Repaso: “Tipos y Estructuras”

INSTITUTO DE INNOVACIÓN Y TECNOLOGÍA APLICADA

IITA

Contenido

I I T A

Profesor

Pablo Sandoval

Soy Ingeniero y Técnico en Informática, graduado de la **UCASAL**. Actualmente me desempeño como desarrollador **Backend,** especializado en la creación de soluciones innovadoras utilizando **NestJS y FastAPI**. Mi enfoque incluye el desarrollo de sistemas avanzados integrados con Inteligencia Artificial, aportando valor mediante tecnología de vanguardia y soluciones eficientes.

¡Bienvenidos al curso de programación! 🎉

Sobre el curso...

Para poder obtener el certificado de fin de curso, hay ciertas condiciones que deberán cumplir:

- 80% de **asistencia** en el curso.
- Entregar **todos** los trabajos prácticos. (Acepto solo 1 entrega fuera de término).
- Realización de un **Proyecto Final**. El tema es a libre elección, pero debe hacer uso de los conceptos aprendidos en clase.
- La dinamica del curso sera tanto teorica/ practica tomando preguntas del mundo laboral como disparadores.

Horarios:

Sábados: 9:00 a 11:00 AM

Preguntas frecuentes

Sobre el curso...

¿ Cuanto Dura el curso?

- Dura 3 meses con un total de aproximadamente **12 clases** (abordando un total de 8 apartados)

Nos prepara para el mundo laboral?

- Este curso refuerza los conocimientos previos y muestra cómo Python se utiliza en campos como desarrollo web, análisis de datos e inteligencia artificial. Ayuda a entender su aplicación en problemas reales, acercándolos a los desafíos del mundo laboral.

Podemos usar chatgpt o googlear los ejercicios?

PUEDEN buscar recursos que los ayuden a comprender y resolver los ejercicios. **SIN EMBARGO,** tienen que comprender que estan haciendo porque despues seran evaluados por ellos.

Preguntas frecuentes

Sobre el curso...

¿Conocimientos previos?

- Este curso **SI** requiere de conocimientos previos en programación

Como se entregan los trabajos practicos?

- Tienen una semana para realizarlos y seran entregados por la plataforma.

Necesito una super computadora?

- NO, solo con unos 8gb de RAM, paciencia y muchas ganas todo se puede.

Modalidad del curso?

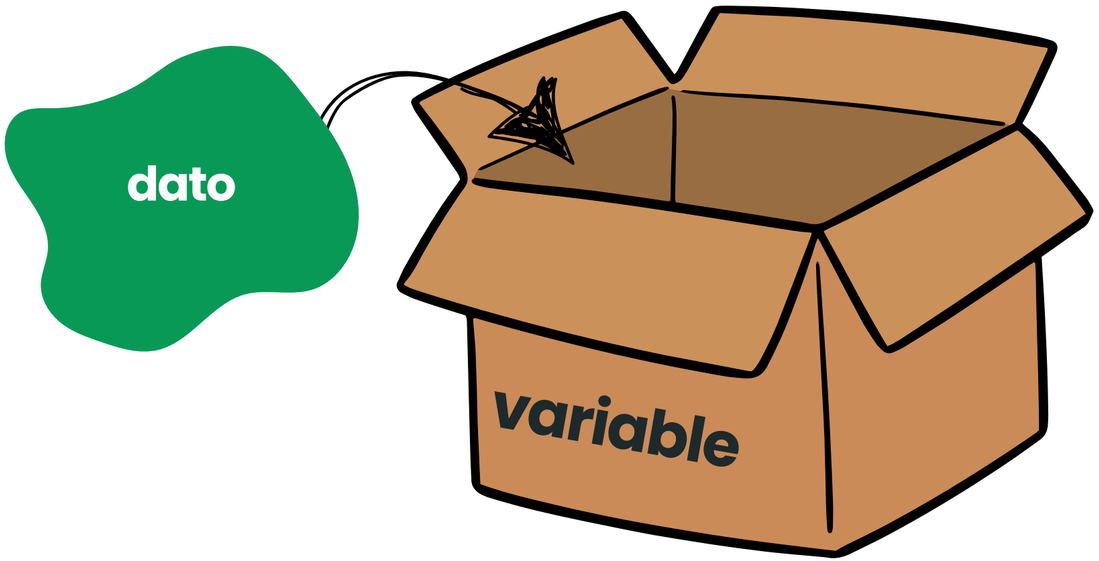
- Es 100% **virtual**

Donde puedo hacer consultas?

- Puedes hacer consultas por la plataforma o por el grupo de whatsapp

¿Qué es una Variable?

BREVE REPASO

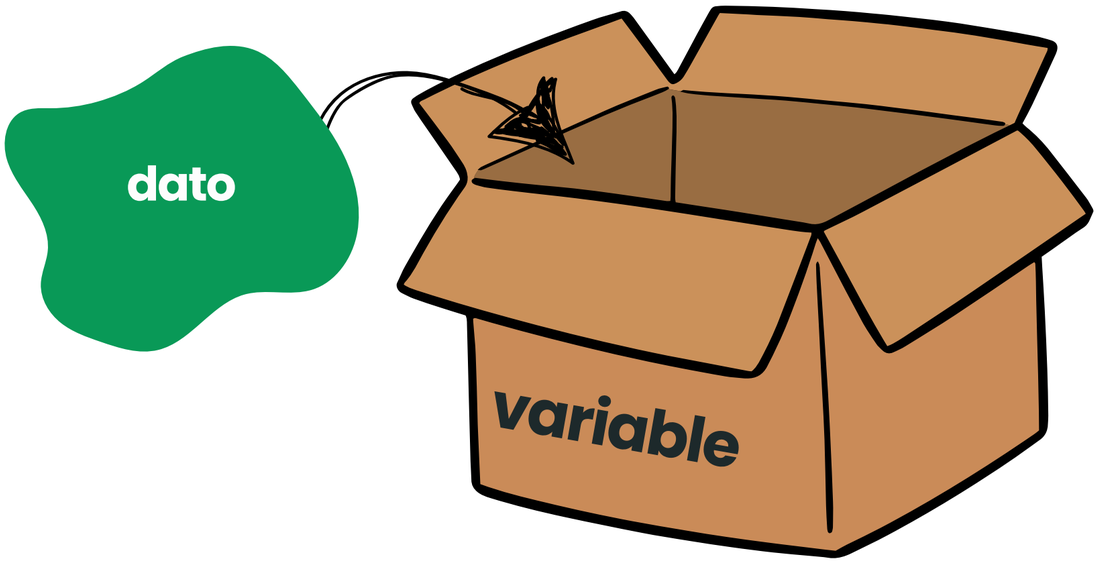


Es un **lugar en la memoria** de la computadora donde guardamos **información**, como un **número o una palabra.** Cada variable tiene un **nombre** que usamos para identificarla y el **valor** que guarda puede cambiar.

¿Qué es un Dato?

PREGUNTA DE ENTREVISTA

BREVE REPASO

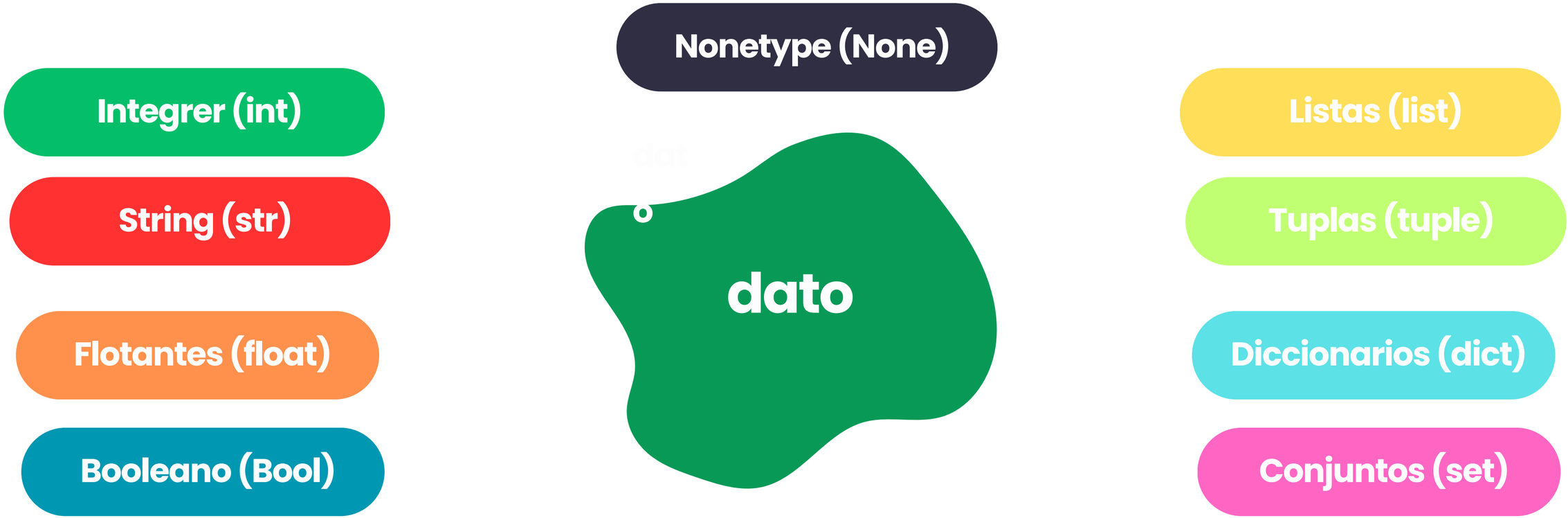


Es **cualquier tipo de información** que usamos en un programa.

Puede ser un **número, una palabra o algo más**.

Vamos a profundizar más eso...

IDENTIDAD, TIPOS Y VALOR

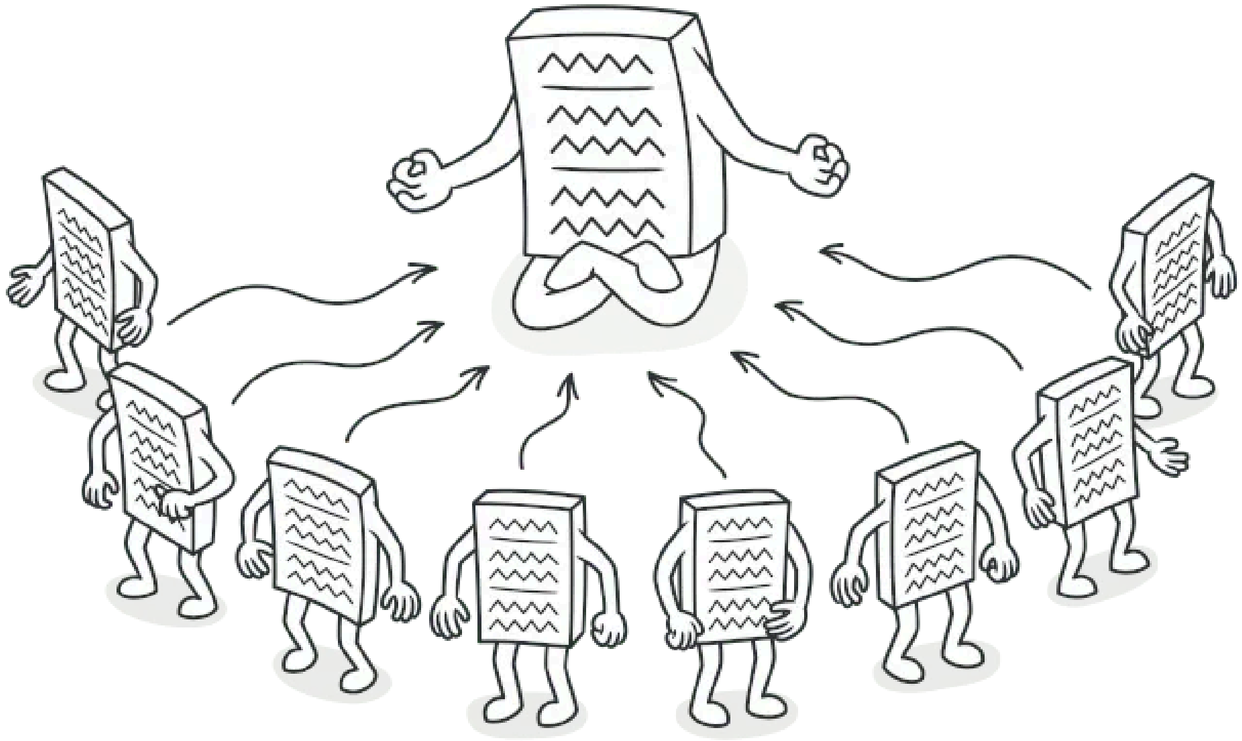


En **python**, todos los tipos de datos se tratan como **objetos**. Desde un simple entero hasta una función

IDENTIDAD, TIPOS Y VALOR

Identidad

El “número de serie” de nuestra caja



Valor

Es lo que hay dentro de la cajita

Tipo

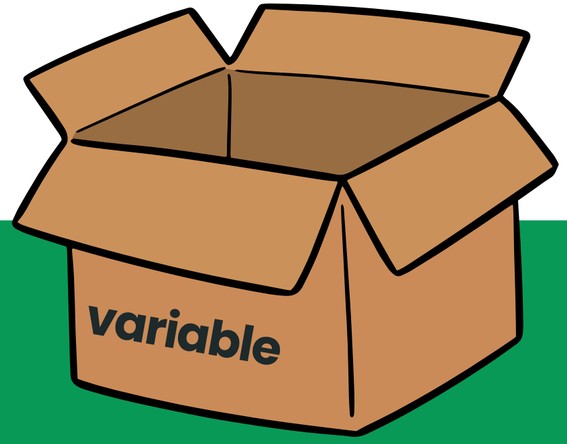
¿Qué tipo de cosa está guardando la cajita?

Veamoslo en código...

¿Qué pasa si intento cambiar el objeto dentro de la caja?

PREGUNTA DE ENTREVISTA

MUTABILIDAD



Cajitas mutables

Son objetos cuyo contenido puede modificarse sin cambiar su identidad (su dirección en memoria sigue siendo la misma).

list, dict,set

VS

Cajitas inmutables

Una vez creados, sus valores no pueden cambiar. Si queremos modificarlos, Python crea una nueva caja con el nuevo valor y reasigna la referencia.

int, str, float

Veamoslo en código...

¿Cuál es la diferencia entre tipos de datos primitivos y no primitivos?

PREGUNTA DE ENTREVISTA

PRIMITIVOS Y NO PRIMITIVOS

- Integrer (int)
- String (str)
- Flotantes (float)
- Booleano (Bool)
- Nonetype (None)

- Listas (list)
- Tuplas (tuple)
- Diccionarios (dict)
- Conjuntos (set)

En **python**, todos los tipos de datos **primitivos** son los mas basicos, mientras que los **no primitivos** son los más complejos

¿Cuál es la diferencia entre paso por referencia y por valor?

PREGUNTA DE ENTREVISTA

POR REFERENCIA VS POR VALOR

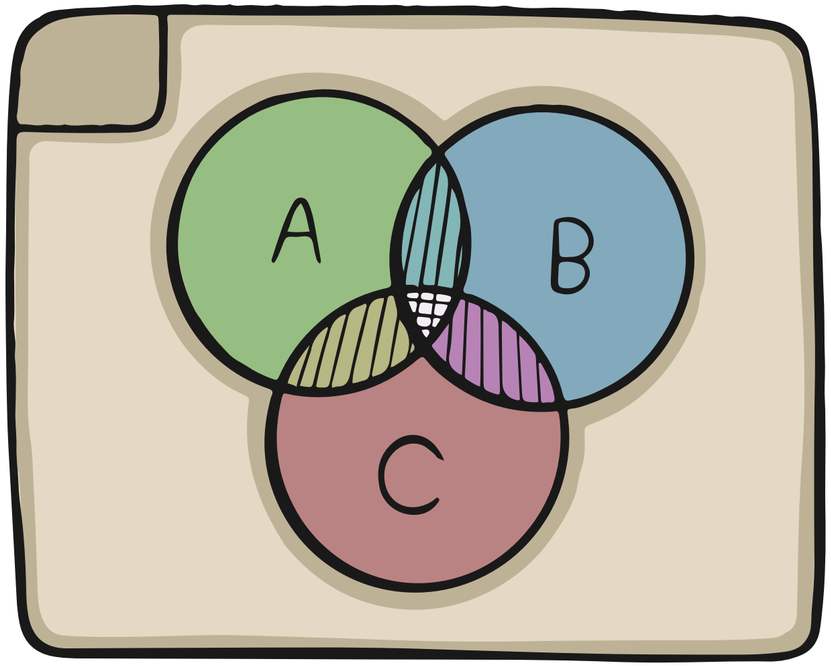
En **Python**, los argumentos para una función de **tipo inmutable**, se pasan **por valor**, lo que significa que cualquier modificación crea un nuevo objeto sin afectar el original. En cambio, los **tipos mutables** se pasan **por referencia**, por lo que los cambios realizados dentro de una función afectan directamente al objeto original.

Veamoslo en código...

¿Qué es un set en python?

PREGUNTA DE ENTREVISTA

CONJUNTOS (SETS)



Un **conjunto** es una colección de **elementos únicos**, sin **ningún orden específico y sin duplicados**. Es una idea matemática fundamental que se utiliza para agrupar cosas que tienen alguna propiedad en común.

CONJUNTOS (SETS)

```python
# Crear un conjunto
s = {1, 2, 3, 4, 5}
print(s)  # {1, 2, 3, 4, 5}

# Crear un conjunto con elementos duplicados
s_con_duplicados = {1, 2, 2, 3, 4}
print(s_con_duplicados)  # {1, 2, 3, 4} (los duplicados se eliminan)
```

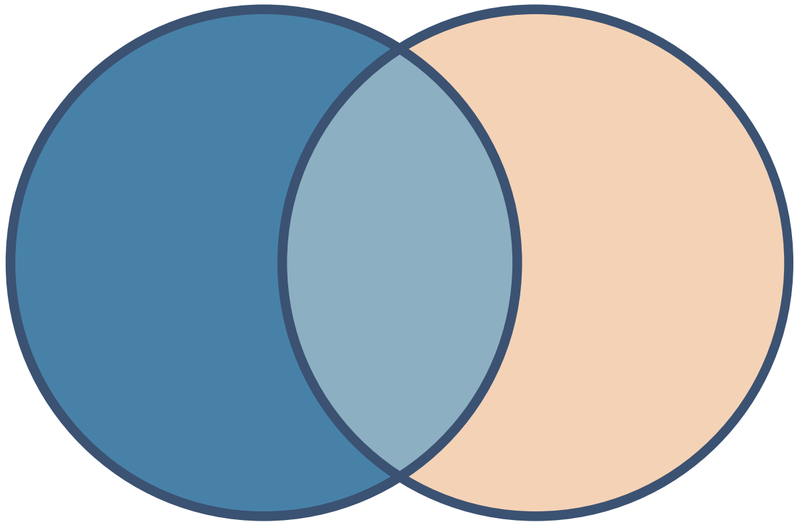
Un **conjunto en Python** se define utilizando llaves {} y puede contener **cualquier tipo de datos inmutables**, como números, cadenas o tuplas. **No pueden contener listas ni diccionarios**, ya que estos son mutables. Si intentamos agregar elementos duplicados, solo se conservará uno.

```python
mi_set = {1, 2, 3}
print(mi_set)  # {1, 2, 3}

# añadir un elemento
mi_set.add(4)
print(mi_set)  # {1, 2, 3, 4}


# eliminar un elemento
mi_set.remove(4)
print(mi_set)  # {1, 2, 3}

# remover un elemento (si no existe no hace nada)
mi_set.discard(4)
print(mi_set)  # {1, 2, 3}


# extraer un elemento aleatorio
numero_extraido = mi_set.pop()
print(numero_extraido)  # 3


# vaciar el set
mi_set.clear()
print(mi_set)  # set()
```

Operaciones con sets

```python
mi_set = {1, 2, 3}
print(mi_set)  # {1, 2, 3}

# añadir un elemento
mi_set.add(4)
print(mi_set)  # {1, 2, 3, 4}


# eliminar un elemento
mi_set.remove(4)
print(mi_set)  # {1, 2, 3}

# remover un elemento (si no existe no hace nada)
mi_set.discard(4)
print(mi_set)  # {1, 2, 3}


# extraer un elemento aleatorio
numero_extraido = mi_set.pop()
print(numero_extraido)  # 3


# vaciar el set
mi_set.clear()
print(mi_set)  # set()
```

Operaciones con sets



Union entre sets

```python
set_uno = {1, 2, 3}
set_dos = {3, 4, 5}

# union
print(set_uno | set_dos)  # {1, 2, 3, 4, 5}
```

Intersección entre sets

```python
# intersección
print(set_uno & set_dos)  # {3}
```

Diferencias entre sets

```python
# diferencia A - B
print(set_uno - set_dos)  # {1, 2}

# diferencia B - A
print(set_dos - set_uno)  # {4, 5}

# diferencia simétrica
print(set_uno ^ set_dos)  # {1, 2, 4, 5}
```

IITA

¡Muchas Gracias!

Por su atención