

# Introducción a python

Luis, FirstName1 `luis.last1@gmail.com`  
Carlos Rodríguez `carlosrdz.isd@gmail.com`

February 28, 2020

# Outline

- 1 Comenzando el viaje a al programación
- 2 Aterrizando a python
- 3 Tipos de datos (otra vez)
- 4 Interactuando con el usuario
- 5 Tipos de datos (última continuación)
- 6 Control de flujo y condicionales
- 7 Funciones
- 8 I/O
- 9 Test de cuadrado

# Constantes y variables

# Constantes y variables

- Una constante es un valor que se mantiene fijo durante todo el programa. Ejemplos: 5, 'a', 38, "www.google.com"
- Una variable es un espacio de almacenamiento en memoria que contiene cierta información, la cuál puede ser modificada durante la ejecución del programa. Ejemplos: variable, a,  $a5 = 32$ , var = "hola"

# Tipos de datos

# Tipos de datos

- Entero (integer)
- Flotante (float)
- Número complejo
- Booleano (boolean)
- String
- Lista
- Tupla
- Diccionario

# Operadores aritméticos

# Operadores aritméticos

- $+$  (suma)
- $-$  (resta)
- $*$  (multiplicación)
- $/$  (división)
- $\%$  (módulo)
- $**$  (exponente)
- $//$  función suelo



# Operadores lógicos

# Operadores lógicos

- $>$  (mayor)
- $<$  (menor)
- $==$  (igual a)
- $!=$  (diferente de)
- $>=$  (mayor o igual)
- $<=$  (menor o igual)

# Operadores lógicos

- and (y)
- or (o)
- not
- y otros...

# Outline

- 1 Comenzando el viaje a al programación
- 2 Aterrizando a python
- 3 Tipos de datos (otra vez)
- 4 Interactuando con el usuario
- 5 Tipos de datos (última continuación)
- 6 Control de flujo y condicionales
- 7 Funciones
- 8 I/O
- 9 Test de cuadrito

# Hola python!

# Hola python!

- # En este programa desplegamos un saludo
- `print("Hola grupo de python!")`

# Comentarios

# Comentarios

- Los comentarios sirven para hacer más entendible el código, tanto para nosotros como para los demás
- Python sabe que es un comentario al escribir el hashtag al inicio de la línea
- # Así por ejemplo
- Otra forma de escribir un comentario es con triple comilla doble al inicio y al final del comentario
- """Este es otro ejemplo de
- comentarios"""



# Outline

- 1 Comenzando el viaje a la programación
- 2 Aterrizando a python
- 3 Tipos de datos (otra vez)**
- 4 Interactuando con el usuario
- 5 Tipos de datos (última continuación)
- 6 Control de flujo y condicionales
- 7 Funciones
- 8 I/O
- 9 Test de cuadrito

# Especificando a python

# Especificando a python

- Para que python sepa qué tipo de dato deseamos utilizar, es necesario realizar un "cast" al tipo de dato. En caso de no especificar el tipo de dato que deseamos, python otorgará el que el crea más apropiado a la variable, por ejemplo:
- `str(texto)`
- `int(numero)`
- `float(texto)`

# Especificando a python

- Para que python sepa qué tipo de dato deseamos utilizar, es necesario realizar un "cast" al tipo de dato. En caso de no especificar el tipo de dato que deseamos, python otorgará el que el crea más apropiado a la variable, por ejemplo:
- `str(texto)`
- `int(numero)`
- `float(texto)` #esto es válido ;)

# Números

# Números

- Los dos tipos principales de números en python son los enteros (integers) y reales (floats)
- En caso de no especificar el tipo de número (o de dato) python decide cuál es el más apropiado (a veces se equivoca)
- Integers (int): 2, 5, 4000, -2
- Floats (float): 2.5, 2.9, -3.1

# Texto

# Texto

- Una variable que almacena texto se llama cadena o string, y para que python las reconozca deben estar en comillas simples o dobles... o triples si necesita varias líneas Ejemplo: "Texto de ejemplo"
- el operador + une dos strings...



# Outline

- 1 Comenzando el viaje a la programación
- 2 Aterrizando a python
- 3 Tipos de datos (otra vez)
- 4 Interactuando con el usuario**
- 5 Tipos de datos (última continuación)
- 6 Control de flujo y condicionales
- 7 Funciones
- 8 I/O
- 9 Test de cuadrito

input y print

## input y print

- Para desplegar algún dato de nuestro programa, basta con utilizar el comando print, mientras que para asignar algún valor dado por el usuario a una variable, se utiliza el comando input. Ejemplos:
- `print("Desplegando un texto")`
- `var=input("Ingresa un texto")`
- `print(var)`

# Outline

- 1 Comenzando el viaje a al programación
- 2 Aterrizando a python
- 3 Tipos de datos (otra vez)
- 4 Interactuando con el usuario
- 5 Tipos de datos (última continuación)**
- 6 Control de flujo y condicionales
- 7 Funciones
- 8 I/O
- 9 Test de cuadrito

# Arreglos (colecciones)

# Arreglos (colecciones)

- Existen 4 tipos de arreglos en python (legacy), los cuáles permiten almacenar varios datos en una sola variable:
- Listas (List): Ordenada e intercambiable. Permite miembros duplicados.
- Tuplas (Tuple): Coleccion ordenada y no intercambiable. Permite miembros duplicados.
- Conjuntos (Set): Desordenada y no indexada. No permite miembros duplicados.
- Diccionarios (Dictionary): Desordenada, intercambiable e indexada. No permite miembros duplicados.

# Listas

# Listas

- La lista es una colección ordenada e intercambiable.
- Para crear una lista en python se escriben sus elementos entre corchetes [a, b, c, "hola"].
- `mi_lista=["un_elemento","otro_elemento","tercer_elemento",  
"otro", "otro", "ya", "son", "muchos"]`
- `print(mi_lista)`
- Para acceder a una lista, especificamos el número de indexación al que queremos acceder (el primer elemento se indentifica con el número 0).
- `print(mi_lista[1])`



# Listas

# Listas

- python permite indexamiento negativo...

# Listas

- python permite indexamiento negativo... no lo usen )=
- Es posible acceder a una sublista de un rango especificado
- `print(mi_lista[2:5])`
- `print(:5)`

# Operaciones de listas

# Operaciones de listas

- Las operaciones más habituales de Python son las siguientes:
- `lista[i]`: Devuelve el elemento que está en la posición `i` de la lista.
- `lista.pop(i)`: Devuelve el elemento en la posición `i` de una lista y luego lo borra.
- `lista.append(elemento)`: Añade elemento al final de la lista.
- `lista.insert(i, elemento)`: Inserta elemento en la posición `i`.
- `lista.extend(lista2)`: Fusiona lista con `lista2`.
- `lista.remove(elemento)`: Elimina la primera vez que aparece elemento.

# Diccionarios

# Diccionarios

- Listas
- Diccionarios
- Conjuntos (sets)
- Tuplas

# Outline

- 1 Comenzando el viaje a la programación
- 2 Aterrizando a python
- 3 Tipos de datos (otra vez)
- 4 Interactuando con el usuario
- 5 Tipos de datos (última continuación)
- 6 Control de flujo y condicionales**
- 7 Funciones
- 8 I/O
- 9 Test de cuadrito



# Control de flujo y condicionales

# Control de flujo y condicionales

- Decimos hola

# Outline

- 1 Comenzando el viaje a al programación
- 2 Aterrizando a python
- 3 Tipos de datos (otra vez)
- 4 Interactuando con el usuario
- 5 Tipos de datos (última continuación)
- 6 Control de flujo y condicionales
- 7 Funciones**
- 8 I/O
- 9 Test de cuadrito

# Scope

# Scope

- Decimos hola

# Llamando una función

# Llamando una función

- Decimos hola

# Declaración de funciones



# Declaración de funciones

- Decimos hola

# Outline

- 1 Comenzando el viaje a al programación
- 2 Aterrizando a python
- 3 Tipos de datos (otra vez)
- 4 Interactuando con el usuario
- 5 Tipos de datos (última continuación)
- 6 Control de flujo y condicionales
- 7 Funciones
- 8 I/O
- 9 Test de cuadrito

# Interactuando con el archivo

# Interactuando con el archivo

- Decimos hola

# Interactuando con la propia computadora

# Interactuando con la propia computadora

- Decimos hola

# Outline

- 1 Comenzando el viaje a al programación
- 2 Aterrizando a python
- 3 Tipos de datos (otra vez)
- 4 Interactuando con el usuario
- 5 Tipos de datos (última continuación)
- 6 Control de flujo y condicionales
- 7 Funciones
- 8 I/O
- 9 Test de cuadrito

# frame con cuadrato

- esta presentacion tiene un cuadrato



# frame con cuadrato

- esta presentación tiene un cuadrato

## Open Definition

[https://www.tutorialspoint.com/python/python\\_basic\\_operations.htm](https://www.tutorialspoint.com/python/python_basic_operations.htm)

<https://www.tutorialsteacher.com/python/python-comparison-operators/>

<https://www.learnpython.org/es/Hello,%20World!>

[https://www.w3schools.com/python/python\\_lists.asp](https://www.w3schools.com/python/python_lists.asp)