



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Sintaxis y Semántica de los lenguajes

Trabajo Práctico N°2 Investigación: Python y JavaScript

GRUPO 7

<Zoe Iael Mancini>	<206.464-9>	<zmancini@frba.utn.edu.ar>	<zoemancini>
<Sofia Zalazar>	<204.178-9>	<szalazar@utn.frba.utn>	<SOFIAZALAZAR>
<Román Brezman>	<203.432-3>	<rbrezman@frba.utn.edu.ar>	<rabrezman>
<Mariano Bruno>	<138.794-7>	<marbruno@frba.utn.edu.ar>	<marianonahuelbruno>
<Guido Moreyra >	<209.802-7>	<gmoreyra@frba.utn.edu.ar	<GuidoMoreyra>

Docente: *Roxana Leituz*

Curso: *K2055*

Breve historia de los lenguajes:

Python

El lenguaje **Python** fue desarrollado a fines de la década de **1980** por **Guido van Rossum**, un programador holandés, y su primera versión fue lanzada en 1991 en Centrum Wiskunde & Informatica - CWI (Instituto Nacional de Investigación en Matemáticas y Ciencias de la Computación), en los Países Bajos. El nombre "Python" viene dado por la afición de Van Rossum al grupo Monty Python.

El lenguaje de programación Python fue originalmente desarrollado para el sistema operativo Amoeba. Ha experimentado grandes avances en el desarrollo de código abierto, utilizando las PEPs ("Propuestas de Mejora de Python") como principal herramienta para sugerencias de mejora y discusiones dentro de la comunidad. Estas PEPs se utilizan para describir cambios en el lenguaje o en sus normas, y son evaluadas por el público, aceptadas o rechazadas después de muchas discusiones. Cualquier persona puede escribir y enviar una PEP para su evaluación.

Actualmente es principalmente utilizado en las áreas de:

- *Análisis de datos: debido a sus bibliotecas especializadas como NumPy, Pandas y Matplotlib, que ofrecen herramientas poderosas para el procesamiento, visualización y análisis de datos.*
- *Machine Learning: las bibliotecas como TensorFlow y scikit-learn proporcionan un conjunto completo de herramientas para construir y entrenar modelos de aprendizaje automático.*
- *Desarrollo web: gracias a frameworks populares como Django y Flask. Estos frameworks ofrecen una estructura sólida y eficiente para crear aplicaciones web, facilitando tareas como el enrutamiento, la gestión de bases de datos y la generación de contenido dinámico.*
- *DevOps: se utiliza en la automatización de tareas y la gestión de infraestructuras en el campo de DevOps. Su facilidad de uso, combinada con bibliotecas como Fabric y Ansible, permite a los desarrolladores automatizar procesos y administrar sistemas de manera eficiente.*

JavaScript

JavaScript fue desarrollado originalmente por **Brendan Eich** de Netscape con el nombre de Mocha, el cual fue renombrado posteriormente a LiveScript, para finalmente quedar como JavaScript. El cambio de nombre coincidió con el momento en que Netscape agregó compatibilidad con la tecnología Java en su navegador web Netscape Navigator en la versión 2.0.2 en diciembre de 1995.

Microsoft, al ver el movimiento de uno de sus principales competidores, también decidió incorporar su propia implementación de este lenguaje, llamada JScript, en la versión 3 de su navegador Internet Explorer. Esto contribuyó todavía más al empuje y popularización del lenguaje, pero comenzaron a presentarse pequeños problemas por las diferencias entre implementaciones. Por lo que en 1996 se decidió estandarizar ambas mediante la versión JavaScript 1.1 como propuesta a ECMA, que culminó con el estándar ECMA-262 en junio de 1997. Poco después también como un estándar ISO. Este estándar dicta la base del lenguaje ECMAScript a través de su sintaxis, tipos, sentencias, palabras clave y reservadas, operadores y objetos, y sobre la cual se pueden construir distintas implementaciones. La versión JavaScript 1.3 fue la primera implementación completa del estándar ECMAScript.

JavaScript se introdujo en 1995 como una forma de agregar programas a páginas web en el navegador Netscape Navigator. En la actualidad los navegadores web no son las únicas plataformas en las que se utiliza JavaScript. También es posible ejecutar código JavaScript en un entorno servidor. De hecho las bases de datos, como MongoDB y CouchDB, usan JavaScript como su lenguaje de scripting y consulta.

Comparación de las funciones de ordenamiento

En **Python** el método estándar **Sort()** utiliza el algoritmo de **Timsort**, el cual funciona de la siguiente manera:

División en RUNs:

Primero se divide el arreglo en lo que se denomina **RUNs**, que son particiones del arreglo ya ordenadas. Si no se encuentra ninguna RUN, la función `insertionSort` se encargará de crearlas.

Unión de RUNs:

Luego, todas estas particiones se unirán mediante la función `Merge`.

Detalles del Proceso:

1. División en RUNs:

El algoritmo comienza buscando particiones ordenadas dentro del arreglo principal. Estas particiones son llamadas RUNs.

Una RUN es una secuencia continua de elementos ya ordenados, la cual puede ser creciente o decreciente. Si se encuentra una secuencia decreciente, se invierte para que sea creciente.

Si no se encuentran RUNs suficientemente largas, el algoritmo aplica `InsertionSort` para crear RUNs artificialmente.

2. Inserción de RUNs:

Una vez identificadas o creadas las RUNs, se almacenan en una pila.

Las RUNs se aseguran de tener un tamaño mínimo para optimizar la eficiencia del proceso de mezcla.

3. Mezcla de RUNs:

El algoritmo mezcla las RUNs usando un procedimiento similar al de Merge Sort.

La mezcla de RUNs se hace de una manera que mantiene la estabilidad del ordenamiento, es decir, los elementos iguales mantienen su orden relativo original.

Ventajas de Timsort:

- **Eficiencia en datos parcialmente ordenados:** Timsort es muy eficiente cuando el arreglo ya contiene partes ordenadas
- **Complejidad:** Tiene una complejidad de tiempo de $O(n \log(n))$ en el peor caso, similar a otros algoritmos de ordenamiento eficientes como Mergesort y Quicksort.

En **JavaScript** el método de ordenamiento estándar **sort()** no cuenta con un algoritmo de ordenamiento específico pues este dependerá del motor de **JS** utilizado en el entorno de ejecución.

Funcionamiento del Método sort():

1. Conversión a Cadenas:

- Por defecto, el método **sort()** convierte los elementos del arreglo a cadenas de texto y los ordena según el valor Unicode de dichas cadenas.

2. Función de Comparación:

- Se puede proporcionar una función de comparación para personalizar el ordenamiento. Esta función debe devolver un valor negativo si el primer argumento es menor que el segundo, cero si son iguales y un valor positivo si es mayor.

Motores Comunes y Algoritmos:

- **V8 (Google Chrome y Node.js):** Utiliza una versión optimizada de **Quicksort** para la mayoría de los casos y un algoritmo de **Mergesort** cuando se necesita estabilidad en el ordenamiento.
- **V8 v7.0 / Chrome 70, V8** Utiliza TimSort para mayor estabilidad.
- **SpiderMonkey (Mozilla Firefox):** Utiliza **MergeSort**.
- **JavaScriptCore (Safari):** Utiliza una combinación de algoritmos, incluyendo **Quicksort** y **Merge Sort**.