

```
32 self.file = ...
33 self.fingerprints = set()
34 self.logdupes = True
35 self.debug = debug
36 self.logger = logging.getLogger(__name__)
37 if path:
38     self.file = open(os.path.join(path, 'requests.log'),
39                     'a')
40     self.file.seek(0)
41     self.fingerprints.update(self._get_fingerprints())
42
43 @classmethod
44 def from_settings(cls, settings):
45     debug = settings.getbool('DEBUG', False)
46     return cls(job_dir(settings), debug)
47
48 def request_seen(self, request):
49     fp = self.request_fingerprint(request)
50     if fp in self.fingerprints:
51         return True
52     self.fingerprints.add(fp)
```

Programar desde cero con Python – *2da edición 2026*

Equipo docente / **Héctor Chocobar Torrejón**
Unidad académica / **Campus Salto**
Referente académico / **Gimena Abreu**

// Modalidad
A distancia

// Carga horaria total
20 horas (15 sincrónicas +
5 asincrónicas de trabajo
autónomo)

// Días y horarios
Lunes y miércoles, 18:30
a 20:00 h

// Fecha de Inicio
Lunes 13 de julio

// Fecha de Finalización
Miércoles 12 de agosto

Programar desde cero con Python – 2da edición 2026

I Dirigido a

El curso está orientado a personas que posean pocos o nulos conceptos de programación. Se busca introducirlas en los conceptos básicos de los algoritmos y el pensamiento computacional utilizando la sintaxis de Python.

I Objetivo

Brindar a los participantes una introducción sólida a la programación, desarrollando el pensamiento computacional y las competencias necesarias para resolver problemas mediante algoritmos, utilizando Python como lenguaje de aprendizaje inicial.

Al finalizar el curso, las y los estudiantes serán capaces de:

1. Comprender los fundamentos de la programación: variables, operadores, estructuras de control, ciclos y funciones.
2. Aplicar buenas prácticas de codificación siguiendo las convenciones de estilo de Python (PEP8).
3. Manipular estructuras de datos básicas (listas, tuplas, conjuntos, diccionarios) y comprender su utilidad en distintos contextos.
4. Diseñar y ejecutar programas simples en Python, aplicando lógica condicional, iteración y modularización del código.
5. Utilizar cadenas y operaciones sobre texto para procesar información de manera eficiente.
6. Integrar librerías estándar y externas para ampliar las funcionalidades básicas de Python.
7. Desarrollar autonomía en el aprendizaje de la programación, explorando recursos,

librerías y documentación disponibles en la comunidad de Python.

I Metodología

Clases teórico-prácticas sincrónicas, complementadas con trabajo autónomo asincrónico.

I Contenido

Se abordarán las bases fundamentales sobre programación, las cuales son válidas y aplicables para cualquier otro lenguaje en el que se proponga desarrollar en el futuro.

Se estudiarán los mecanismos de creación de variables, operaciones lógicas y aritméticas, dirección del flujo de un programa, estructuras de datos y funciones.

Entorno de trabajo y Conceptos Básicos

- Descargar e instalar Python y PyCharm
- PEP8: La guía de estilo de código de Python
- Mi primer script: Hola Mundo
- Salida y entrada de datos. Tipos de datos
- Asignación de valores, operador de asignación
- Operadores aritméticos, relacionales, lógicos
- Comentarios
- Palabras claves o reservadas de Python
- Funciones integradas de Python

Estructuras condicionales

- Estructura if-elif-else
- Condicionales anidados, one-liners: if else

Colecciones

- Listas, Tuplas, Conjuntos, Diccionarios
- Lista de Diccionarios. El formato de datos: JSON

Programar desde cero con Python – 2da edición 2026

Estructuras de ciclos

- Ciclo while, ciclo for
- Instrucciones continue y break

Cadenas

- Literales de cadena formateados: f-strings
- Cadena de caracteres. Índices y slicing

Funciones

- Declaración e invocación de funciones
- Argumentos y parámetros

Librerías de Python

- Librerías standard de Python
- PyPI: Repositorio de software para el lenguaje de programación Python

| Equipo docente

Héctor Chocobar Torrejón

Ingeniero en Sistemas y Profesor universitario con más de 30 años de experiencia en desarrollo de software e infraestructura de redes informáticas. Además de su labor en la UCU, Héctor es Senior Mentor en los bootcamps de 4Geeks Academy, donde comparte su experiencia y conocimientos en las áreas de Full-stack Development, Data Science y Machine Learning. Complementando su sólida trayectoria, Héctor ha obtenido certificaciones de la University of Michigan en áreas como Data Science, Programación con Python y Web Development & Coding.

| Costo total

\$8.000 pesos uruguayos

Financiación hasta en 6 cuotas con tarjetas de crédito. Pago en efectivo a través de redes de cobranza o transferencia bancaria.

Beneficios: Estudiantes UCU 25% - Alumni UCU 25% - Personal docente y administrativo UCU consultar. Consultar por convenios con otras instituciones.