main.py saved

```
def darPuntaje():
    print('Que puntuación le da a la liga que escogió de 1 a 5')
    puntuacion = input('')
    return puntuacion


print('Hola. Bienvenido a votamatic')
print('1. BundesLiga')
print('2. Liga Santander')
print('3. Premiere Liga')


eleccionUsuario = input('Esc

if eleccionUsuario == '1':
    puntos = darPuntaje()
    eleccion = 'BundesLiga'
dsad
    dsad
if eleccionUsuario == '2':
    puntos = darPuntaje()
    eleccion = 'Liga Santander'
```

PYTHON

python™

¿QUÉ ES UN LENGUAJE DE PROGRAMACIÓN?

LENGUAJES DE PROGRAMACIÓN

- Aplicaciones diseñadas para crear tareas u otras aplicaciones.
- Se basan en un **conjunto de instrucciones.**
- Son **códigos integrados** con un **vocabulario**, una **sintaxis** y una **semántica** especificas para cada lenguaje.

¿QUÉ ES UN PROGRAMA?

PROGRAMA

- **Conjunto de instrucciones ordenadas** que indican al ordenador qué procesos y tareas debe seguir.
- Cada una de las **instrucciones** tiene una **función específica** y está escrita en un lenguaje que el ordenador entienda.

¿QUÉ ES UN TRADUCTOR?

TRADUCTOR

- Herramienta encargada de convertir el código fuente de un determinado lenguaje de programación a código máquina que pueda «entender» el ordenador.

TRADUCTOR - INTERPRETES

- Un intérprete es un traductor que **ejecuta las líneas de código** que conforman un programa **una a una** y directamente.
- Es un programa que va leyendo el código fuente de otro programa y lo va ejecutando según lo lee.
- Ejemplo: **Python**, **JavaScript**, **PHP**

TRADUCTOR - COMPILADORES

- El **programa** fuente **será convertido**, **sentencia a sentencia**, a código máquina, creando un programa objeto o código objeto.
- Para **crear** el programa final, **autoejecutable**, será **necesario** un proceso adicional: el enlazado o montaje (realizado por el programa montador, **enlazador** o linker).
- Ejemplo: **C**, **C++**, **Rust**

TRADUCTOR - INTERMEDIOS

- Otros traductores producen un código intermedio entre el código fuente y el máquina.
- **Java** por ejemplo, genera un recurso bytecode: código precompilado que necesita interpretarse por la **JVM** o **máquina virtual de Java** para ejecutarse.

¿POR QUÉ UTILIZAR PYTHON?

¿POR QUÉ UTILIZAR PYTHON?

- Python es un lenguaje que lleva por debajo C
- Lenguaje de alto nivel
- Lenguaje de propósito general
- Librerías y frameworks
- Compatible con todos los sistemas operativos
- Código abierto
- Baja curva de aprendizaje

MUNDO ACTUAL PYTHON

Noviembre de 2023

MUNDO ACTUAL – APRENDIZAJE AUTOMÁTICO

- TensorFlow es la apuesta clave de Google para construir el ecosistema del futuro del Machine Learning que pueda ser ejecutado en la nube, en aplicaciones o en dispositivos hardware de todo tipo.
- El carácter exploratorio del aprendizaje automático se ajusta a la perfección a Python, así nos podemos encontrar librerías como Keras, PyBrain o scikit-learn para realizar tareas de clasificaciones, regresión, clustering, preprocesamiento o generación de modelos de algoritmos.

MUNDO ACTUAL – DEV OPS

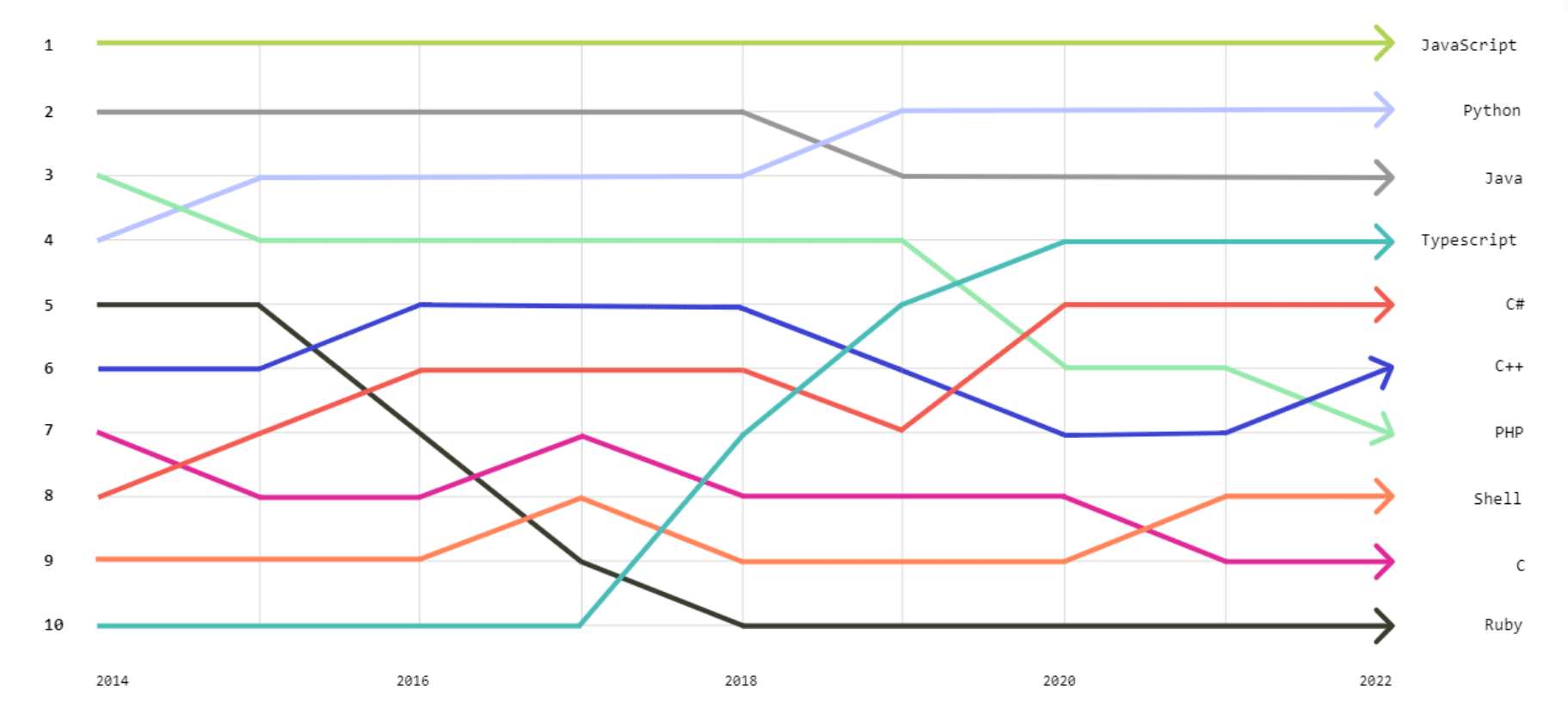
- Se utiliza para realizar scripts y automatizar procesos.
- El hecho de que herramientas como Ansible y SaltStak estén escritas en Python demuestran las capacidades del lenguaje para tareas de automatización.
- Cursos de Data Science o Machine Learning, también podemos destacar cursos para System Admin utilizando Python 3.

MUNDO ACTUAL – SERVICIOS WEB

- Django, el framework para aplicaciones web open source para Python.
- Además, la importancia para crear APIs Restful con librerías como Graphene.

¿CUÁL ES EL NÚMERO REAL DE DESARROLLADORES EN PYTHON?

DESARROLLADORES EN GITHUB



INFORME TIOBE

INFORME TIOBE

Oct 2023	Oct 2022	Change	Programming Language	Ratings	Change
1	1		Python	14.82%	-2.25%
2	2		C	12.08%	-3.13%
3	4	↑	C++	10.67%	+0.74%
4	3	↓	Java	8.92%	-3.92%
5	5		C#	7.71%	+3.29%
6	7	↑	JavaScript	2.91%	+0.17%
7	6	↓	Visual Basic	2.13%	-1.82%
8	9	↑	PHP	1.90%	-0.14%
9	10	↑	SQL	1.78%	+0.00%
10	8	↓	Assembly language	1.64%	-0.75%
11	11		Go	1.37%	+0.10%
12	23	⇈	Scratch	1.37%	+0.69%

INFRAESTRUCTURA COMO CÓDIGO

INFRAESTRUCTURA COMO CÓDIGO

- La infraestructura como código ayuda a la **transición** de la gestión de infraestructura **desde** el **hardware físico** de los centros de datos **a la virtualización**, **los contenedores** y la **computación en la nube**.
- Permite **automatizar** y **estandarizar** el aprovisionamiento y la **gestión de recursos** de infraestructura, como **servidores**, **redes** y **bases de datos**.

MUNDO DE LA SALUD EN PYTHON

MUNDO DE LA SALUD EN PYTHON

- **Análisis de datos médicos**: Analizar grandes conjuntos de datos de pacientes, ensayos clínicos… Bibliotecas: **NumPy** y **Panda**
- **Visualización de datos**: Se utilizan para crear gráficos y visualizaciones de datos médicos. Bibliotecas: **Matplotlib** y **Seaborn**
- **Aprendizaje automático e inteligencia artificial**: Aprendizaje automático en diagnóstico médico, pronóstico de enfermedades y detección de patrones en imágenes médica.

MUNDO DE LA SALUD EN PYTHON

- **Desarrollo de aplicaciones médicas**: Se utiliza para el desarrollo aplicaciones médicas, sistemas de gestión de registros de pacientes, herramientas de telemedicina…
- **Simulación y modelado**: Se utiliza para realizar simulaciones y modelado de sistemas biológicos y médicos.
- **Genómica y bioinformática**: es ampliamente utilizado en genómica para analizar secuencias de ADN, ARN y proteínas, así como en la investigación en biología computacional.