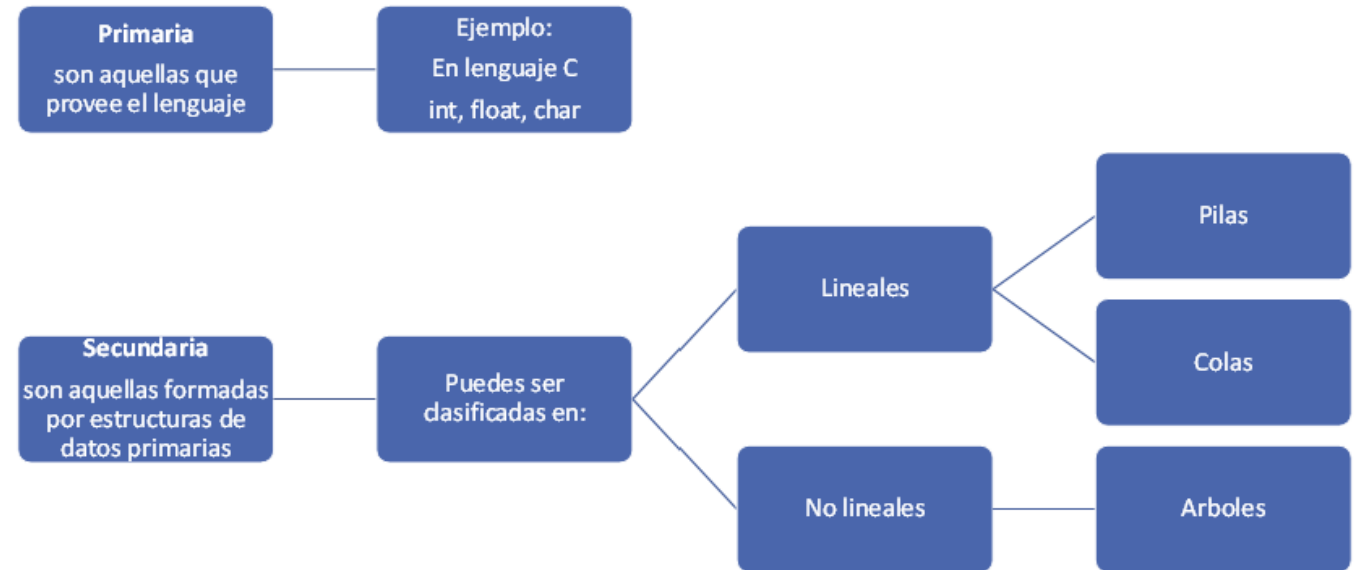


Estructuras de Datos en Programación

Importancia de las Estructuras de Datos

- Esencialidad de las Estructuras de Datos
 - Cruciales para la organización y acceso a datos en proyectos.
- Definición y Aplicación
 - Parte integral de la programación conocida como estructuras de datos.
- Beneficios del Conocimiento
 - Contribuye a la eficiencia y efectividad de los programas.
 - Facilita la resolución de problemas diversos.



Clasificación de las Estructuras de Datos: String

str



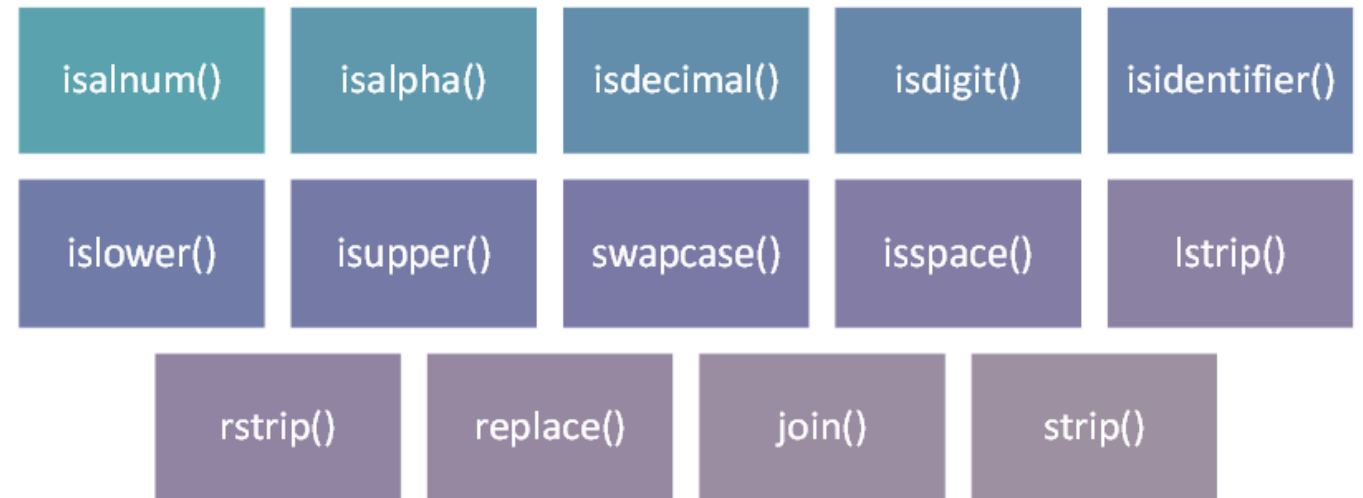
Strings en Python

- Las cadenas **str** son una colección de caracteres en Python.
- En Python, las cadenas **str** son inmutables, lo que significa que una vez establecidas, el valor de una cadena no se puede modificar.
- Python ofrece muchas funciones integradas para manipular cadenas.



Características de los Strings en Python

- Objeto Predefinido de Caracteres
 - Una cadena es una colección de caracteres en Python.
- Inmutabilidad de las Cadenas
 - Una vez establecido, el valor de una cadena no puede modificarse.
- Funciones Asociadas
 - Existen diversas funciones para trabajar con cadenas en Python.



Clasificación de las Estructuras de Datos: Listas

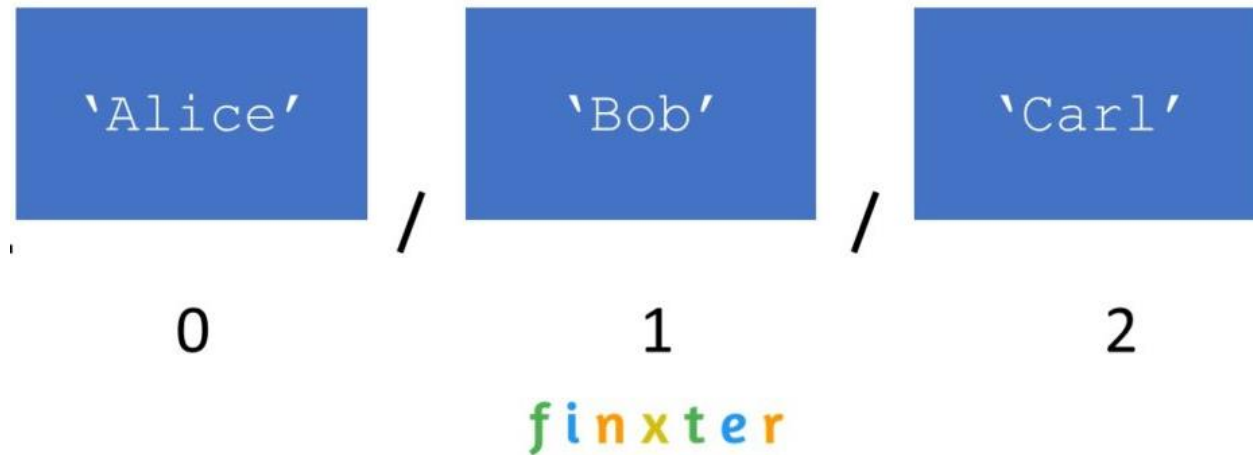
list



Listas en Python



Python Lists



- Una lista es una estructura de datos que permite almacenar una colección de elementos
- Las listas son mutables y pueden contener elementos de diferentes tipos
- Las listas se definen usando corchetes y los elementos se separan por comas

Funciones de la clase list



La clase list en Python tiene una variedad de funciones incorporadas.



`append()`, `insert()`, y `extend()` son algunas de las funciones más utilizadas.



Otras funciones útiles incluyen `index()`, `sort()`, y `reverse()`.

Clasificación de las Estructuras de Datos: Tuplas

tuple



Clasificación de las Estructuras de Datos: Tuplas

Las tuplas son una colección de elementos accesibles individualmente o en grupo

En Python, las tuplas son inmutables, lo que significa que una vez establecidas, el valor de una tupla no se puede modificar

Las tuplas son heterogéneas, lo que significa que pueden contener elementos de diferentes tipos

Características de las Tuplas en Python

Definición de Tupla

- Una colección de elementos accesibles individualmente o en grupo

Impresión de Tuplas

- Utilización de la función `print()` para mostrar tuplas

Elementos en Tuplas

- Admite heterogeneidad y elementos duplicados

Operaciones con Tuplas

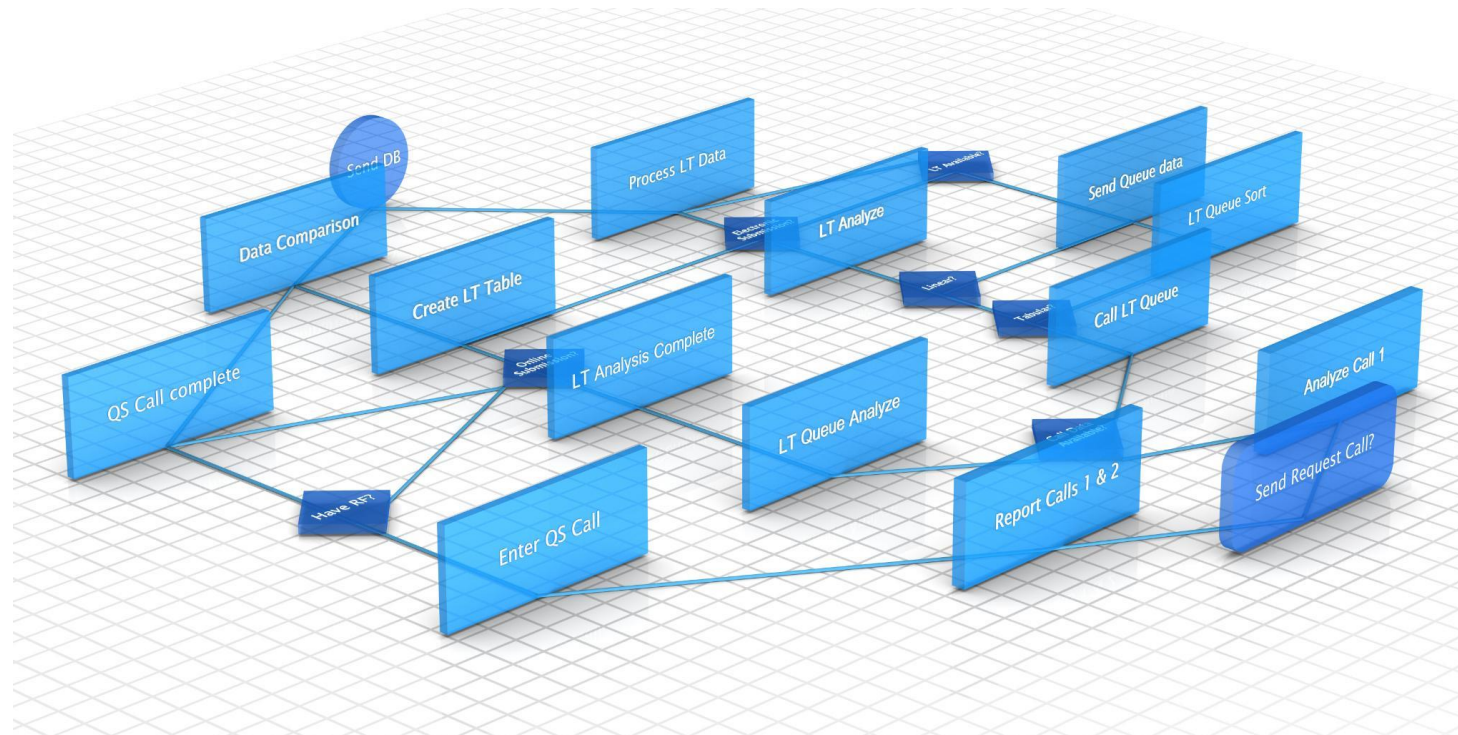
- Permite concatenación pero no la edición de elementos

Inmutabilidad de Tuplas

- Los elementos no pueden ser reasignados a otros valores

Métodos de la Clase Tuple

- La clase tuple en Python tiene varios métodos incorporados
- Los métodos `count()` e `index()` son algunos de los métodos más utilizados
- Otros métodos útiles incluyen `len()`, `max()`, `min()` y `sum()`

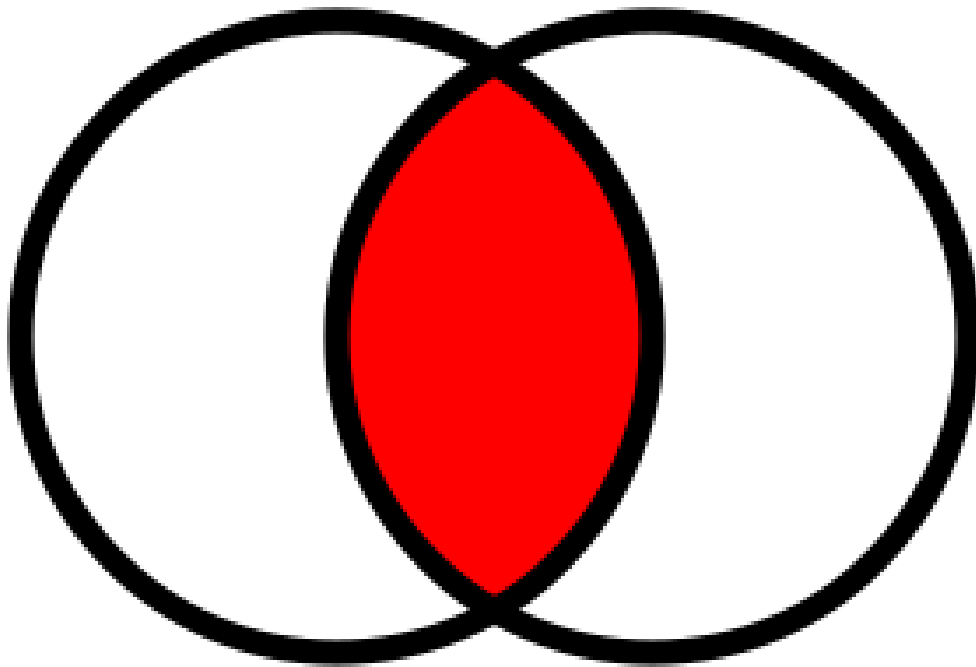


Clasificación de las Estructuras de Datos: Conjuntos

set



Conuntos en Python



This Photo by Unknown Author is licensed under [CC BY-SA](#)

- Un conjunto es una colección de elementos únicos
- Los elementos de un conjunto están desordenados
- Los conjuntos en Python se pueden crear usando llaves o el constructor `set()`

Características de los Conjuntos en Python

No admiten elementos repetidos

Inmutabilidad de los Elementos

- Similitud con las Tuplas
- No se pueden cambiar, solo agregar o eliminar con función

Impresión de Conjuntos

- Utilización de la función `print()` estándar

Contenido Heterogéneo

- Posibilidad de incluir diferentes tipos de elementos

Operaciones con Conjuntos

- Permite la concatenación, no la edición de elementos

Métodos de la clase Set en Python



La clase Set en Python tiene una variedad de funciones incorporadas.



Algunas de las funciones más utilizadas son `add()`, `remove()` y `clear()`.



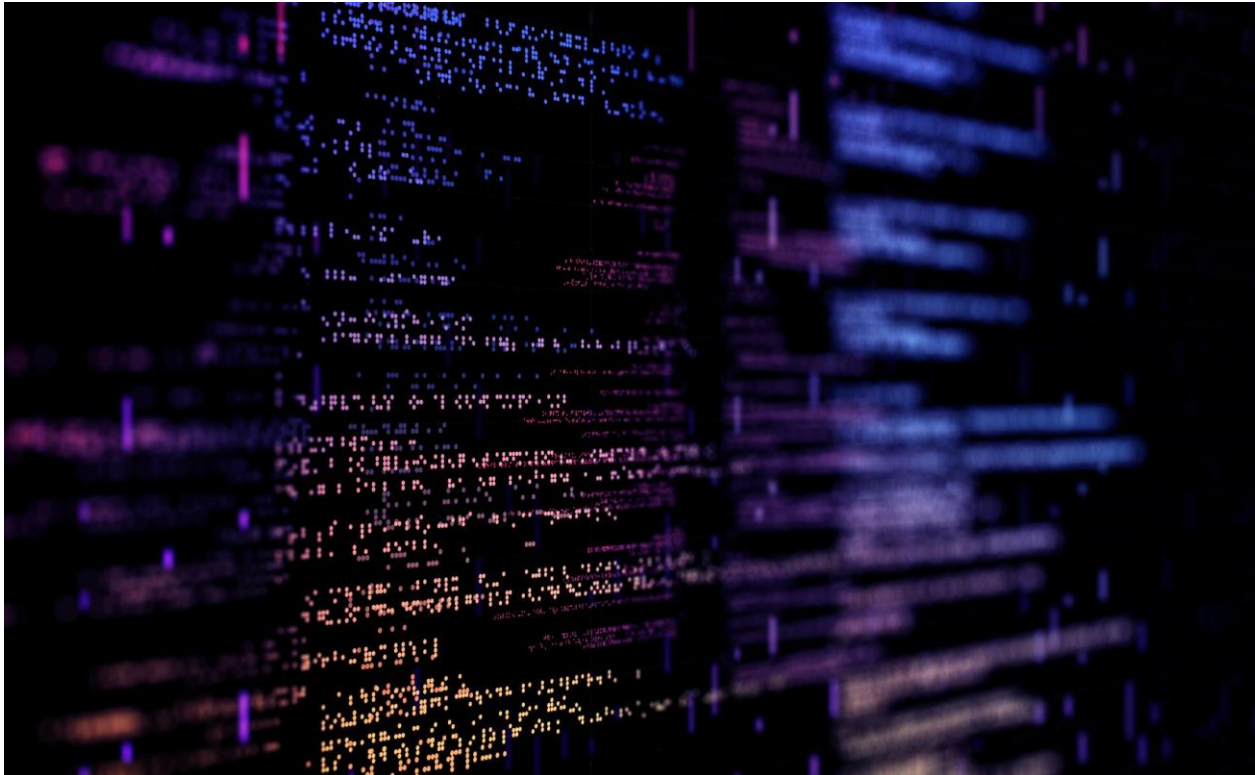
Otras funciones útiles incluyen `union()`, `intersection()` y `difference()`.

Clasificación de las Estructuras de Datos: Diccionarios

dict



Diccionarios en Python



- Los diccionarios son una colección de pares clave-valor en Python.
- Las claves son objetos únicos e inmutables, mientras que los valores pueden ser cualquier objeto.
- Los diccionarios son mutables y pueden modificarse según sea necesario.

Características de los Dicionarios en Python

- Sintaxis de Dicionarios
 - Creación con llaves y valores
 - Acceso y modificación de valores
- Operaciones Comunes
 - Agregar y cambiar valores
 - Eliminar valores con o sin retorno
- Características Clave
 - Claves únicas y mutabilidad
 - Indexación de elementos

```
mirror_mod = modifier_ob.  
set mirror object to mirror.  
mirror_mod.mirror_object =  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True  
  
selection at the end -add  
mirror_ob.select= 1  
mirror_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob.  
mirror_ob.select = 0  
= bpy.context.selected_object  
data.objects[one.name].select  
  
print("please select exactly  
  
-- OPERATOR CLASSES ----  
  
types.Operator):  
X mirror to the selected  
object.mirror_mirror_x"  
mirror X"  
  
context):  
context.active_object is not
```

Métodos de la Clase Dict en Python

Los métodos incorporados de la clase dict son útiles para trabajar con diccionarios

El método `.keys()` devuelve una lista con todas las claves del diccionario.

El método `.values()` devuelve una lista con todos los valores del diccionario.

El método `.items()` devuelve una lista de tuplas de pares clave-valor del diccionario.

El método `.get()` devuelve el valor de la clave dada, o un valor predeterminado si la clave no está en el diccionario.



Clasificación de las Estructuras de Datos: Estructuras Anidadas

Estructuras anidadas

Estructuras Anidadas en Programación

Creación de Estructuras Anidadas

- Capacidad para contener datos heterogéneos
- Flexibilidad en la organización de datos

Tipos de Estructuras de Datos

- Listas, tuplas y conjuntos
- Diccionarios y combinaciones

Aplicaciones de Anidación

- Mejora en la gestión de datos complejos
- Facilita la representación de relaciones entre datos

Comparativa entre Estructuras de Datos

Comparativa				
	Listas	Strings	Tuplas	Conjuntos
Palabra clave	list	str	tuple	set
Delimitadores	[]	" o ""	() pero ooo si solo es un elemento (ele,)	{}
Tipo de datos	Cualquier tipo: int, float, str, list, etc.	Char (caracteres) - símbolo ASCII	Cualquier tipo: int, float, str, list, etc.	Cualquier tipo: int, float, str, list, etc.
	Homogeneas o heterogeas		Homogeneas o heterogeas	Homogeneas o heterogeas
Mutable? Modificar elemento individual	Si	No	No	No
Indexada?	Si	Si	Si	No
Agregar	Si	concatenar	concatenar	Si
Eliminar	Si	No	No	Si
Elementos repetidos?	Si	Si	Si	No
Referencia	https://www.w3schools.com/python/python_lists.asp	https://www.w3schools.com/python/python_string.asp	https://www.w3schools.com/python/python_tuple.asp	https://www.w3schools.com/python/python_sets.asp