partir de una cadena de texto conteniendo la representación de un número complejo.

hay dos formas para declararlo en código, la primera es primero poniendo la parte real y luego la parte imaginaria separadas por una coma, la parte imaginaria se identifica con la letra `j` o la `J`:

```
numero = complex(1 , 3)
print(numero)
```

Esto nos imprimirá `1+3J` que es la representación grafica de un numero complejo. La otra forma para declararlo es usando una cadena compleja escribiendo el numero real mas el simbolo `+` y el numero imaginario con una `J` al lado:

```
numero = complex(1+3j)
print(numero)
```

Texto

En Python el texto solo se maneja con cadenas de caracteres llamados `string` estas cadenas pueden almacenar números, letras , caracteres, frases incluso espacios, para escribir en

cadenas de caracteres se usan comillas simples `' '` o dobles `" "` , en código se usa la función `str`

```
texto = str("Hola mundo")
texto = "Hola mundo"
```

El carácter especial para el texto siendo las comillas dobles no pueden ser usadas en el valor de la variable es por eso que se usa la `\` para poner comillas

```
texto = "\"hola\" mundo"
print(texto)
```

Al igual que con los comentarios se puede usar la triple comilla para hacer textos de mas líneas

```
texto = """

este texto tendra 1 espacio de por medio

"""
print(texto)
```

También se puede almacenar varios textos en una sola variable en paréntesis `()` y separadas en comas:

```
textos = ("Hola" , "mundo" , "Python")
print(textos)
```

Concatenar texto

Para concatenar texto o cadenas de texto se usa el carácter de suma `+` que este funcionara para fusionar dos textos

```
texto1 = "hola"
texto2 = "mundo"
print(texto1 + texto2)
```

y se pueden repetir con el carácter del asterisco `*`

```
texto1 = "hola"
texto2 = "mundo"
print( 3 * texto1 + texto2)
```

aquí se repetiría el "hola" 3 veces.

Dos o más *cadenas literales* (es decir, las encerradas entre comillas) una al lado de la otra se concatenan automáticamente.

```
print('hola' 'mundo')
```

Esto solo funciona con dos literales, no con variables ni expresiones, para concatenar variables y texto se usa el carácter ya mencionado de suma +

```
texto = "hola"
print(texto + "mundo")
```

Subíndices en texto

Esto hace encontrar la letra en cierta posición esta empezara desde el 0 en adelante, para hacer esto se usa una variable y al llamar la variable se usara un "[]" para la posición:

```
palabra = "Python"
print(palabra[0])
#esto nos dara la primera letra que es "P"
```

Este también puede ser negativo, esto hará que lea de derecha a izquierda empezando desde el -1 dando la letra que esta de ultimo si es -1

```
palabra = "python"
print(palabra[-1] + " Esta es la ultima letra...")
print(palabra[-3] + " Esta es la antepenultima letra...")
```

si pones -0 es lo mismo que poner 0 lo que significa que te dará la primera letra:

```
palabra = "python"
print(palabra[-0])
```

Texto en Slicing

Mejor dicho como la parte de un texto, con esto seleccionaremos de que letra a que letra queremos que imprima para hacer esto usaremos el mismo "[]" pero con dos números siendo las posiciones de las letras separadas por dos puntos:

```
palabra = "Python"
print(palabra[2:5])
#aquí imprimiria desde la letra en la posicion 2 que es la "t" hasta la quinta
que es la "o" recordemos que empieza desde el 0
```

El slicing tiene valores por defecto siendo este dejar el numero en nulo es decir no escribir el numero, esto hará que si el numero nulo es el de la izquierda empieza el texto desde el inicio del texto y si es el de la derecha será hasta la ultima letra del texto:

```
palabra = "Python es lo mejor"
print(palabra[2:] + " :Aquí empieza desde la letra con 2da posicion hasta el
resto de la frase")
print(palabra[:6] + " :Aquí empieza desde el principio de la frase hasta la
letra con la posicion 6")
```

También pueden usarse negativos:

```
texto = "python es lo mejor"
print(texto[-5:])
#aquí el texto empezara desde la anteantepenultima letra hasta el final de
la frase
```

Una forma de recordar cómo funcionan las rebanadas es pensar que los índices apuntan *entre* caracteres, con el borde izquierdo del primer carácter numerado 0. Luego, el punto derecho del último carácter de una cadena de n caracteres tiene un índice n , por ejemplo

```
+---+---+---+---+---+---+
| P | y | t | h | o | n |
+---+---+---+---+---+---+
0   1   2   3   4   5   6
-6  -5  -4  -3  -2  -1
```

tampoco te pases de listo poniendo un numero mayor al tamaño de la frase o palabra pues esta dará error

una forma para saber el tamaño de un texto es con la función `len()` :

```
texto = "Python es el lenguaje de programación mas facil"  
print(len(texto))  
#nos devolvera el numero de caracteres siendo en este caso 47
```

El texto tiene muchas mas cosas que ver y mas los strings hablaremos mas a fondo en su respectiva nota.