

Análisis de datos

Nivel Básico – Explorador

Misión 1

Tipos de datos

Números

Los números son principalmente de dos tipos en Python: enteros ('int') y decimales ('float').

a. Enteros

El tipo de dato 'int' son los números enteros; dicho de otra manera, son los números que hacen parte del conjunto de números enteros (todos los números negativos y positivos, incluyendo el 0). La sintaxis literal [1] es simplemente el número deseado.

```
18
```

b. Decimales

El tipo de dato 'float' (llamado así en inglés por tener un punto "flotante") son los números decimales, o los que tienen un punto en medio. La sintaxis literal de este tipo de dato puede variar:

- i. Escribir un número entero seguido de un punto y otro número.

```
2.5
```

- ii. Escribir un número entero seguido de un punto crea el mismo número, pero float, donde la parte decimal es 0.

```
18.
```

- iii. Escribir un punto seguido de un número entero crea un float, donde la parte entera es 0.

```
.05
```

Texto

Los tipos de datos textuales o 'str' son conocidos como "string" (cadena, en inglés) debido a que son una cadena de **caracteres** [2]. Es preciso anotar que, para crear un tipo de dato textual, no se puede simplemente escribir el texto y nada más, debido a que el código no podría diferenciar la cadena de texto del resto de código; para esto se debe utilizar un indicador con el fin de definir una sección que corresponde a un tipo de dato textual.

a. Comillas simples

La primera sintaxis para un string utiliza comillas simples, donde el dato textual que deseemos crear irá encerrado entre estas comillas simples.

```
'Hola Mundo'
```

b. Comillas dobles

Esta sintaxis funciona exactamente de la misma manera, solo que ahora utiliza comillas dobles.

```
"Hola Mundo"
```

c. Comillas triples

Esta sintaxis trae una función espacial, ya que a diferencia de las anteriores permite hacer saltos de línea dentro del dato textual. Consiste en utilizar una de las comillas anteriores, pero escribiéndola tres veces en cada extremo.

```
"""Hola  
Mundo"""
```

d. Concatenar strings

Una técnica importante a la hora de utilizar strings consiste en concatenar cadenas de texto para poder convertirlas en un mismo string. Esto se hace poniendo el primer string seguido de un signo mas "+" y finalmente el último string; a este proceso de unir strings se lo llama **concatenación**[3].

```
"Hola" + "Mundo"
```

Booleano

Un tipo de dato 'bool' es un dato que solo puede tener dos valores, Falso o Verdadero, y puede ser muy útil para cualquier tipo de dato que se quiera representar con solo dos estados.

- a. El primer estado de un 'bool' es Falso, y su sintaxis es la palabra False, en inglés, y capitalizada [4] (todo en minúscula, pero empezando por mayúscula).

False

- b. El primer estado de un 'bool' es Verdadero, y su sintaxis es la palabra True, en inglés, y capitalizada [4] (todo en minúscula, pero empezando por mayúscula).

True

Variable

En un punto se hace necesario almacenar nuestro dato en algún lugar para su posterior uso o modificación, y es allí donde aparecen las variables. Como su nombre lo dice, su valor puede variar, siendo estas no más que un espacio de memoria en nuestro computador donde se puede almacenar información.

a. Declaración de una variable

Una variable se crea (o, dicho técnicamente, se [declara \[5\]](#)) al escribir el nombre que se le quiera dar (conocido técnicamente como [identificador \[6\]](#)) seguido del signo asignar ("=") y del dato que se quiere asignar.

variable = 18

b. Reglas para declarar una variable

Debido a la manera en la que funciona el código de programación, se deben respetar ciertas reglas a la hora de declarar variables.

- i. El identificador no puede empezar por dígitos ni [caracteres especiales \[7\]](#) (a excepción del guion bajo "_").
- ii. El resto del identificador no debe contener espacios ni caracteres especiales (a excepción del guion bajo "_"), pero sí puede contener dígitos.

iii. Los identificadores son sensibles a las mayúsculas, es decir, la variable `myvariable` se identificará como diferente a `myVariable`. Nótese que la diferencia de las variables es la `v` en mayúscula.

c. Convenciones para declarar una variable

Un grupo de personas puede establecer un acuerdo para que todos hagan algo de la misma forma, ya sea por eficiencia, facilidad o incluso comodidad. A esto se lo llama convención. [8] Para declarar una variable, por ejemplo, se han establecido varias convenciones con el fin de mejorar su lectura y distinción.

i. No utilizar una simple letra para una variable.

ii. En Python, la más famosa es Snake Case, que consiste en que todo va en minúscula y las palabras están separadas por guiones bajos.

```
my_variable = 18
```

iii. Si el objetivo de una variable es funcionar como constante, o sea que su valor nunca cambiará, se escribe todo en mayúscula y las palabras se separan por guiones bajos.

```
PI_NUMBER = 3.1416
```

Lectura por consola

Casi siempre será necesario que haya una interacción del programa con el usuario. Para fines prácticos, esta interacción se llevará a cabo a través de la consola, en la que se imprime un mensaje pidiendo lo que se necesita del usuario, y el usuario procede a ingresar el dato. Para esto utilizaremos las funciones `print()` e `input()`, y aunque aún no profundizaremos en qué es una función, podemos usarlas sin ninguna dificultad.

a. `print()`

La función `print`, como su nombre en inglés lo dice, permite imprimir cualquier dato en la consola. Basta con escribir la palabra `print`, y entre los paréntesis irá el dato que se desea imprimir, el cual puede ser cualquier tipo de dato.

```
my_name = "Santiago"
print(my_name)
```

Consola:

Santiago

b. input()

La función input permite ingresar un dato a través de la consola, el cual se podrá almacenar en una variable como un 'str' o hacer algo con él. Si bien se puede escribir la función input() sin más, al igual que con print(), se puede ingresar un dato dentro de los paréntesis para que imprima un mensaje antes de esperar el ingreso de dato.

```
my_message = "Ingrese su Nombre: "
my_name = input(my_message)
print(my_name)
```

Consola:

Ingrese su Nombre:

El programa esperará a que escribamos algo y presionemos Enter para continuar. Inmediatamente el programa asignará lo ingresado a la variable my_name y lo imprimirá.

Ingrese su Nombre: Santiago
Santiago

Conversión de tipos de dato

Frecuentemente se hallará que es necesario el cambio de un tipo de dato para algún fin. En el actual contexto, se puede aplicar al caso de que todo lo recibido a través de consola con input() es un tipo de dato 'str' y habrá ocasiones en las que se necesite recibir un entero o float.

a. type()

Antes de pasar a convertir datos, primero se debe saber qué tipo de dato se tiene a la mano. Para esta tarea se puede utilizar la función type, dentro de los paréntesis se escribe un dato y la función devolverá el tipo de dato al que corresponde el dato que ingresamos. Posteriormente podemos imprimir el tipo de dato utilizando la función print().

```
my_data = 18
my_data_type = type(my_data)
```

```
print(my_data_type)
```

Consola:

```
<class 'int'>
```

b. str()

La función str() permite convertir un dato a tipo string con solo poner el dato dentro de los paréntesis.

```
_int = 18
print(type(_int))
_int_to_str = str(my_age)
print(type(_int_to_str))
```

(Nótese que se pueden tener funciones dentro de otras funciones)

Consola:

```
<class 'int'>
<class 'str'>
```

c. int()

La función int() permite convertir un dato a tipo int con solo poner el dato dentro de los paréntesis.

```
str_to_int = int('32')
print(type(str_to_int))
```

Consola:

```
<class 'int'>
```

Si lo utilizamos con un número 'float', se deshará de la parte decimal.

```
float_to_int = int(2.5)
print(float_to_int)
```

Consola:

```
2
```

d. float()

La función float() permite convertir un dato a tipo float con solo poner el dato dentro de los paréntesis.

```
str_to_float = float("3.82")
print(type(str_to_float))
```

Consola:

```
<class 'float'>
```

e. bool()

La función bool() permite convertir un dato a tipo booleano con solo poner el dato dentro de los paréntesis. Para esto se debe tener una consideración especial para cada uno de los tipos de datos.

i. int y float

Para los tipos de datos numéricos, un número 0 convertido a bool es un False, y cualquier otro valor es True.

```
print(bool(0))
print(bool(1))
```

Consola:

```
False
True
```

ii. string

Para los strings, un string vacío corresponde a False, y cualquier otro valor corresponde a True.

```
print(bool(""))
print(bool("Hola Mundo"))
```

Consola:

```
False
True
```

Glosario:

Capitalizada: del idioma inglés, se refiere a escribir una palabra con la primera letra en mayúscula.

Concatenación: unión de múltiples datos de tipo cadena de texto o string.

Convención: acuerdo formal de un conjunto de normas y estándares en la práctica de algo.

Caracteres: unidad informática que corresponde a un símbolo de un lenguaje natural (letra, número, jeroglífico, etc.).

Caracteres especiales: símbolo que no está directamente relacionado con un lenguaje natural, por ejemplo <, >, =, ¿, ¡, etc.

Declara: creación de una variable en una posición de la memoria del computador.

Identificador: nombre con el cual se identifica la posición de memoria de una variable.

Sintaxis literal: escritura de una sintaxis sin la necesidad de usarla en algo.