
Repaso de tools

Git, Github y Python

Git

- Sistema de control de versiones distribuido.
- Permite mantener un registro de cambios en el código fuente de un proyecto, lo que facilita la **colaboración**, la **reversión** de cambios y la gestión del **versionado** del mismo proyecto.
- En lugar de tener una única copia del código fuente, Git almacena una serie de instantáneas llamadas "**commits**" que representan el *estado del proyecto en diferentes puntos de tiempo*.



Github

Plataforma en línea para alojar repositorios de control de versiones basados en Git.



GitHub

Github Desktop

Aplicación que proporciona una GUI para trabajar con repositorios Git en una computadora local.



Ejemplo de uso Git - Github - Github desktop

1. Creación de un repositorio en github
2. Crear un archivo desde el repositorio remoto
3. Clonar el repositorio remoto usando github desktop
4. Ir a la carpeta local del repositorio clonado
5. Modificar en local el archivo creado en 2.
6. Agregar un archivo en el directorio local
7. Crear una nueva branch llamada “dev” desde github desktop
8. Solicitar un PR

Tarea #2: Crear el repositorio del curso

1. Crear un repositorio en su cuenta de github llamado: istea-mineria-datos-i
2. Agregar un archivo “leeme” con un comentario relacionado al curso de minería de datos I.
3. Clonar el repositorio a nivel local
4. Crear una branch llamada “dev”
5. Enviar al profesor (por mail) el link al repositorio.

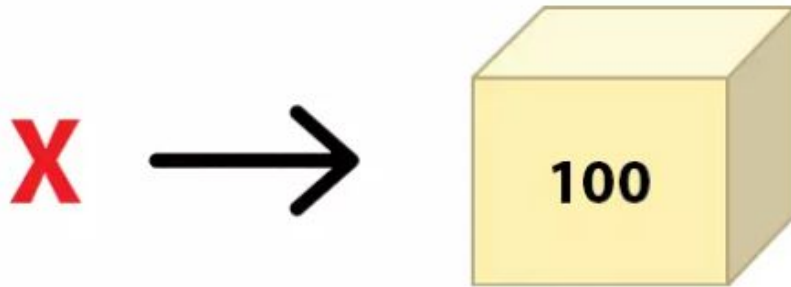
Python

Variables y tipos de datos

Variables

X = 100

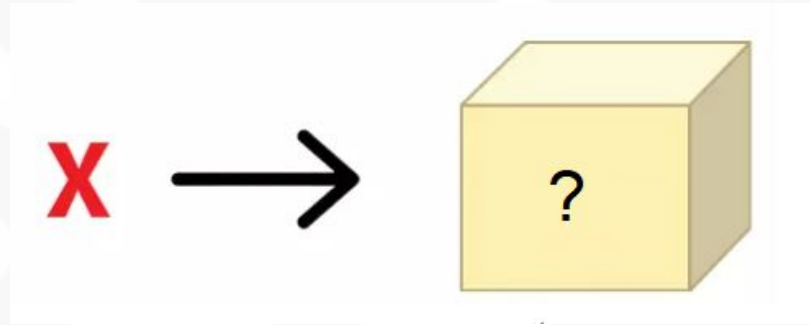
→ Instrucción de
asignación



Las variables son contenedores
para almacenar valores en la
memoria.

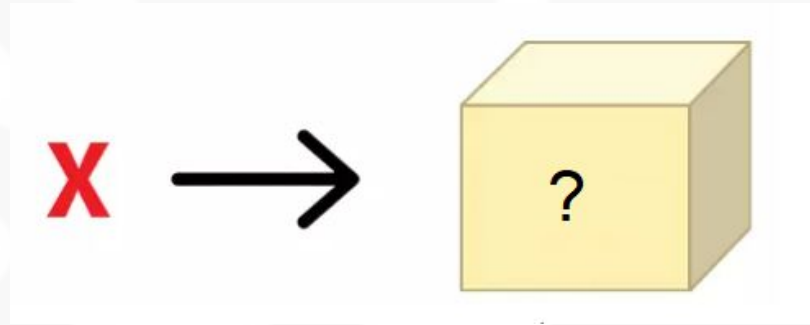
Variables

¿Cómo sabe el sistema de qué tamaño es el contenedor?



Variables

¿Cómo sabe el sistema de qué tamaño es el contenedor?



Gracias a los tipos de datos

Variables

Asignación encadenada

$a = b = c = 300$

Variables globales y variables locales

¿Cuál es la salida de estos dos bloques de código?

```
# Declarar una variable e inicializarla
f = 101
print(f)

# Variables globales vs variables locales en funciones
def algunaFuncion():
    f = 'Estoy repasando Python'
    print(f)

algunaFuncion()
print(f)
```

```
# Declarar una variable e inicializarla
f = 101
print(f)

# Variables globales vs variables locales en funciones
def algunaFuncion():
    global f
    print(f)
    f = 'Cambiando una variable global'

algunaFuncion()
print(f)
```

Variables globales y variables locales

¿Cuál es la salida de estos dos bloques de código?

```
# Declarar una variable e inicializarla
f = 101
print(f)

# Variables globales vs variables locales en funciones
def algunaFuncion():
    f = 'Estoy repasando Python'
    print(f)

algunaFuncion()
print(f)
```

```
101
Estoy repasando Python
101
```

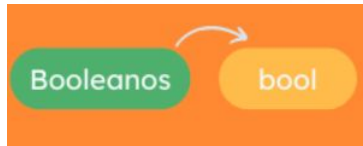
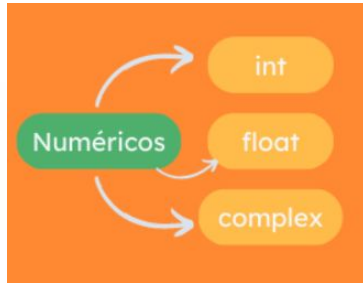
```
# Declarar una variable e inicializarla
f = 101
print(f)

# Variables globales vs variables locales en funciones
def algunaFuncion():
    global f
    print(f)
    f = 'Cambiando una variable global'

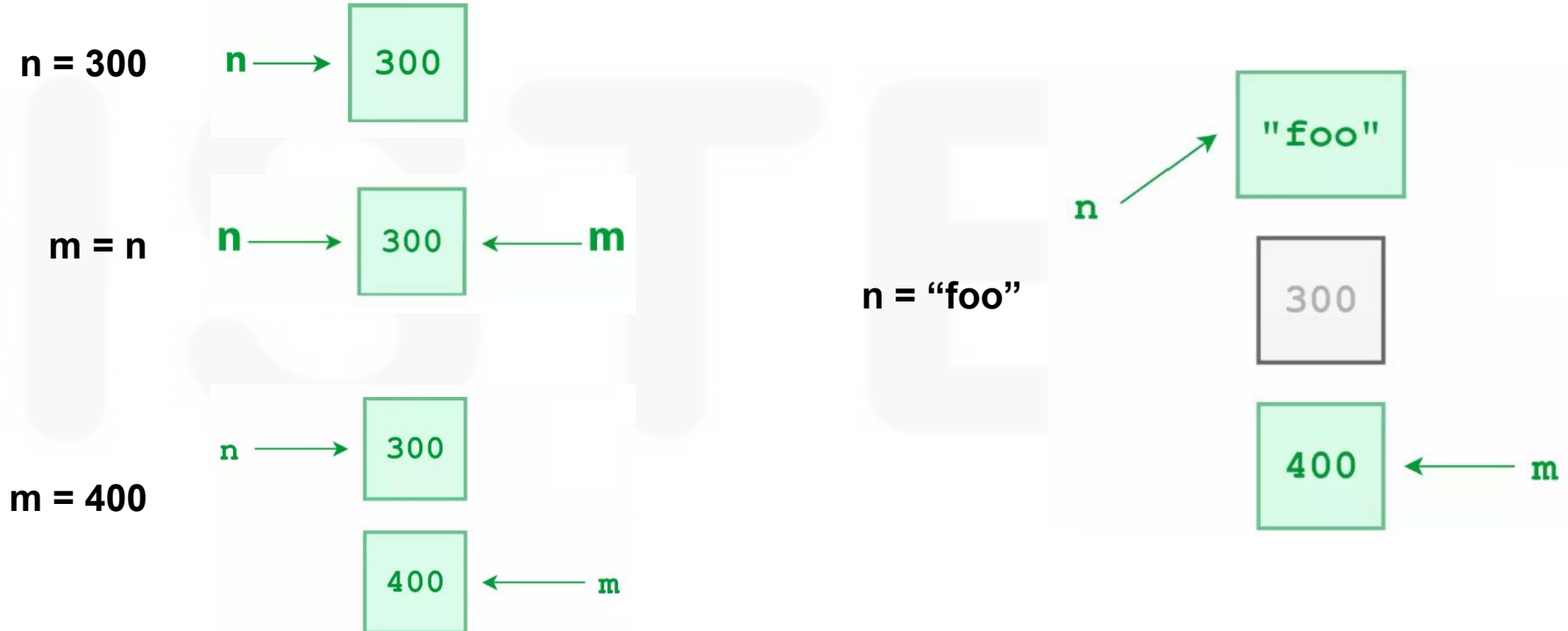
algunaFuncion()
print(f)
```

```
101
101
Cambiando una variable global
```

Tipos de datos



Referencias a objetos



Clases y objetos

Conceptos fundamentales

Clases

Utiliza la palabra clave **class** para definir una clase y sus atributos.

Atributos

Son las características que tienen los objetos.
Los atributos se acceden con la notación de punto (objeto.atributo)

Objetos

Crea objetos utilizando el constructor de la clase

Métodos

Son funciones que pueden ser invocadas en esos objetos para realizar acciones específicas.

Self

Primer parámetro de un método, que hace referencia a la instancia en sí.

Ejemplo

```
class Perro:
    def __init__(self, nombre):
        self.nombre = nombre

    def ladrar(self):
        print(f"{self.nombre} está ladrando")

perro1 = Perro("Max")
perro1.ladrar()
```

Tarea #3: Programa en Python (30 min)

Sistema de Recomendación de Libros



Consigna

En la próxima sesión veremos...

CRIPS-DM:

- Modelo de referencia
- Ejemplo
- Guía de usuario

Preguntas