

Características del curso

Información para instructores

Este documento describe las características del curso, destinada sobre todo a los instructores. Para comenzar el estudio de Python se puede ir directamente al capítulo 1, Introducción a Python.

Objetivo

Este curso tiene por objeto capacitar a los estudiantes en la programación en lenguaje Python, optimizando el tiempo de dedicación tanto para instructores como para estudiantes.

La propuesta fue desarrollada intentando contemplar los siguientes requerimientos:

- *Autoaprendizaje.* Permitir e incentivar el autoaprendizaje mediante la provisión de información segmentada en ítems breves, cada uno con ejemplos resueltos y propuestas de tareas como forma de asegurar la actividad, única forma de asegurar un aprendizaje efectivo. Bajo esta consigna, el curso puede seguirse como actividad de auto formación, sin intervención de un instructor.
- *Mínima dedicación docente.* El diseño para autoaprendizaje permite concentrar la dedicación docente en la aclaración de dudas, la discusión de puntos particulares por su dificultad o importancia, la propuesta de actividades complementarias de revisión o de refuerzo.
- *Evaluación automática.* La propuesta de tareas de programación pasibles de ser evaluadas a través de una aplicación permite una evaluación rápida de los estudiantes, donde el tiempo se concentra sobre todo en la preparación de los resultados.

Soporte y formas de trabajo

Para promover el aprendizaje activo es necesario poder experimentar cada aspecto del lenguaje, ya sea un comando, una forma de sintaxis, o una construcción más elaborada como una función o un módulo. Algunos escenarios posibles son:

- *una aplicación* como [Jupyter Notebook](#), última versión [Jupyter Lab](#), donde el texto explicativo puede complementarse con ejemplos interactivos; el estudiante puede también ingresar su propio código en los lugares previstos y ver el resultado de su ejecución. Esta opción tiene un costo de preparación importante, y no permite escribir programas en forma autónoma. Accesible en dos modalidades: 1) instalación de Jupyter Notebook en la máquina del estudiante (es software libre), o 2) el uso de una plataforma como [Google Colab](#), lo cual obliga al estudiante a tener un usuario en Google.
- *texto clásico*, en formato de libro o página web, complementado con un repositorio de ejemplos (archivos *.py). El estudiante debe trabajar en dos ventanas, una para la lectura del texto, la otra para correr los programas provistos como ejemplo. Para experimentar su propio código interactivo, deberá invocar Python directamente en un terminal, o a través de un shell interactivo como [IPython](#). Para escribir sus propios programas, el estudiante deberá contar con un editor de texto llano.
- *texto clásico con ejemplos e IDE.* Los ejemplos son pequeños programas que pueden correrse en una terminal o en un IDE (Integrated Development Environment). Entre las

muchas recomendaciones posibles, Python [IDLE](#) es el IDE oficial de Python; es software libre, puede instalarse como parte de la distribución, es multiplataforma, permite trabajo interactivo y en scripts (módulos .py), y ofrece un buen balance entre complejidad y capacidad. Estas características lo hacen accesible a principiantes, pero permite también desarrollos complejos, como puede esperarse en proyectos fin de carrera.

Esta última modalidad es la adoptada en este curso. Una plataforma como [Jupyter Lab](#) tiene características muy atractivas, pero su propia facilidad tiende a reducir la actividad del estudiante, por lo que requiere un enfoque didáctico adecuado para lograr un aprendizaje efectivo, implementando intervenciones o tareas específicas más demandantes, ya sea dentro o fuera de la plataforma.

Estructura del curso

El curso está estructurado a partir del siguiente conjunto de materiales:

- *Tutoriales por tema*, con la siguiente estructura:
 - Introducción breve a cada unidad.
 - Lecturas recomendadas, en general secciones específicas del [Tutorial de Python](#) en español, en la documentación oficial de Python.
 - Explicación concisa de los distintos temas de la unidad; el formato es un intermedio entre el tutorial convencional y una referencia rápida (Quick Reference).
- *Ejemplos*, pequeños programas que el estudiante puede mirar y correr.
- *Ejercicios*, pequeños programas que el estudiante debe escribir.
- [Tutorial de Python](#), en la [Documentación de Python 3](#).
- [Manual de IDLE](#) en español, en la [Documentación de Python 3](#).

Este material deberá ser complementado con:

- *Ejercicios de evaluación*, pequeños programas que deberá resolver el estudiante y por cuyos resultados será calificado.
- *Corrección automática*. La enseñanza de un lenguaje de programación permite fácilmente la evaluación automática a través de una aplicación. Una propuesta puede verse en el documento [Evaluación automática](#).

Metodología

El curso se puede impartir en forma presencial, pero para muchos auditorios resultará más apto y menos demandante en modalidad semi presencial, dando una breve introducción y luego marcando temas a estudiar para próximos encuentros, reservando las sesiones presenciales para consultas y aclaración de dudas. En modalidad remota, con sesiones de encuentro virtual también para introducción, consultas y aclaración de dudas. Finalmente, el material y armado del curso permite el auto estudio, más difícil que las modalidades anteriores, pero también posible.

Bibliografía

Lecturas recomendadas para aprender el lenguaje.

1. Python. *Tutorial de Python*, en [español](#), en [inglés](#). La versión en español tiene algunos defectos de traducción, pero resulta perfectamente legible.
2. Severance, Charles. [Python para Todos](#). 2020. Introducción general a Python en los primeros 10 capítulos; los 6 capítulos restantes orientados a procesamiento de datos. Cada

capítulo incluye un glosario y ejercicios. En PDF, 262 páginas. Sitio para bajar el libro y otros recursos: [PY4E-ES](#).

Documentación oficial para consulta.

1. *Referencia del lenguaje Python*. Descripción de la sintaxis y los elementos del lenguaje. En inglés, [Language Reference](#); en español, [Referencia del lenguaje](#).
2. *Referencia de la biblioteca*. Describe la biblioteca estándar de Python (constantes, tipos de datos, funciones, excepciones, servicios, módulos), más algunos complementos incluidos en la distribución estándar de Python. En inglés, [The Python Standard Library](#); en español, [Biblioteca estándar de Python](#).

Otros recursos.

1. PEW's Corner. [Python 3 Quick Reference](#). Referencia muy detallada en forma de líneas de código comentadas.
2. Markham, Kevin. [Quick Reference to Python in a single script](#). Una referencia rápida a los comandos de Python en forma de módulo, disponible como módulo .py o como Jupyter notebook.
3. W3 Schools. [Python Tutorial](#). Curso en línea, interactivo, en pasos muy simples. (En inglés).
4. Klein, Bernd. [Python Course](#). Para un estudio exhaustivo del lenguaje Python, experimentando cada particularidad. Extenso, claro, muy completo. (En inglés).