



William Alan M.

Board Member at OpenEDG



Raquel Martinez

Technical Manager EMEAR
Cisco



Jordi Ariño

Python Institute Tech Lead
en PUE Academy

Python Institute®

**Formación y
certificación oficial en
Python para alcanzar
el éxito**



**Facilitamos la formación
y certificación oficial en tecnología
al mundo educativo**

Python Institute

- ✓ **Python Institute** de OpenEDG ofrece un **programa** global **de formación y certificación** para el lenguaje de programación **Python**.
- ✓ Este programa consiste en una **ruta de varios tracks** de para diferentes áreas de especialización.
- ✓ Dentro de cada ruta o track existen **3 niveles** de especialización: **Entry, Associate, and Professional**.
- ✓ El curso core Python Essentials está integrado dentro de la plataforma Cisco Networking Academy.





	Entry	Associate	Professional	
	 Networking CISCO Academy			
General-Purpose Programming	 PCEP-30-xx Certified Entry-Level Python Programmer	 PCAP-31-xx Certified Associate Python Programmer	 PCPP-32-1xx Certified Professional Python Programmer Level 1	 PCPP-32-2xx Certified Professional Python Programmer Level 2 coming Q1/Q2 2024
Data Science	 PCED-30-xx Certified Entry-Level Data Analyst with Python coming Q4 2023	 PCAD-31-xx Certified Associate Data Analyst with Python coming Q4 2023/Q1 2024		
Testing	 PCET-30-xx Certified Entry-Level Tester with Python coming Q4 2023/Q1 2024	 PCAT-31-xx Certified Associate Tester with Python coming Q4 2023	 PCPT-32-xx Certified Professional Tester with Python coming Q3/Q4 2024	
Security	 PCES-30-xx Certified Entry-Level Security Specialist with Python coming Q1 2024	 PCAS-31-xx Certified Associate Security Specialist with Python coming Q3 2024		
Network Programming	 PCEN-30-xx Certified Entry-Level Network Programmer with Python coming Q2 2024	 PCAN-31-xx Certified Associate Network Programmer with Python coming Q2/Q3 2024		
Web Development	 PCEW-30-xx Certified Entry-Level Web Developer with Python coming Q1/Q2 2024	 PCAW-31-xx Certified Associate Web Developer with Python coming Q3/Q4 2024		



Python
<https://pythoninstitute.org>





Raquel Martinez

Technical Manager
EMEAR Cisco



Cisco Networking Academy

Cisco Networking Academy is the **world's largest and longest running** corporate social responsibility education program, powering an inclusive future for students and instructors around the world.

Cisco partners with learning institutions worldwide to **deliver technical training and problem-solving experiences** to individuals studying networking, security, and IoT technologies.

OpenEDG Python Institute is proud to partner with Cisco with a vision and desire to **create career opportunities, transform the workforce, and improve the lives** of millions of people all across the globe.



→ Required prior knowledge

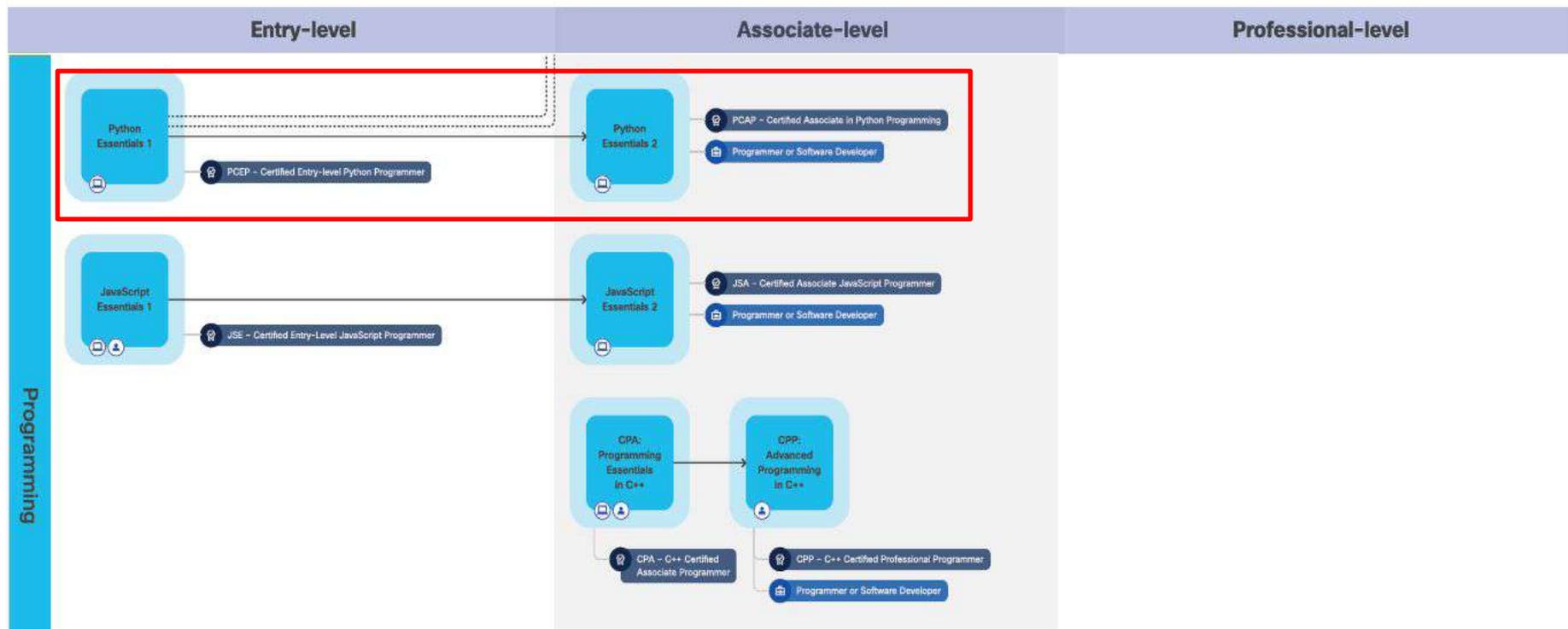
.....→ Recommended prior knowledge

Aligns to Certification

Aligns to Job Role

Self-paced course

Instructor-led course



Catalog > Fundamentos de Python 1

Fundamentos de Python 1

This course is part of the [Learning Collections - Python](#)

Aprenda los conceptos fundamentales de la programación informática y comience a desarrollar habilidades de codificación con el lenguaje de programación Python.

LANGUAGES AVAILABLE

English, Español, Português, Український.

Español (Spanish) ▾

Get Started

Overview | Curriculum

Utilizado por nuevas empresas y gigantes tecnológicos como Google, Facebook, Netflix y más, Python le ofrece infinitas posibilidades para crear proyectos de software a pequeña y gran escala. Fácil de usar con un código fácil de leer, Python es un excelente primer lenguaje de programación para aprender y no requiere conocimientos previos de programación. Las habilidades de Python lo abren a carreras en casi cualquier industria y son necesarias si desea continuar con roles de ingeniería y desarrollo de software más avanzados y mejor pagados, como ingeniero de software, administrador de sistemas e ingeniero de seguridad.

En este curso, aprenderá habilidades en demanda, como diseñar, desarrollar y mejorar programas de computadora, métodos para analizar problemas usando programación, mejores prácticas de programación y más. El curso también lo prepara para la certificación PCEP: programador de Python de nivel de entrada certificado (examen PCEP-30-0X).

Python
INSTITUTE PI



FREE



30 HOURS



BEGINNER



30 LABS



SELF-PACED

Achievements

Badges you can earn in this course.



Catalog > Python > Fundamentos de Python 2

Fundamentos de Python 2

This course is part of the [Learning Collections - Python](#)

Amplíe sus habilidades de programación de computadoras utilizando Python y prepárese para obtener una certificación.

LANGUAGES AVAILABLE

English, Español.

Español (Spanish)

Get Started



[Overview](#) | [Curriculum](#)

Un sinfín de posibilidades te esperan cuando adquieres habilidades de programación en demanda. Python es un lenguaje de programación multiparadigma utilizado por nuevas empresas y gigantes tecnológicos como Google, Facebook, Netflix y más. Con una sintaxis intuitiva y legible, Python es un excelente primer lenguaje de programación que puedes aprender. Tener habilidades de Python lo califica para carreras en casi cualquier industria y es un requisito para roles de ingeniería y desarrollo de software avanzados y mejor pagados.

Este curso es el segundo de la serie de dos cursos de Fundamentos de Python y está diseñado para estudiantes que ya están familiarizados con los conceptos básicos de Python. Sumérjase en conceptos avanzados de Python, aprenda programación orientada a objetos (POO) y desarrolle su propia cartera de programación. Este curso lo prepara para la certificación [PCAP - Certified Associate in Python Programming certification](#) (Exam PCAP-31-0x).



FREE



40 HOURS



INTERMEDIATE



21 LABS



SELF-PACED

Achievements

Badges you can earn in this course.





William Alan M.

Board Member
at OpenEDG



Python Institute

Demanda laboral



Top 10 lenguajes de programación más demandados en 2023

Al observar el mayor número de ofertas de empleo en LinkedIn en Latinoamérica, Estados Unidos y Europa, **Python** continúa en el **primer puesto**.

En **segundo lugar**, está **JavaScript**, mientras que **C++**, **C#**, TypeScript, PHP, Go, Swift, Java, SQL completan el **top 10**

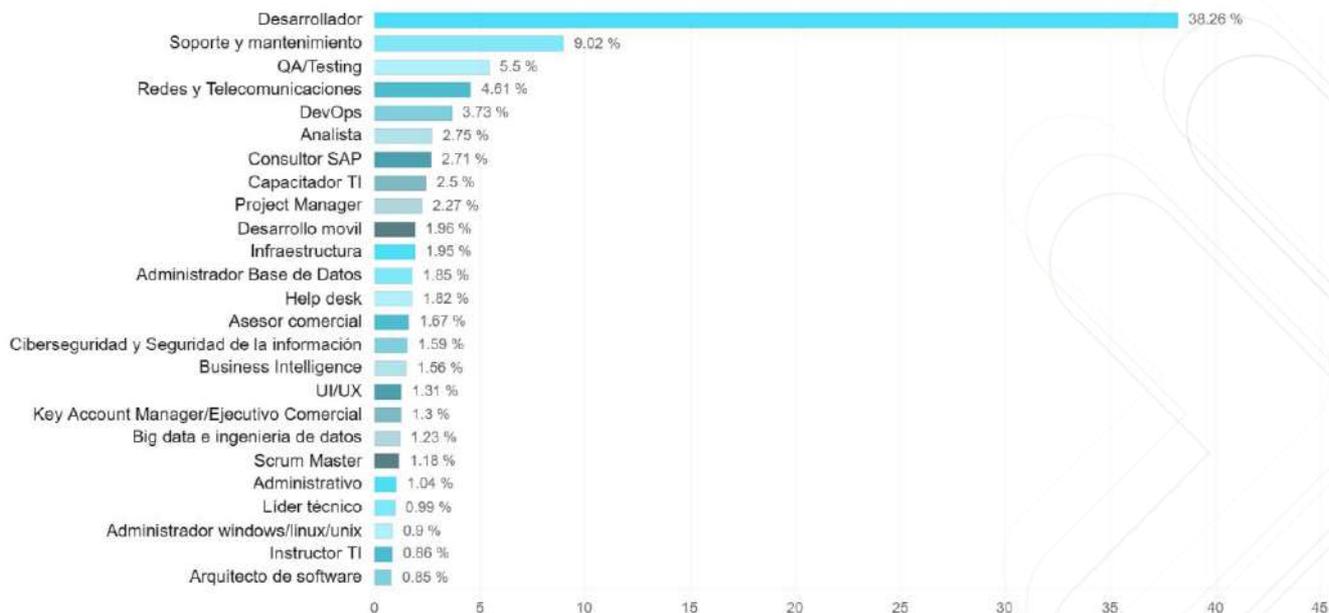




Tecnologías más demandadas

El perfil de tecnología más demandado en **TI** es el de **desarrollador o programador de software**. Específicamente, aquellos que tienen conocimientos en **Java, .NET y Python**.

Este perfil representó un **38.36%** del total de las vacantes de **TI** disponibles analizadas para el Reporte.





Oportunidades laborales



Nuevos trabajos de Python developer -
22 de agosto, 2023 | Indeed.com



Python - Trabajo Freelance - Workana



Empleos de Desarrollador C++ (hireline.io)



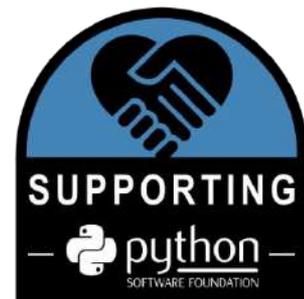
OpenEDG | Python Institute

Promovemos lenguajes de programación capacitando a una nueva generación de programadores.

Trabajamos en estrecha colaboración con escuelas, colegios, universidades, empresas, organizaciones e instituciones para ayudarlos a cumplir con sus objetivos con respecto lenguajes como **C, C++, Python y Java Script.**

Estas son algunas de las formas en que lo hacemos:

- Creamos contenido de cursos para mejorar la calidad de la educación de.
- Desarrollamos evaluaciones de habilidades, certificaciones.
- Analizamos la demanda de los lenguajes para cerrar la brecha entre la educación y la industria.
- Proporcionamos herramientas y recursos de enseñanza y aprendizaje para Instituciones educativas y estudiantes individuales.
- Apoyamos el trabajo y la misión de **Python Software Foundation.**



Python Institute | ¿Qué es Python?

Python es una ... víbora!



Un lenguaje de programación interpretado, orientado a objetos de alto nivel con semántica dinámica, utilizado para la programación de propósito general.

Lanzado por primera vez en 1991, creado originalmente por **Guido van Rossum**



¡El lenguaje de programación más popular del mundo!

Worldwide, Mar 2024

Rank	Change	Language	Share	1-year trend
1		Python	28.59 %	+1.0 %
2		Java	15.79 %	-0.5 %
3		JavaScript	8.7 %	-0.8 %
4		C#	6.77 %	-0.0 %
5		C/C++	6.76 %	-0.0 %
6	↑	R	4.71 %	+0.5 %
7	↓	PHP	4.5 %	-0.7 %
8		TypeScript	2.86 %	+0.1 %
9		Swift	2.74 %	+0.5 %
10		Objective-C	2.4 %	+0.1 %



PYPL Popularity of Programming Language index

Python | Certificaciones





EDUBE

Entorno de Aprendizaje

Nuestro entorno de aprendizaje www.edube.org proporciona acceso gratuito a todos nuestros cursos de preparación para las actuales y futuras certificaciones ofertadas por **Python Institute**, **C++ Institute** y **JS Institute**

Objetivos

- Familiarizarse con el concepto de almacenar y trabajar con diferentes tipos de datos en Python.
- Experimentar con el código en Python.

Escenario

A continuación una historia:

Érase una vez en la Tierra de las Manzanas, Juan tenía tres manzanas, María tenía cinco manzanas, y Aidán tenía seis manzanas. Todos eran muy felices y vivieron por muchísimo tiempo. Fin de la Historia.

Tu tarea es:

- Crear las variables: `juan`, `maria`, y `aidan`.
- Asignar valores a las variables. El valor debe de ser igual al número de manzanas que cada quien tenía.
- Una vez almacenados los números en las variables, imprimir las variables en una línea, y separar cada una de ellas con una coma.
- Después se debe crear una nueva variable llamada `total_manzanas` y se debe igualar a la suma de las tres variables anteriores.
- Imprime el valor almacenado en `total_manzanas` en la consola.
- Experimenta con tu código: crea nuevas variables, asigna diferentes valores a ellas, y realiza varias operaciones aritméticas con ellas (por ejemplo, `+`, `*`, `/`, etc.). Intenta poner una cadena con un entero juntos en la misma línea, por ejemplo, "Número Total de Manzanas:" y `total_manzanas`.

```

1 juan = 3
2 maria = 5
3 aidan = 6
4
5 print(juan, maria, aidan, sep=',')
6
7 total_manzanas = juan + maria + aidan
8 print(total_manzanas)
9
10 piedad = 10.5
11 cubada = 2
12 print(cubada / piedad)
13 print("Número Total de Manzanas:", total_manzanas)
14

```

Output:

```

3, 5, 6
14
0.25
Número Total de Manzanas: 14

```

Python Institute

Move your career in the right direction.

Python Essentials

Learn. Certify. Excel.

Python Essentials 1 (Python 3.10.0) | Python Essentials 2 (Python 3.10.0) | Python Essentials 3 (Python 3.10.0) | Python Essentials 4 (Python 3.10.0)

Python Essentials 5 (Python 3.10.0) | Python Essentials 6 (Python 3.10.0) | Python Essentials 7 (Python 3.10.0) | Python Essentials 8 (Python 3.10.0)

Python Essentials 9 (Python 3.10.0) | Python Essentials 10 (Python 3.10.0) | Python Essentials 11 (Python 3.10.0) | Python Essentials 12 (Python 3.10.0)

<http://www.edube.org/>

PE2 -- Module 4 Test

Progress (6%) | Time Left: 29:07

Question 1/16

What is the meaning of the value represented by `errno.EXIST` ?

- Bad file number
- File exists
- File doesn't exist
- Permission denied



EDUBE | Laboratorios de Práctica

Todos nuestros cursos incluyen laboratorios (+300) que ayudan al estudiante a practicar las habilidades cubiertas en clase o prepararse para la próxima clase. Para realizar las prácticas, puede utilizar un entorno de programación en línea dedicado e interactivo: **Edube**, una herramienta integrada con el curso como intérprete que le permite iniciar, realizar y probar ejercicios de laboratorio diseñados específicamente para este curso.

```
1. Python program computes the number of seconds in a given number of hours
2. # this program has been written 100 days ago
3.
4. n = 2 # number of hours
5. seconds = 3600 # number of seconds in 1 hour
6.
7. print("Hours: ", n) #printing the number of hours
8. # print("seconds in hours: ", n * seconds) # printing the number of seconds in a giv
9.
10. #here we should also print "Goodbye", but a programmer didn't have time to write any
11. #this is the end of the program that computes the number of seconds in 3 hours
```

Chapter 1 Assessment (CLA)
Time left: 28:36



Select the correct answer (single choice)

a natural language

a machine language

a programming language

Submit Test



Certificaciones | Encuesta sobre el Valor

En este año 2023, realizamos una encuesta a **12 mil estudiantes** certificados entre el 1 de enero y el 31 de diciembre del 2022 en [PCEP Certification \(Entry-Level\) | Python Institute](#) y [PCAP Certification \(Associate\) | Python Institute](#)

Respondieron **1,100** personas certificadas en una certificación de **Python (PCEP/PCAP)**:

- ❑ **82% / 85%:** Obtener la certificación me ha hecho más valioso en el mercado laboral.
- ❑ **73% / 77%:** Obtener la certificación ha aumentado mi satisfacción laboral.
- ❑ **90% / 90%:** Obtener la certificación ha aumentado mi confianza en mis habilidades y/o competencia profesional.
- ❑ **64% / 66%:** Obtener la certificación PCEP/PCAP me ha ayudado a hacer crecer mi red profesional.





Certificaciones | Encuesta sobre el Valor

Al obtener la certificación PCEP/PCAP:

- 84% / 85%:** Me ha dado mayor determinación para triunfar profesionalmente.
- 77% / 82%:** Ha aumentado la calidad y el valor de mis contribuciones laborales.
- 75% / 80%:** Ha aumentado la eficacia y eficiencia de mi trabajo.
- 90% / 90%:** Me ha dado una mejor idea sobre lo que todavía necesito aprender (dónde están mis carencias de habilidades).
- 83% / 82%:** Me ha dado una mejor comprensión del alcance del conocimiento que se necesita para trabajos relacionados.
- 64% / 66%:** Mejoré mi currículum
- 32% / 31%:** Consiguieron trabajo
- 1 de cada 3 consiguieron un empleo gracias a la certificación**



OpenEDG | Referencias



Página principal:

<https://openedg.org>



C++

<https://cppinstitute.org>



JavaScript

<https://js.institute/>



Python

<https://pythoninstitute.org>



EnglishForIT

<https://englishforit.org/>



Plataforma de aprendizaje e información sobre las certificaciones:

<https://edube.org/>



Tienda para vouchers

<https://ums.edube.org/store>





William Allan

Board Member at OpenEDG





Jordi Ariño

Python Institute Tech Lead
en PUE Academy

Python Institute: Curso Python Essentials





	Entry	Associate	Professional
General-Purpose Programming	PCEP-30-xx Certified Entry-Level Python Programmer	PCAP-31-xx Certified Associate Python Programmer	PCPP-32-1xx Certified Professional Python Programmer Level 1 PCPP-32-2xx Certified Professional Python Programmer Level 2 coming Q1/Q2 2024
Data Science	PCED-30-xx Certified Entry-Level Data Analyst with Python coming Q4 2023	PCAD-31-xx Certified Associate Data Analyst with Python coming Q4 2023/Q1 2024	
Testing	PCET-30-xx Certified Entry-Level Tester with Python coming Q4 2023/Q1 2024	PCAT-31-xx Certified Associate Tester with Python coming Q4 2023	PCPT-32-xx Certified Professional Tester with Python coming Q3/Q4 2024
Security	PCES-30-xx Certified Entry-Level Security Specialist with Python coming Q1 2024	PCAS-31-xx Certified Associate Security Specialist with Python coming Q3 2024	
Network Programming	PCEN-30-xx Certified Entry-Level Network Programmer with Python coming Q2 2024	PCAN-31-xx Certified Associate Network Programmer with Python coming Q2/Q3 2024	
Web Development	PCEW-30-xx Certified Entry-Level Web Developer with Python coming Q1/Q2 2024	PCAW-31-xx Certified Associate Web Developer with Python coming Q3/Q4 2024	



Python
<https://pythoninstitute.org>



Courses

Certifications

Level ▾

Category ▾

Technology ▾

Clear

Course

Python
Essentials 1



PYTHON ESSENTIALS 1
Python 101 (PE1)

BEGINNER

Aligned with PCEP-30-01/PCEP-30-02

Course

Python
Essentials 2



PYTHON ESSENTIALS 2
Python 102 (PE2)

INTERMEDIATE

Aligned with PCAP-31-02/PCAP-31-03

Course Series

Advanced
OOP



PYTHON ADVANCED 1
Python 201 (Advanced OOP)

ADVANCED

Aligned with PCPP-32-101

Course Series

Best Practices
and
Standardization



PYTHON ADVANCED 2
Python 202 (PEPs)

ADVANCED

Aligned with PCPP-32-101

Course Series

GUI
Programming



PYTHON ADVANCED 3
Python 203 (GUI Programming)

ADVANCED

Aligned with PCPP-32-101

Course Series

Working with
RESTful APIs



PYTHON ADVANCED 4
Python 204 (RESTful APIs)

ADVANCED

Aligned with PCPP-32-101

Course Series

File
Processing



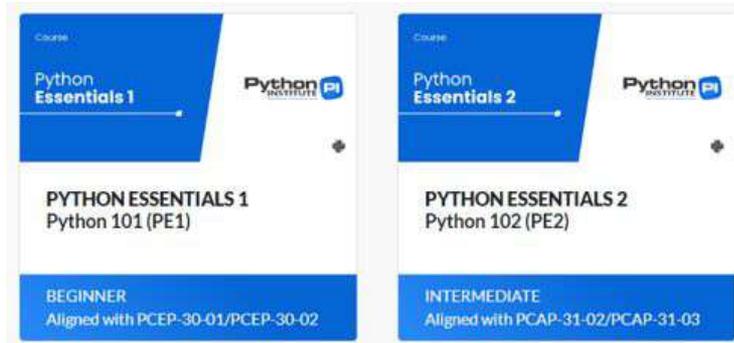
PYTHON ADVANCED 5
Python 205 (File Processing)

ADVANCED

Aligned with PCPP-32-101

Python Essentials

- ✓ El curso de Python Essentials cubre todos los conceptos básicos de la programación en Python, así como conceptos y técnicas generales de programación informática y el enfoque orientado a objetos.
- ✓ El curso está dividido en dos partes:
Python Essentials 1 (PE1) -> 4 módulos.
Python Essentials 2 (PE2) -> 4 módulos.



Python Essentials

- ✓ Aprender los conceptos universales de la programación.
- ✓ Practicas habilidades para resolver desafíos típicos de implementación.
- ✓ Instalar un entorno de ejecución de Python.
- ✓ Escribir tus propios scripts Python.
- ✓ Aprender la sintaxis y la semántica del lenguaje Python.
- ✓ Aprender técnicas generales de codificación y mejores prácticas.
- ✓ Aprender a utilizar la programación orientada a objetos en Python.
- ✓ Aprender a importar y utilizar módulos de Python.
- ✓ Aprender a manejar excepciones.
- ✓ Aprender a procesar archivos.
- ✓ Utilice los componentes más importantes de la biblioteca estándar de Python (PSTL).



Python Essentials Part 1

Demo



Python 101 (PE1): Python Essentials - Part 1



Python **Essentials**
Part 1

Offered by

 **PYTHON INSTITUTE**
Open Education & Development Group

 python™





Python **Essentials 1**


Entry-Level

Python Beginner

PYTHON ESSENTIALS 1
PYTHON 101 (PE1)

Python 101 (PE1): Python Essentials - Part 1



PE1: Módulo 1. Introducción a Python y a la Programación Informática



PE1: Módulo 2. Tipos de datos, Variables, Operaciones Básicas de Entrada y Salida, Operadores Básicos



PE1: Módulo 3. Valores Booleanos, Ejecución Condicional, Bucles, Listas y su procesamiento, Operaciones Lógicas y de Bit a Bit



PE1: Módulo 4. Funciones, Tuplas, Diccionarios, Excepciones y Procesamiento de Datos



Fundamentos de Python 1 (PE1) Examen Final del Curso



Python
Essentials 1



Entry-Level

Python

Beginner

PYTHON ESSENTIALS 1

PYTHON 101 (PE1)

Catalog > Fundamentos de Python 1

Fundamentos de Python 1

This course is part of the **Learning Collections - Python**

Aprenda los conceptos fundamentales de la programación informática y comience a desarrollar habilidades de codificación con el lenguaje de programación Python.

LANGUAGES AVAILABLE

English, Español, Português, Український.

Español (Spanish) ▾

[Get Started](#)

Overview | Curriculum

Utilizado por nuevas empresas y gigantes tecnológicos como Google, Facebook, Netflix y más, Python le ofrece infinitas posibilidades para crear proyectos de software a pequeña y gran escala. Fácil de usar con un código fácil de leer, Python es un excelente primer lenguaje de programación para aprender y no requiere conocimientos previos de programación. Las habilidades de Python lo abren a carreras en casi cualquier industria y son necesarias si desea continuar con roles de ingeniería y desarrollo de software más avanzados y mejor pagados, como ingeniero de software, administrador de sistemas e ingeniero de seguridad.

En este curso, aprenderá habilidades en demanda, como diseñar, desarrollar y mejorar programas de computadora, métodos para analizar problemas usando programación, mejores prácticas de programación y más. El curso también lo prepara para la certificación PCEP: programador de Python de nivel de entrada certificado (examen PCEP-30-0X).



FREE



30 HOURS



BEGINNER



30 LABS



SELF-PACED

Achievements

Badges you can earn in this course.



Descripción general del curso

Este curso enseña habilidades bajo demanda , incluido cómo diseñar, desarrollar y mejorar programas informáticos utilizando Python.

Beneficios

Utilizado por nuevas empresas y gigantes tecnológicos como Google, Facebook, Netflix y más, Python con un código fácil de usar y fácil de leer ofrece infinitas posibilidades para crear proyectos de software a pequeña y gran escala.

Explore oportunidades en tecnología

- ✓ Explora el mundo de la programación informática y las carreras que ofrece.
- ✓ Desarrolla habilidades de codificación usando Python.
- ✓ Enseña sobre tipos de datos, variables, operaciones de E/S, flujo de control y funciones.

Detalles del curso

Público objetivo: Estudiantes de Secundaria y Universitarios de 2 años, público general

Tiempo estimado para completar: 30 horas

Prerequisitos: Ninguno

Entrega del curso: Impartida por instructor o Propio Ritmo

Aspectos destacados del componente de aprendizaje:

- ✓ 4 módulos y 30 laboratorios de práctica.
- ✓ Actividades interactivas y cuestionarios
- ✓ 1 proyecto final
- ✓ 1 prueba final

Reconocimientos del Curso: Insignia Digital

Certificación alineada : PCEP:
certificación de programador Python certificado de nivel inicial

Próximo curso recomendado:
Python Essentials 2

skillforall.com



Requisitos

- Alineación ASC: Recomendada
- Formación de instructores: Opcional
- Equipo básico: Computadora/ dispositivo móvil e Internet
- Equipo adicional requerido: No

Desarrollado en colaboración con



Networking
CISCO Academy

Fundamentos de Python 1

Esquema de Curso Recursos

Buscar esquema del curso

PE1: Módulo 1. Introducción a Python y a la Programación Informática 0%

1.0. Bienvenido a Fundamentos de Python 1 0/4

- 1.0.1 Aprende Python - el lenguaje de hoy y mañana
- 1.0.2 Acerca del curso
- 1.0.3 Plan de Estudios
- 1.0.4 Prepárate para el examen PCEP-30-0x

1.1. Sección 1 - Introducción a la programación 0/7

1.2. Sección 2 - Introducción a Python 0/9

1.3. Sección 3 - Descarga e instalación de Python 0/15

1.4. Finalización del Módulo 1 - Prueba del Módulo

PE1: Módulo 2. Tipos de datos, Variables, Operaciones Básicas de Entrada y Salida, Operadores Básicos

PE1: Módulo 3. Valores Booleanos, Ejecución Condicional, Bucles, Listas y su procesamiento, Operaciones Lógicas y de Bit a Bit

1.0 Bienvenido a Fundamentos de Python 1

Deslizar para comenzar



Buscar esquema del curso

El módulo 1: Introducción a Python y a la programación

Informática

27%

1.0. Bienvenido a Fundamentos de Python 1

4 / 4

1.1. Sección 1 - Introducción a la programación

0 / 7

1.2. Sección 2 - Introducción a Python

3 / 9

1.2.1 Python - una herramienta, no un reptil

1.2.2 ¿Quién creó Python?

1.2.3 Un proyecto de programación por pasatiempo

1.2.4 ¿Qué hace que Python sea tan especial?

1.2.5 ¿Ritales de Python?

1.2.6 ¿Dónde podemos ver a Python en acción?

1.2.7 ¿Por qué no Python?

1.2.8 Existe más de un Python

1.2.9 Implementaciones de Python

1.3. Sección 3 - Descarga e instalación de Python

0 / 5

1.4. Finalización del Módulo 1 - Prueba del Módulo

PE1: Módulo 2. Tipos de datos, Variables, Operaciones Básicas de

1.2.2 ¿Quién creó Python?

1.2.2 ¿Quién creó Python?

Una de las características sorprendentes de Python es el hecho de que en realidad es el trabajo de una persona. Por lo general, los grandes lenguajes de programación son desarrollados y publicados por grandes compañías que emplean a muchos profesionales, y debido a las normas de derechos de autor, es muy difícil nombrar a cualquiera de las personas involucradas en el proyecto. Python es una excepción.

No existen muchos lenguajes de programación cuyos autores sean conocidos por su nombre. Python fue creado por [Guido van Rossum](#), nacido en 1956 en Haarlem, Países Bajos. Por supuesto, Guido van Rossum no desarrolló y evolucionó todos los componentes de Python.

La velocidad con la que Python se ha extendido por todo el mundo es el resultado del trabajo continuo de miles de (muy a menudo anónimos) programadores, testers, usuarios (muchos de ellos no son especialistas en TI) y entusiastas, pero hay que decir que la primera idea (la semilla de la que brotó Python) llegó a una cabeza: la de Guido.



Guido van Rossum



CISCO Networking Academy | Fundamentos de Python 1

Esquema de Curso | Recursos

Buscar esquema del curso

PE1: Módulo 1. Introducción a Python y a la Programación Informática 40%

- 1.0. Bienvenido a Fundamentos de Python 1 4/4
- 1.1. Sección 1 - Introducción a la programación 0/7
- 1.2. Sección 2 - Introducción a Python 0/9
- 1.3. Sección 3 - Descarga e instalación de Python** 0/5
 - 1.3.1 Comienza tu viaje con Python
 - 1.3.2 Como descargar, instalar y configurar Python
 - 1.3.3 Comenzando tu trabajo con Python
 - 1.3.4 Tu primer programa antes de tu primer programa...
 - 1.3.5 Como estropear y arreglar tu código
- 1.4. Finalización del Módulo 1 - Prueba del Módulo

PE1: Módulo 2. Tipos de datos, Variables, Operaciones Básicas de Entrada y Salida, Operadores Básicos

PE1: Módulo 3. Valores Booleanos, Ejecución Condicional, Bucles, Matrices, Tuplas, Diccionarios, Listas, etc.

1.3. Sección 3 - Descarga e instalación de Python

1.3 Sección 3 - Descarga e instalación de Python

Deslizar para comenzar



Esquema de Curso

Recursos

Buscar esquema del curso

PE1: Módulo 1. Introducción a Python y a la Programación Informática

56%

- 1.0 Bienvenido a Fundamentos de Python 1 4 / 4
- 1.1 Sección 1 - Introducción a la programación 0 / 7
- 1.2 Sección 2 - Introducción a Python 0 / 2
- 1.3 Sección 3 - Descarga e instalación de Python 4 / 5
- 1.4. Finalización del Módulo 1 - Prueba del Módulo

PE1: Módulo 2. Tipos de datos, Variables, Operaciones Básicas de Entrada y Salida, Operadores Básicos

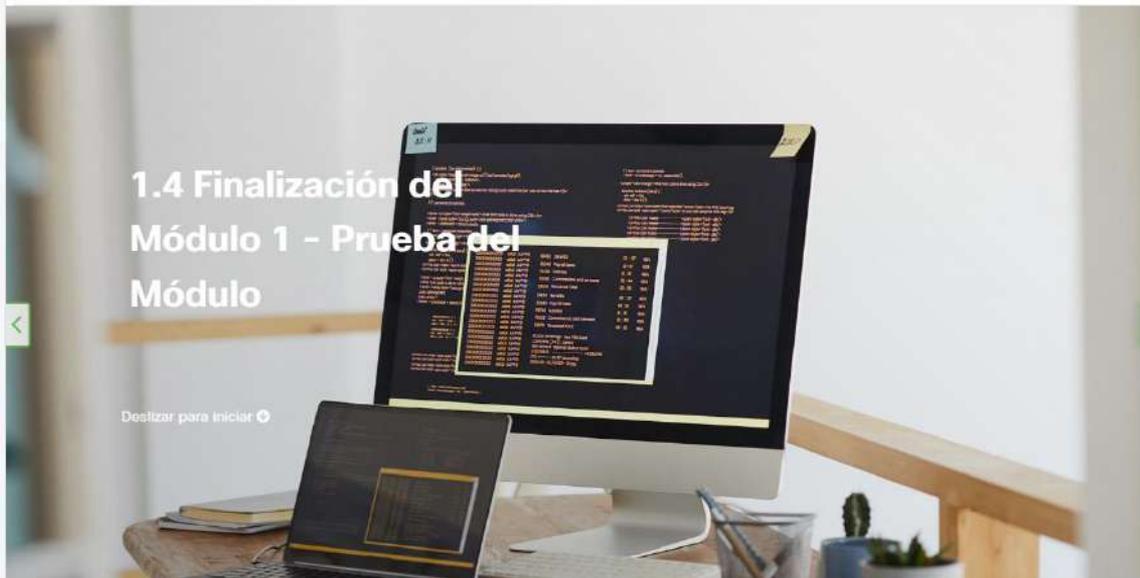
PE1: Módulo 3. Valores Booleanos, Ejecución Condicional, Bucles, Listas y su procesamiento, Operaciones Lógicas y de Bit a Bit

PE1: Módulo 4. Funciones, Tuplas, Diccionarios, Excepciones y Procesamiento de Datos

Fundamentos de Python 1 (PE1) Examen Final del Curso

1.4. Finalización del Módulo 1 - Prueba del Módulo

🔍 🌐 ES ✎



1.4 Finalización del Módulo 1 - Prueba del Módulo

Deslizar para iniciar



Busca esquemas del curso

PE1: Módulo 1. Introducción a Python y a la Programación
Informática

56%

- 1.0. Bienvenido a Fundamentos de Python 1 4 / 4
- 1.1. Sección 1 - Introducción a la programación 0 / 7
- 1.2. Sección 2 - Introducción a Python 9 / 9
- 1.3. Sección 3 - Descarga e instalación de Python 4 / 5
- 1.4. Finalización del Módulo 1 - Prueba del Módulo

PE1: Módulo 2. Tipos de datos, Variables, Operaciones Básicas de
Entrada y Salida, Operadores Básicos

PE1: Módulo 3. Valores Booleanos, Ejecución Condicional, Bucles,
Listas y su procesamiento, Operaciones Lógicas y de Bit a Bit

PE1: Módulo 4. Funciones, Tuplas, Diccionarios, Excepciones y
Procesamiento de Datos

Fundamentos de Python 1 (PE1) Examen Final del Curso

Pregunta 9

¿Cuál es el comportamiento esperado del siguiente programa?

```
print("¡Hola!")
```

- El programa mostrará "¡Hola!" en la pantalla
- El programa mostrará "¡Hola" en la pantalla
- El programa mostrará ¡Hola! en la pantalla
- El programa generará un mensaje de error en la pantalla

Esquema de Curso

Recursos

Buscar esquema del curso

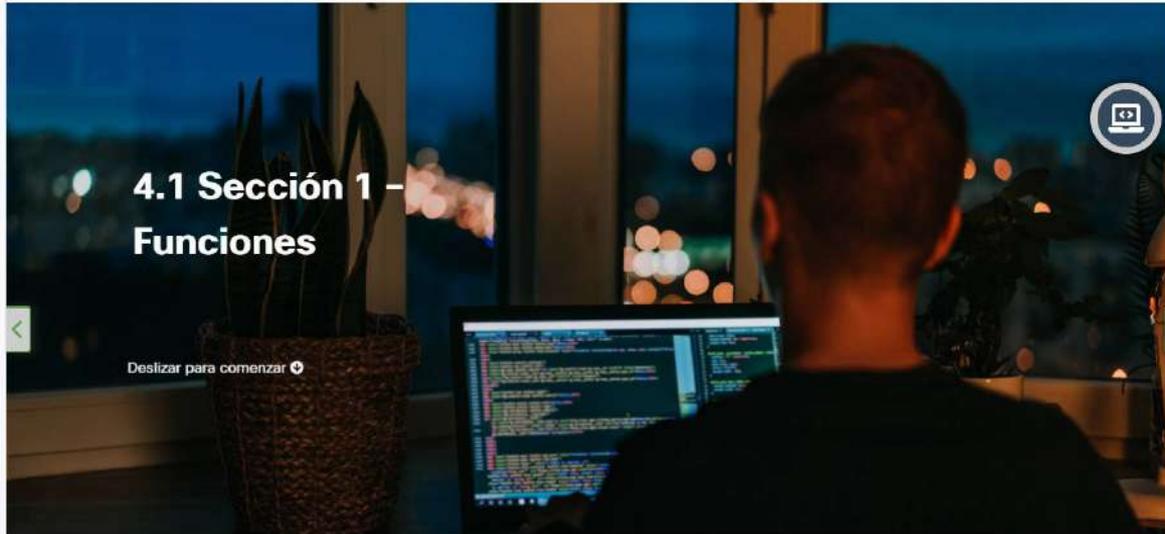
- 4.1.3 ¿De dónde provienen las funciones?
 - 4.1.4 Tu primera función
 - 4.1.5 Cómo funcionan las funciones
 - 4.1.6 RESUMEN DE SECCIÓN
 - 4.1.7 PRUEBA DE SECCIÓN
-
- 4.2. Sección 2 - Cómo se comunican las funciones con su entorno 0/7
 - 4.3. Sección 3 - Devolviendo el resultado de una función 0/10
 - 4.4. Sección 4 - Alcances en Python 0/5
 - 4.5. Sección 5: Creación de funciones con múltiples parámetros 0/7
 - 4.6. Sección 6 - Tuplas y diccionarios 0/7
 - 4.7. Sección 7 - Excepciones 0/15
 - 4.8. Finalización del Módulo 4: Prueba del Módulo

Fundamentos de Python 1 (PE1) Examen Final del Curso

Examen de certificación PCEP - Certified Entry-Level Python Programmer

4.1. Sección 1 - Funciones

🌙 🔍 🗂️ ES ✎



Buscar esquema del curso

PE1: Módulo 4. Funciones, Tuplas, Diccionarios, Excepciones y Procesamiento de Datos

5%

4.1. Sección 1 - Funciones

5 / 7

- 4.1.1 ¿Por qué necesitamos funciones?
- 4.1.2 Descomposición
- 4.1.3 ¿De dónde provienen las funciones?
- 4.1.4 Tu primera función
- 4.1.5 Cómo funcionan las funciones**
- 4.1.6 RESUMEN DE SECCIÓN
- 4.1.7 PRUEBA DE SECCIÓN

- 4.2. Sección 2 - Cómo se comunican las funciones con su entorno 0 / 7
- 4.3. Sección 3 - Devolviendo el resultado de una función 0 / 10
- 4.4. Sección 4 - Alcances en Python 0 / 5
- 4.5. Sección 5: Creación de funciones con múltiples parámetros 0 / 7
- 4.6. Sección 6 - Tuplas y diccionarios 0 / 7
- 4.7. Sección 7 - Excepciones 0 / 15

4.1.5 Cómo funcionan las funciones

4.1.5 Cómo funcionan las funciones

Observa la imagen.

Intenta mostrarte el proceso completo:

- cuando se **invoca** una función, Python recuerda el lugar donde esto ocurre y **salta** hacia dentro de la función invocada;
- el cuerpo de la función es entonces **ejecutado**;
- al llegar al final de la función, Python **regresa** al lugar inmediato después de donde ocurrió la invocación.



Existen dos consideraciones muy importantes, la primera de ellas es:

No se debe invocar una función antes de que se haya definido.

Recuerda - Python lee el código de arriba hacia abajo. No va a adelantarse en el código para determinar si la función invocada está definida más adelante, (el lugar "correcto" para definirla es "antes de ser invocada".)

Se ha insertado un error en el siguiente código - ¿puedes notar la diferencia?

```

1 print("Inicia aqui.")
2 message()
3 print("Termina aqui.")
4
5
6 def message():
7     print("Ingresar valor: ")
    
```

Buscar ejemplos del curso

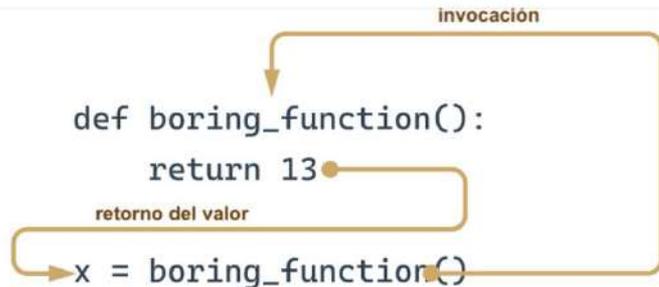
Procesamiento de Datos

13%

- 4.1. Sección 1 - Funciones 6 / 7
- 4.2. Sección 2 - Cómo se comunican las funciones con su entorno 5 / 7
- 4.3. Sección 3 - Devolviendo el resultado de una función 1 / 10
 - 4.3.1 Efectos y resultados: la instrucción return
 - 4.3.2 Unas pocas palabras sobre None
 - 4.3.3 Efectos y resultados: listas y funciones
 - 4.3.4 LAB Un año bisiesto: escribiendo tus propias funciones
 - 4.3.5 LAB Cuántos días: escribiendo y usando tus propias funciones
 - 4.3.6 LAB Día del año: escribiendo y usando tus propias funciones
 - 4.3.7 LAB Números primos - cómo encontrarlos
 - 4.3.8 LAB Conversión del consumo de combustible
 - 4.3.9 RESUMEN DE SECCIÓN
 - 4.3.10 PRUEBA DE SECCIÓN
- 4.4. Sección 4 - Alcances en Python 0 / 5
- 4.5. Sección 5: Creación de funciones con múltiples parámetros 0 / 7
- 4.6. Sección 6 - Tuplas y diccionarios 0 / 7

4.3.1 Efectos y resultados: la instrucción return

🔍 🔍 ES ✎



La instrucción `return`, enriquecida con la expresión (la expresión es muy simple aquí), "transporta" el valor de la expresión al lugar donde se ha invocado la función.

El resultado se puede usar libremente aquí, por ejemplo, para ser asignado a una variable.

También puede ignorarse por completo y perderse sin dejar rastro.

Tenga en cuenta que no estamos siendo demasiado educados aquí: la función devuelve un valor y lo ignoramos (no lo usamos de ninguna manera):

```

1 def boring_function():
2     print("¡Mado aburrimiento' ON.")
3     return 123
    
```

Buscar esquema del curso

Procesamiento de Datos

14%

- 4.1. Sección 1 - Funciones 8 / 7
- 4.2. Sección 2 - Cómo se comunican las funciones con su entorno 5 / 7
- 4.3. Sección 3 - Devolviendo el resultado de una función 2 / 10**
 - 4.3.1 Efectos y resultados: la instrucción return
 - 4.3.2 Unas pocas palabras sobre None
 - 4.3.3 Efectos y resultados: listas y funciones
 - 4.3.4 LAB Un año bisesto: escribiendo tus propias funciones
 - 4.3.5 LAB Cuántos días: escribiendo y usando tus propias funciones
 - 4.3.6 LAB Día del año: escribiendo y usando tus propias funciones
 - 4.3.7 LAB Números primos - cómo encontrarlos**
 - 4.3.8 LAB Conversión del consumo de combustible
 - 4.3.9 RESUMEN DE SECCIÓN
 - 4.3.10 PRUEBA DE SECCIÓN
- 4.4. Sección 4 - Alcances en Python 0 / 5
- 4.5. Sección 5: Creación de funciones con múltiples parámetros 0 / 7
- 4.6. Sección 6 - Tuplas y diccionarios 0 / 7

4.3.7 LAB Números primos - cómo encontrarlos

Sugerencia: intenta dividir el argumento por todos los valores posteriores (comenzando desde 2) y verifica el resto - si es cero, tu número no puede ser un número primo; analiza cuidadosamente cuándo deberías detener el proceso.

Si necesitas conocer la raíz cuadrada de cualquier valor, puedes utilizar el operador `**`. Recuerda: la raíz cuadrada de x es lo mismo que $x^{0.5}$.

Complementa el código en el editor.

Ejecuta tu código y verifica si tu salida es la misma que la nuestra.

Salida esperada:

2 3 5 7 11 13 17 19

```

1 def is_prime(num):
2     #
3     # Escribe tu código aquí.
4     #
5
6 for i in range(1, 20):
7     if is_prime(i + 1):
8         print(i + 1, end=" ")
9 print()
10

```

Console >_

Buscar asignatura del curso

Procesamiento de Datos

17%

4.1. Sección 1 - Funciones 6/7

4.2. Sección 2 - Cómo se comunican las funciones con su entorno 5/7

4.3. Sección 3 - Devolviendo el resultado de una función 6/10

4.3.1 Efectos y resultados: la instrucción return

4.3.2 Unas pocas palabras sobre None

4.3.3 Efectos y resultados: listas y funciones

4.3.4 LAB Un año bisiesto: escribiendo tus propias funciones

4.3.5 LAB Cuantos días: escribiendo y usando tus propias funciones

4.3.6 LAB Día del año: escribiendo y usando tus propias funciones

4.3.7 LAB Números primos - cómo encontrarlos

4.3.8 LAB Conversión del consumo de combustible

4.3.9 RESUMEN DE SECCIÓN

4.3.10 PRUEBA DE SECCIÓN

4.4. Sección 4 - Alcances en Python 0/5

4.5. Sección 5: Creación de funciones con multiples parámetros 0/7

4.6. Sección 6 - Tuplas y diccionarios 0/7

4.3.4 LAB Un año bisiesto: escribiendo tus propias funciones

Tu tarea es escribir y probar una función que toma un argumento (un año) y devuelve `True` si el año es un *año bisiesto*, o `False` si no lo es.

La semilla de la función ya se muestra en el código esqueleto del editor.

Nota: también hemos preparado un breve código de prueba, que puedes utilizar para probar tu función.

El código utiliza dos listas - una con los datos de prueba y la otra con los resultados esperados. El código te dirá si alguno de tus resultados no es válido.

```

1 def is_year_leap(year):
2     if year % 4 != 0:
3         return False
4     elif year % 100 != 0:
5         return True
6     elif year % 400 != 0:
7         return False
8     else:
9         return True
10
11 test_data = [1900, 2000, 2016, 1987]
12 test_results = [False, True, True, False]
13 for i in range(len(test_data)):
14     yr = test_data[i]

```

Console >_



 **Fundamentos de Python 1**

Esquema de Curso | Recursos

Proyecto Final

ES

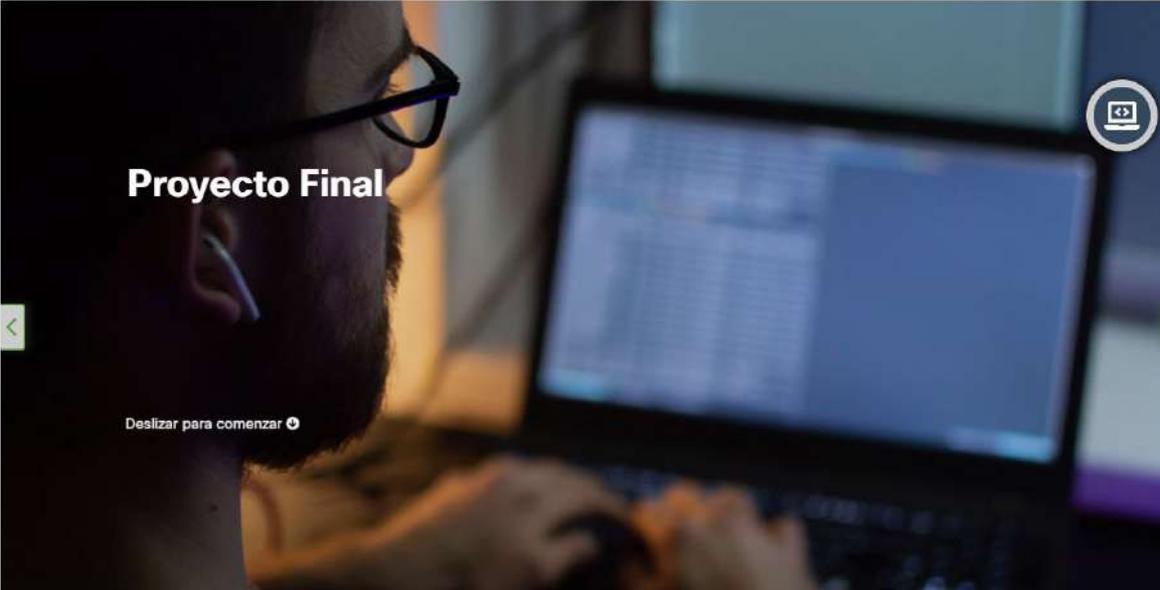
BUSCAR RECURSOS DEL CURSO

- PE1: Módulo 2. Tipos de datos, Variables, Operaciones Básicas de Entrada y Salida, Operadores Básicos 0%
- PE1: Módulo 3. Valores Booleanos, Ejecución Condicional, Bucles, Listas y su procesamiento, Operaciones Lógicas y de Bit a Bit
- PE1: Módulo 4. Funciones, Tuplas, Diccionarios, Excepciones y Procesamiento de Datos 21%
- Fundamentos de Python 1 (PE1) Examen Final del Curso 33%**
- ✓ Proyecto Final**
- Prueba Final
- Encuesta de Fin de Curso

Examen de certificación PCEP - Certified Entry-Level Python Programmer

Proyecto Final

Deslizar para comenzar



Buscar esquemas del curso 🔍

PE1: Módulo 2. Tipos de datos, variables, operaciones básicas de Entrada y Salida, Operadores Básicos



0%

PE1: Módulo 3. Valores Booleanos, Ejecución Condicional, Bucles, Listas y su procesamiento, Operaciones Lógicas y de Bit a Bit



PE1: Módulo 4. Funciones, Tuplas, Diccionarios, Excepciones y Procesamiento de Datos



21%

Fundamentos de Python 1 (PE1) Examen Final del Curso



33%

✔ Proyecto Final

➤ Prueba Final

➤ Encuesta de Fin de Curso

Examen de certificación PCEP - Certified Entry-Level Python Programmer



PROYECTO Tic-Tac-Toe

Escenario

Tu tarea es escribir **un simple programa que simule jugar a tic-tac-toe (nombre en inglés) con el usuario**. Para hacerlo más fácil, hemos decidido simplificar el juego. Aquí están nuestras reglas:

- la maquina (por ejemplo, el programa) jugará utilizando las **"x"**s;
- el usuario (por ejemplo, tu) jugarás utilizando las **"o"**s;
- el primer movimiento es de la maquina - siempre coloca una **"x"** en el centro del tablero;
- todos los cuadros están numerados comenzando con el **1** (observa el ejemplo para que tengas una referencia)
- el usuario ingresa su movimiento introduciendo el número de cuadro elegido - el número debe de ser valido, por ejemplo un valor entero mayor que **0** y menor que **10**, y no puede ser un cuadro que ya esté ocupado;
- el programa verifica si el juego ha terminado - existen cuatro posibles veredictos: el juego continua, el juego termina en empate, tu ganas, o la maquina gana;
- la maquina responde con su movimiento y se verifica el estado del juego;
- no se debe implementar algún tipo de inteligencia artificial - la maquina elegirá un cuadro de manera aleatoria, eso es suficiente para este juego.

El ejemplo del programa es el siguiente:

1	2	3
4	X	6



Esquema de Curso

Recursos

Buscar esquema del curso

PE1: Módulo 2. Tipos de Datos, Variables, Operadores Básicos de

Entrada y Salida, Operadores Básicos



0%

PE1: Módulo 3. Valores Booleanos, Ejecución Condicional, Bucles, Listas y su procesamiento, Operaciones Lógicas y de Bit a Bit



PE1: Módulo 4. Funciones, Tuplas, Diccionarios, Excepciones y Procesamiento de Datos



21%

Fundamentos de Python 1 (PE1) Examen Final del Curso



33%

✓ Proyecto Final

➔ Prueba Final

➔ Encuesta de Fin de Curso

Examen de certificación PCEP - Certified Entry-Level Python Programmer



Proyecto Final

🌙 ES ↻

Nota: la instrucción `from import` provee acceso a la función `randrange` definida en un módulo externo de Python denominado `random`.

```
from random import randrange

for i in range(10):
    print(randrange(8))
```



```
▶ ↺ ⤴ ⤵ ⌛ ⌚
```

```
1 def display_board(board):
2     # La función acepta un parámetro el cual contiene el estado actual del tablero
3     # y lo muestra en la consola.
4
5
6 def enter_move(board):
7     # La función acepta el estado actual del tablero y pregunta al usuario acerca de su movimiento,
8     # verifica la entrada y actualiza el tablero acorde a la decisión del usuario.
9
10
11 def make_list_of_free_fields(board):
12     # La función examina el tablero y construye una lista de todos los cuadros vacíos.
13     # La lista esta compuesta por tuplas, cada tupla es un par de números que indican la fila y columna.
14
15
```

Console ▶



Networking Academy | Fundamentos de Python 1

Esquema de Curso | Recursos

Prueba Final

Buscar cualquier parte del curso

- Entrada y Salida, Operadores Básicos 0%
- PE1: Módulo 3. Valores Booleanos, Ejecución Condicional, Bucles, Listas y su procesamiento, Operaciones Lógicas y de Bit a Bit
- PE1: Módulo 4. Funciones, Tuplas, Diccionarios, Excepciones y Procesamiento de Datos 21%
- Fundamentos de Python 1 (PE1) Examen Final del Curso 33%**
- Proyecto Final
- Prueba Final**
- Encuesta de Fin de Curso

Examen de certificación PCEP – Certified Entry-Level Python Programmer

Deslizar para comenzar



Buscar por tema del curso 🔍

PE1: Módulo 2. Tipos de datos, variables, Operadores básicos de Entrada y Salida, Operadores Básicos

0%

PE1: Módulo 3. Valores Booleanos, Ejecución Condicional, Bucles, Listas y su procesamiento, Operaciones Lógicas y de Bit a Bit

0%

PE1: Módulo 4. Funciones, Tuplas, Diccionarios, Excepciones y Procesamiento de Datos

21%

Fundamentos de Python 1 (PE1) Examen Final del Curso

33%

✔ Proyecto Final

🔄 Prueba Final

👉 Encuesta de Fin de Curso

Examen de certificación PCEP - Certified Entry-Level Python Programmer

0%

Pregunta 1

¿Cuál es el resultado del siguiente fragmento de código?

```

1 my_list = [1, 2]
2
3 for v in range(2):
4     my_list.insert(-1, my_list[v])
5
6 print(my_list)
7
    
```

[1, 2, 1, 2]

[1, 2, 2, 2]

[2, 1, 1, 2]

[1, 1, 1, 2]



Python 101 (PE1): Python Essentials - Part 1



Networking
CISCO Academy

General-Purpose
Programming



OpenEDG Python Institute Certification Roadmap
for the General-Purpose Programming track



Certificación PCEP™

- ✓ **Credencial:** PCEP™ – Certified Entry-Level Python Programmer
- ✓ **Código:** PCEP-30-02
- ✓ **Nivel:** Associate
- ✓ **Formato:** Preguntas tipo test | Python 3.x
- ✓ **Duración:** 40 min (examen) + 5 min (NDA/Tutorial)
- ✓ **Preguntas:** 30 preguntas
- ✓ **Passing Score:** 70%
- ✓ **Idioma:** Inglés y Español
- ✓ **Canal:** OpenEDG Testing Service – TestNow™



Certificación PCEPT™

What is the value type returned after executing the following snippet?

```
x = 0
y = 2
z = len("Python")
x = y > z
print(x)
```

- A. int
- B. float
- C. str
- D. bool
- E. NoneType



Certificación IT Specialist

- ✓ **Credencial:** IT Specialist - Python
- ✓ **Nivel:** IT Specialist
- ✓ **Duración:** 50 min (examen)
- ✓ **Preguntas:** 33 - 43 preguntas
- ✓ **Passing Score:** 70%
- ✓ **Idioma:** Inglés y Español
- ✓ **Canal:** Certiport



Python Essentials Part 2

Demo



Python 102 (PE2): Python Essentials - Part 2



Python **Essentials**
Part 2

Offered by

 **PYTHON INSTITUTE**
Open Education & Development Group

 python™





Python **Essentials 2**


Intermediate

Python Beginner/Intermediate

PYTHON ESSENTIALS 2
PYTHON 102 (PE2)

Python 102 (PE2): Python Essentials - Part 2

	PE2: Module 1. Modules, Packages, and PIP	
	PE2: Module 2. Strings, String and List Methods, Exceptions	
	PE2: Module 3. Object-Oriented Programming	
	PE2: Module 4. Miscellaneous	
	Python Essentials 2 (PE2) Course Final Exam	



Python
Essentials 2



Intermediate

Python Beginner/Intermediate

PYTHON ESSENTIALS 2
PYTHON 102 (PE2)

Catalog > Python > Fundamentos de Python 2

Fundamentos de Python 2

This course is part of the **Learning Collections - Python**

Amplíe sus habilidades de programación de computadoras utilizando Python y prepárese para obtener una certificación.

LANGUAGES AVAILABLE

English, Español.

Español (Spanish)

Get Started



Overview | Curriculum

Un sinfín de posibilidades te esperan cuando adquieres habilidades de programación en demanda. Python es un lenguaje de programación multiparadigma utilizado por nuevas empresas y gigantes tecnológicos como Google, Facebook, Netflix y más. Con una sintaxis intuitiva y legible, Python es un excelente primer lenguaje de programación que puedes aprender. Tener habilidades de Python lo califica para carreras en casi cualquier industria y es un requisito para roles de ingeniería y desarrollo de software avanzados y mejor pagados.

Este curso es el segundo de la serie de dos cursos de Fundamentos de Python y está diseñado para estudiantes que ya están familiarizados con los conceptos básicos de Python. Sumérjase en conceptos avanzados de Python, aprenda programación orientada a objetos (POO) y desarrolle su propia cartera de programación. Este curso lo prepara para la certificación **PCAP - Certified Associate in Python Programming certification** (Exam PCAP-31-0x).



FREE



40 HOURS



INTERMEDIATE



21 LABS



SELF-PACED

Achievements

Badges you can earn in this course.



Descripción general del curso

Este curso está diseñado para estudiantes que ya están familiarizados con los conceptos básicos de Python, conceptos universales de programación informática como tipos de datos, contenedores, funciones, condiciones y bucles, y los conceptos básicos de programación funcional, estructural y de procedimientos.

Beneficios

Utilizado por nuevas empresas y gigantes tecnológicos como Google, Facebook, Netflix y más, Python con código fácil de usar y de leer ofrece infinitas posibilidades para crear proyectos de software a pequeña y gran escala.

Prepárate para las carreras

- ✓ Explora el mundo de la programación
- ✓ Diseñar, desarrollar y mejorar aplicaciones informáticas de varios módulos codificadas en Python
- ✓ Analizar y modelar problemas de la vida real utilizando Programación Orientada a Objetos
- ✓ Crear y desarrollar un portafolio de programación

Detalles del curso

Público objetivo: Estudiantes de secundaria y universitarios de 2 años, público general

Tiempo estimado para completar: 40 horas

Prerrequisitos: Python Essentials 1, certificación PCEP o equivalente

Entrega del curso: Guiado por un instructor o a su propio ritmo

Aspectos destacados del componente de aprendizaje:

- ✓ 4 módulos y 21 laboratorios de práctica
- ✓ Actividades interactivas y cuestionarios
- ✓ 1 examen final
- ✓ Cupón de descuento de certificación PCAP

Reconocimientos del curso: Insignia digital

Certificación Alineada : PCAP - Certificación de Asociado Certificado en Programación Python

Próximo curso recomendado: DevNet Associate

skillsforall.com

Requisitos

- Alineación ASC: Recomendada
- Formación de instructores: Opcional
- Equipo básico: Computadora/ dispositivo móvil e Internet
- Equipo adicional requerido: No

Desarrollado en colaboración con



Esquema de Curso

Recursos

1.0. Bienvenido a Fundamentos de Python 2

🌙 🔍 🌐 ES ✎

Buscar esquema del curso

PE2: Módulo 1. Módulos, Paquetes, y PIP

27%

1.0. Bienvenido a Fundamentos de Python 2

3 / 3

1.0.1 Acerca del Curso

1.0.2 Tercero

1.0.3 Prepárate para el examen PCAP-31-0x

1.1. Sección 1 - Introducción a los módulos en Python

9 / 10

1.2. Sección 2 - Módulos Selectos de Python (math, random, platform)

0 / 9

1.3. Sección 3 - Módulos y Paquetes

0 / 5

1.4. Sección 4 - Instalador de paquetes de Python (PIP)

0 / 9

1.5. Finalización del Módulo 1 - Prueba del Módulo

PE2: Módulo 2. Cadenas, Métodos de Cadenas y listas, Excepciones

✓

PE2: Módulo 3. Programación Orientada a Objetos

✓

1.0 Bienvenido a Fundamentos de Python 2

Deslizar para comenzar



Busca: aspectos del curso

PE2: Módulo 1. Módulos, Paquetes, y PIP

20%

1.0. Bienvenido a Fundamentos de Python 2

3 / 3

1.1. Sección 1 - Introducción a los módulos en Python

3 / 10

- 1.1.1 ¿Qué es un módulo?
- 1.1.2 ¿Cómo hacer uso de un módulo?
- 1.1.3 Importando un módulo
- 1.1.4 Namespace
- 1.1.5 Importando un módulo: continuación
- 1.1.6 Importando un módulo: *
- 1.1.7 La palabra clave as
- 1.1.8 Aliasing (renombrado)
- 1.1.9 RESUMEN DE LA SECCIÓN
- 1.1.10 CUESTIONARIO DE SECCIÓN

1.2. Sección 2 - Módulos Selectos de Python (math, random, platform)

0 / 9

1.3. Sección 3 - Módulos y Paquetes

0 / 5

1.4. Sección 4 - Instalador de paquetes de Python (PIP)

0 / 9

1.1.3 Importando un módulo

Para que un módulo sea utilizable, hay que **importarlo** (piensa en ello como sacar un libro del estante). La importación de un módulo se realiza mediante una instrucción llamada `import`. Nota: `import` es también una palabra clave reservada (con todas sus implicaciones).

Supongamos que deseas utilizar dos entidades proporcionadas por el módulo `math`:

- Un símbolo (constante) que representa un valor preciso (tan preciso como sea posible usando aritmética de punto flotante doble) de π (aunque usar una letra griega para nombrar una variable es totalmente posible en Python, el símbolo se llama `pi` - es una solución más conveniente, especialmente para esa parte del mundo que ni tiene ni va a usar un Teclado Griego).
- Una función llamada `sin()` (el equivalente informático de la función matemática *seno*).

Ambas entidades están disponibles a través del módulo `math`, pero la forma en que se pueden usar depende en gran medida de como se haya realizado la importación.

La forma más sencilla de importar un módulo en particular es usar la instrucción para importar de la siguiente manera:

```
1 import math
2
```

La cláusula contiene:



Buscar esquemas del curso



PE2: Módulo 1. Módulos, Paquetes, y PIP



27%

1.0. Bienvenido a Fundamentos de Python 2

3 / 3

1.1. Sección 1 - Introducción a los módulos en Python

2 / 10

- 1.1.1 ¿Qué es un módulo?
- 1.1.2 ¿Cómo hacer uso de un módulo?
- 1.1.3 Importando un módulo
- 1.1.4 Namespce
- 1.1.5 Importando un módulo: continuación
- 1.1.6 Importando un módulo: *
- 1.1.7 La palabra clave as
- 1.1.8 Aliasing (renombrado)
- 1.1.9 RESUMEN DE LA SECCIÓN
- 1.1.10 CUESTIONARIO DE SECCIÓN

1.2. Sección 2 - Módulos Selectos de Python (math, random, platform)

0 / 9

1.3. Sección 3 - Módulos y Paquetes

0 / 9

1.4. Sección 4 - Instalador de paquetes de Python (PIP)

0 / 9

1.1.10 CUESTIONARIO DE SECCIÓN



Pregunta 1: Quieres invocar la función `make_money()` contenida en el módulo llamado `mint`. Tu código comienza con la siguiente línea:

```
import mint
```

¿Cuál es la forma adecuada de invocar a la función?

Revisar

Pregunta 2: Deseas invocar la función `make_money()` contenida en el módulo llamado `mint`. Tu código comienza con la siguiente línea:

```
from mint import make_money
```



Buscar esquema del curso

PE2: Módulo 1. Módulos, Paquetes, y PIP

41%

1.0. Bienvenido a Fundamentos de Python 2 3 / 3

1.1. Sección 1 - Introducción a los módulos en Python 8 / 10

1.2. Sección 2 - Módulos Selectos de Python (math, random, platform) 8 / 9

1.3. Sección 3 - Módulos y Paquetes 1 / 5

1.3.1 ¿Qué es un paquete?

1.3.2 Tu primer módulo

1.3.3 Tu primer paquete

1.3.4 RESUMEN DE SECCIÓN

1.3.5 CUESTIONARIO DE SECCIÓN

1.4. Sección 4 - Instalador de paquetes de Python (PIP) 0 / 9

1.5. Finalización del Módulo 1 - Prueba del Módulo

PE2: Módulo 2. Cadenas, Métodos de Cadenas y listas,

Excepciones

1.3.1 ¿Qué es un paquete?

Escribir tus propios módulos no difiere mucho de escribir scripts comunes.

Existen algunos aspectos específicos que se deben tomar en cuenta, pero definitivamente no es algo complicado. Lo verás pronto.

Resumamos algunos aspectos importantes:

- Un **módulo es un contenedor lleno de funciones** - puedes empaquetar tantas funciones como desees en un módulo y distribuirlo por todo el mundo.
- Por supuesto, no es una buena idea mezclar funciones con diferentes áreas de aplicación dentro de un módulo (al igual que en una biblioteca: nadie espera que los trabajos científicos se incluyan entre los cómics), así que se deben agrupar las funciones cuidadosamente y asignar un nombre claro e intuitivo al módulo que las contiene (por ejemplo, no le des el nombre `video_juegos` a un módulo que contiene funciones destinadas a particionar y formatear discos duros).



- Crear muchos módulos puede causar desorden: tarde que temprano querrás **agrupar tus módulos** de la misma manera que previamente has agrupado funciones: ¿Existe un contenedor más general que un módulo?
- Si lo hay, es un **paquete**: en el mundo de los módulos, un paquete juega un papel similar al de una carpeta o directorio en el mundo

Buscar respuestas del curso

PE2: Módulo 1. Módulos, Paquetes, y PIP

45%

1.0. Bienvenido a Fundamentos de Python 2

3 / 3

1.1. Sección 1 - Introducción a los módulos en Python

9 / 10

1.2. Sección 2 - Módulos Selectos de Python (math, random, platform)

8 / 5

1.3. Sección 3 - Módulos y Paquetes

3 / 5

1.3.1 ¿Qué es un paquete?

1.3.2 Tu primer módulo

1.3.3 Tu primer paquete

1.3.4 RESUMEN DE SECCIÓN

1.3.5 CUESTIONARIO DE SECCIÓN

1.4. Sección 4 - Instalador de paquetes de Python (PIP)

0 / 0

1.5. Finalización del Módulo 1 - Prueba del Módulo

PE2: Módulo 2. Cadenas, Métodos de Cadenas y listas, Excepciones

Paso 1

Imagina que en un futuro no muy lejano, tu y tus socios escriben una gran cantidad de funciones en Python.

Tu equipo decide agrupar las funciones en módulos separados, y este es el resultado final:

```

1  #! /usr/bin/env python3
2
3  """ module: alpha """
4
5  def funA():
6      return "Alpha"
7
8  if __name__ == "__main__":
9      print("Prefiero ser un módulo.")
10

```

alpha.py

Nota: hemos presentado todo el contenido solo para el módulo alpha.py, supongamos que todos los módulos tienen un aspecto similar (contienen una función denominada `funX`, donde X es la primera letra del nombre del módulo).

alpha.py

```

#! /usr/bin/env python3

""" module: alpha """

def funA():
    return "Alpha"

```

beta.py

```
def funB(): ...
```

iota.py

```
def funI(): ...
```

Buscar esquema del curso

PE2: Módulo 1. Módulos, Paquetes, y PIP



58%

- 1.0. Bienvenido a Fundamentos de Python 2 3/3
- 1.1. Sección 1 - Introducción a los módulos en Python 9/10
- 1.2. Sección 2 - Módulos Selectos de Python (math, random, platform) 8/9
- 1.3. Sección 3 - Módulos y Paquetes 4/5
- 1.4. Sección 4 - Instalador de paquetes de Python (PIP) 7/9
- 1.5. Finalización del Módulo 1 - Prueba del Módulo**

PE2: Módulo 2. Cadenas, Metodos de Cadenas y listas, Excepciones



PE2: Módulo 3. Programación Orientada a Objetos



PE2: Módulo 4. Misceláneos



Prueba Final del curso Fundamentos de Python 2 (PE2)



1.5 Finalización del Módulo 1 - Prueba del Módulo



Deslizar para comenzar



Buscar esquema del curso

PE2: Módulo 1. Módulos, Paquetes, y PIP

58%

1.0. Bienvenido a Fundamentos de Python 2 3 / 3

1.1. Sección 1 – Introducción a los módulos en Python 9 / 10

1.2. Sección 2 – Módulos Selectos de Python (math, random, platform) 8 / 9

1.3. Sección 3 – Módulos y Paquetes 4 / 5

1.4. Sección 4 – Instalador de paquetes de Python (PIP) 7 / 9

1.5. Finalización del Módulo 1 – Prueba del Módulo

PE2: Módulo 2. Cadenas, Métodos de Cadenas y listas, Excepciones

PE2: Módulo 3. Programación Orientada a Objetos

PE2: Módulo 4. Misceláneos

Prueba Final del curso Fundamentos de Python 2 (PE2)

Pregunta 18

¿Qué comando de `pip` se puede emplear para eliminar un paquete instalado?

`pip install --uninstall package`

`pip --uninstall package`

`pip remove package`

`pip uninstall package`



Buscar secciones del curso

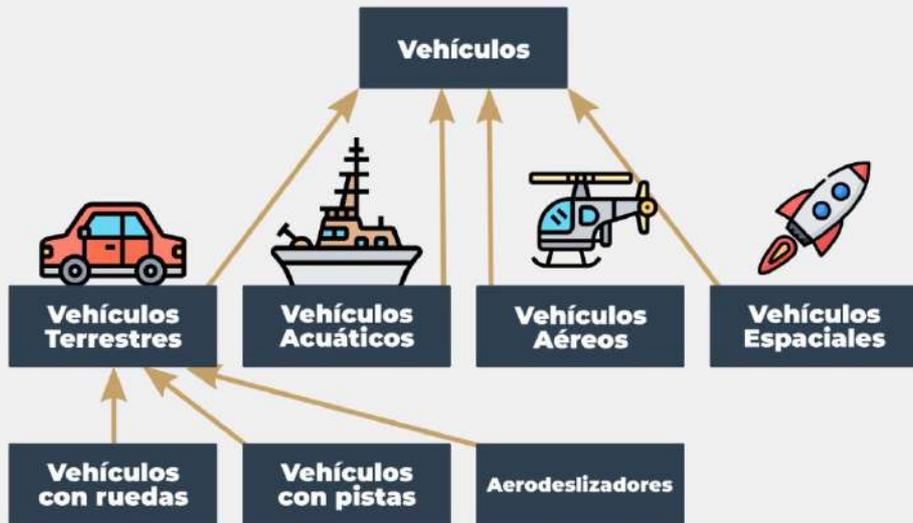
PE2: Módulo 2. Cadenas, Métodos de Cadenas y listas, Excepciones 0%

PE2: Módulo 3. Programación Orientada a Objetos 3%

3.1. Sección 1 - Los fundamentos de la POO 2 / 10

- 3.1.1 Los conceptos básicos del enfoque orientado a objetos
- 3.1.2 Enfoque procedimental versus el enfoque orientado a objetos
- 3.1.3 Jerarquía de clases
- 3.1.4 ¿Qué es un objeto?
- 3.1.5 Herencia
- 3.1.6 ¿Qué contiene un objeto?
- 3.1.7 Tu primera clase
- 3.1.8 Tu primer objeto
- 3.1.9 RESUMEN DE SECCIÓN
- 3.1.10 CUESTIONARIO DE SECCIÓN

3.2. Sección 2 - Un corto viaje desde el enfoque procedimental hacia el orientado a objetos 0 / 10



Veamos por un momento los vehículos. Todos los vehículos existentes (y los que aún no existen) están **relacionados por una sola característica importante**: la capacidad de moverse. Puedes argumentar que un perro también se mueve; ¿Es un perro un vehículo? No lo es. Tenemos que mejorar la definición, es decir, enriquecerla con otros criterios, distinguir los vehículos de otros seres y crear una



Buscar esquema del curso

- 3.2.2 La pila: el enfoque procedimental
- 3.2.3 La pila: el enfoque procedimental versus el enfoque orientado a objetos
- 3.2.4 La pila: el enfoque orientado a objetos
- 3.2.5 El enfoque orientado a objetos: una pila desde cero
- 3.2.6 RESUMEN DE SECCIÓN
- 3.2.7 CUESTIONARIO DE SECCIÓN
- 3.2.8 LAB Pila contadora
- 3.2.9 LAB Colas alias FIFO
- 3.2.10 LAB Colas alias FIFO: parte 2

- 3.3 Sección 3 - POO: Propiedades 0/5
- 3.4 Sección 4 - POO: Métodos 0/10
- 3.5 Sección 5 - Fundamentos de la POO: Herencia 0/11
- 3.6 Sección 6 - Más sobre excepciones 0/6
- 3.7 Finalización del Módulo 3 - Prueba del Módulo

PE2: Módulo 4. Misceláneos

Prueba Final del curso Fundamentos de Python 2 (PE2)

3.2.9 LAB Colas alias FIFO

Como una nueva excepción llamada `QueueError` (elige una excepción de la cual se derive) y genera cuando `get()` intenta operar en una lista vacía.

Completa el código que te proporcionamos en el editor. Ejecútalo para comprobar si tu salida es similar a nuestra.

Salida Esperada

```
Output
1
perro
False
Queue error
```

```

1- class QueueError(???): # Elige la clase base para la nueva excepción.
2- #
3- # Escribe el código aquí.
4-
5-
6- |
7- class Queue:
8-     def __init__(self):
9-         #
10-        # Escribe el código aquí.
11-
12-
13-     def put(self, elem):
14-         #

```

Console >

Buscar esquema del curso

- 3.1. Sección 1 - Los fundamentos de la POO 9 / 10
- 3.2. Sección 2 - Un corto viaje desde el enfoque procedimental hacia el orientado a objetos 7 / 10
- 3.3. Sección 3 - POO: Propiedades 3 / 5
- 3.4. Sección 4 - POO: Métodos 2 / 10
 - 3.4.1 Métodos a detalle
 - 3.4.2 La vida al interior de las clases y objetos
 - 3.4.3 Reflexión e introspección
 - 3.4.4 Investigando Clases
 - 3.4.5 RESUMEN DE SECCIÓN
 - 3.4.6 CUESTIONARIO DE SECCIÓN
 - 3.4.7 LAB La clase Timer
 - 3.4.8 LAB Días de la semana
 - 3.4.9 LAB Puntos en un plano
 - 3.4.10 LAB Triángulo
- 3.5. Sección 5 - Fundamentos de la POO: Herencia 0 / 11
- 3.6. Sección 6 - Más sobre excepciones 0 / 6
- 3.7. Finalización del Módulo 3 - Prueba del Módulo

3.4.7 LAB La clase Timer

Salida Esperada

23:59:59
00:00:00
23:59:59

Output

```

1 class Timer:
2     def __init__( ??? ):
3         #
4         # Escribir código aquí
5
6
7     def __str__(self):
8         #
9         # Escribir código aquí
10
11
12     def next_second(self):
13         #
14         # Escribir código aquí

```

Console >

Solución de muestra

Buscar esquema del curso

3.5. Sección 5 - Fundamentos de la POO: Herencia 2 / 11

3.5.1 Herencia: ¿por qué y cómo?

3.5.2 issubclass()

3.5.3 isinstance()

3.5.4 El operador is

3.5.5 Cómo Python encuentra propiedades y métodos

3.5.6 Cómo construir una jerarquía de clases

3.5.7 Herencia simple versus herencia múltiple

3.5.8 ¿Qué es el Orden de Resolución de Métodos (MRO) y por qué no todas las herencias tienen sentido?

3.5.9 El problema del diamante

3.5.10 RESUMEN DE SECCIÓN

3.5.11 QUESTIONARIO DE SECCIÓN

3.6. Sección 6 - Más sobre excepciones 0 / 9

3.7. Finalización del Módulo 3 - Prueba del Módulo

PE2: Módulo 4. Misceláneos

Prueba Final del curso Fundamentos de Python 2 (PE2)

3.5.1 Herencia: ¿por qué y cómo?

El término herencia es más antiguo que la programación de computadoras, y describe la práctica común de pasar diferentes bienes de una persona a otra después de la muerte de esa persona. El término, cuando se relaciona con la programación de computadoras, tiene un significado completamente diferente.

Definamos el término para nuestros propósitos:

La herencia es una práctica común (en la programación de objetos) de **pasar atributos y métodos de la superclase (definida y existente) a una clase recién creada, llamada subclase.**

En otras palabras, la herencia es **una forma de construir una nueva clase, no desde cero, sino utilizando un repertorio de rasgos ya definido.** La nueva clase hereda (y esta es la clave) todo el equipamiento ya existente, pero puedes agregar algo nuevo si es necesario.

Gracias a eso, es posible **construir clases más especializadas (más concretas)** utilizando algunos conjuntos de reglas y comportamientos generales predefinidos.



HERENCIA



Buscar esquema del curso

3.5. Sección 5 - Fundamentos de la POO: Herencia 2 / 11

3.5.1 Herencia: ¿por qué y cómo?

3.5.2 issubclass()

3.5.3 isinstance()

3.5.4 El operador is

3.5.5 Cómo Python encuentra propiedades y métodos

3.5.6 Cómo construir una jerarquía de clases

3.5.7 Herencia simple versus herencia múltiple

3.5.8 ¿Qué es el Orden de Resolución de Métodos (MRO) y por qué no todas las herencias tienen sentido?

3.5.9 El problema del diamante

3.5.10 RESUMEN DE SECCIÓN

3.5.11 CUESTIONARIO DE SECCIÓN

3.6. Sección 6 - Más sobre excepciones 0 / 6

3.7. Finalización del Módulo 3 - Prueba del Módulo

PE2: Módulo 4. Misceláneos

Prueba Final del curso Fundamentos de Python 2 (PE2)

3.5.1 Herencia: ¿por qué y cómo?

Así es como se ve:

```
1 | issubclass(ClassOne, ClassTwo)
```

La función devuelve `True` si `ClassOne` es una subclase de `ClassTwo`, y `False` de lo contrario.

Vamos a verlo en acción, puede sorprenderte. Mira el código en el editor. Léelo cuidadosamente.

```
1 class Vehicle:
2     pass
3
4
5 class LandVehicle(Vehicle):
6     pass
7
8
9 class TrackedVehicle(LandVehicle):
10    pass
11
12
13 for cls1 in [Vehicle, LandVehicle, TrackedVehicle]:
14     for cls2 in [Vehicle, LandVehicle, TrackedVehicle]:
```

Console >

Hay dos bucles anidados. Su propósito es verificar todos los pares de clases ordenadas posibles y que imprima los resultados de la verificación para determinar si el par coincide con la relación subclase-superclase.

Esquema de Curso

Recursos

Prueba Final

🌙 ES ↗

Buscar sugiere del curso

56%

PE2: Módulo 2. Cadenas, Métodos de Cadenas y listas, Excepciones



0%

PE2: Módulo 3. Programación Orientada a Objetos



28%

PE2: Módulo 4. Misceláneos



5%

Prueba Final del curso Fundamentos de Python 2 (PE2)



0%

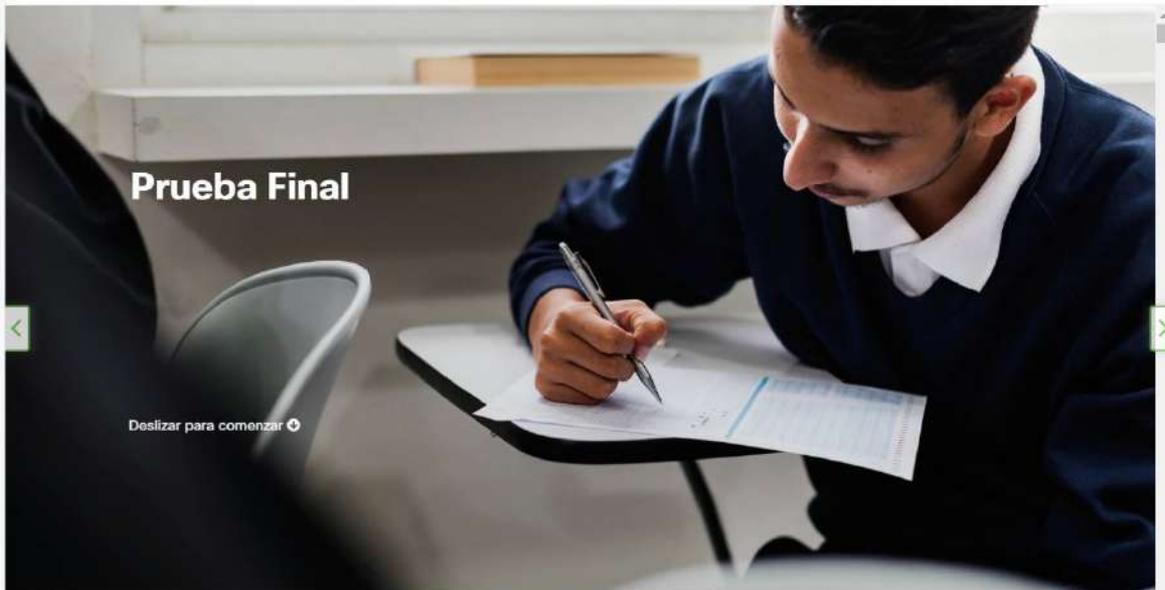
Prueba Final

Encuesta de Fin de Curso

Examen de certificación PCAP - Certified Associate in Python Programming



100%



Buscar esquema del curso



58%

PE2: Módulo 2. Cadenas, Métodos de Cadenas y listas, Excepciones



0%

PE2: Módulo 3. Programación Orientada a Objetos



28%

PE2: Módulo 4. Misceláneos



5%

Prueba Final del curso Fundamentos de Python 2 (PE2)



0%

Prueba Final

Encuesta de Fin de Curso

Examen de certificación PCAP - Certified Associate in Python Programming



100%

Pregunta 2

¿Qué resultado aparecerá después de ejecutar el siguiente fragmento de código?

```
1 import math
2 print(dir(math))
3
```

- El número de entidades que residen dentro del módulo `math`
- Un mensaje de error
- Una cadena que contiene el nombre completo del módulo
- Una lista de todas las entidades que residen dentro del módulo `math`

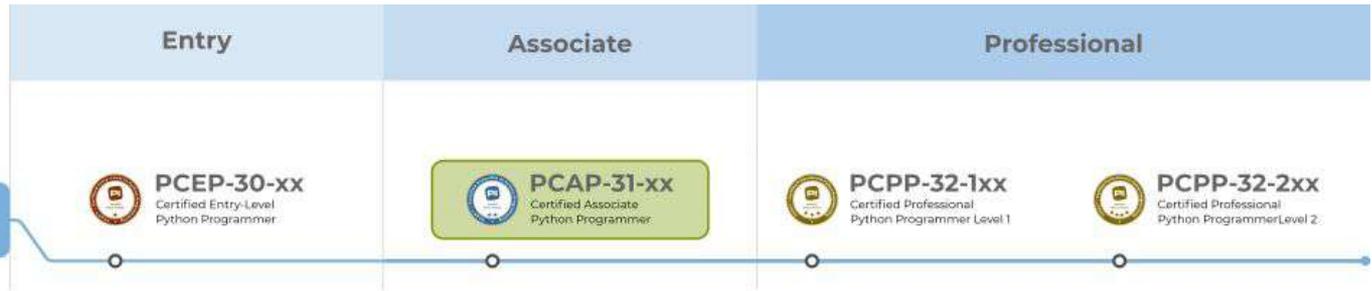


Python 102 (PE2): Python Essentials - Part 2



Networking
CISCO Academy

General-Purpose
Programming

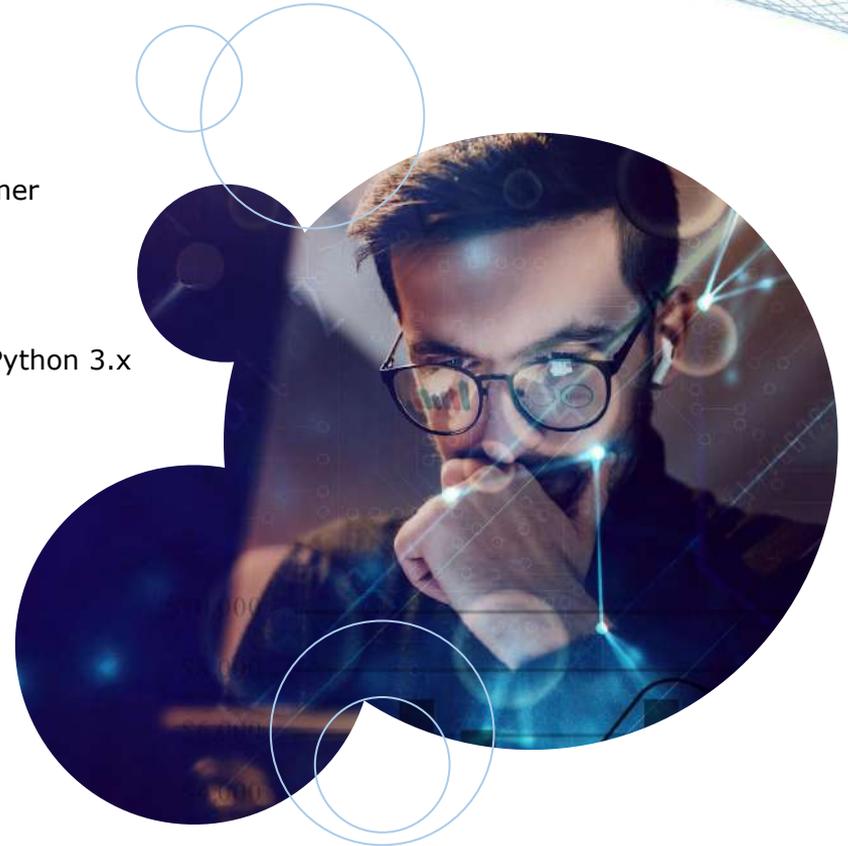


OpenEDG Python Institute Certification Roadmap
for the General-Purpose Programming track.



Certificación PCAP™

- ✓ **Credencial:** PCAP™ – Certified Associate Python Programmer
- ✓ **Código:** PCAP-31-03
- ✓ **Nivel:** Associate
- ✓ **Formato:** Preguntas tipo test, Drag&drop, Gap fill, .. | Python 3.x
- ✓ **Duración:** 65 min (examen) + 10 min (NDA/Tutorial)
- ✓ **Preguntas:** 40 preguntas
- ✓ **Passing Score:** 70%
- ✓ **Idioma:** Inglés
- ✓ **Canal:** Pearson VUE Testing Centers
OnVUE Online Proctoring



Certificación PCAP™

What is the expected output of the following snippet?

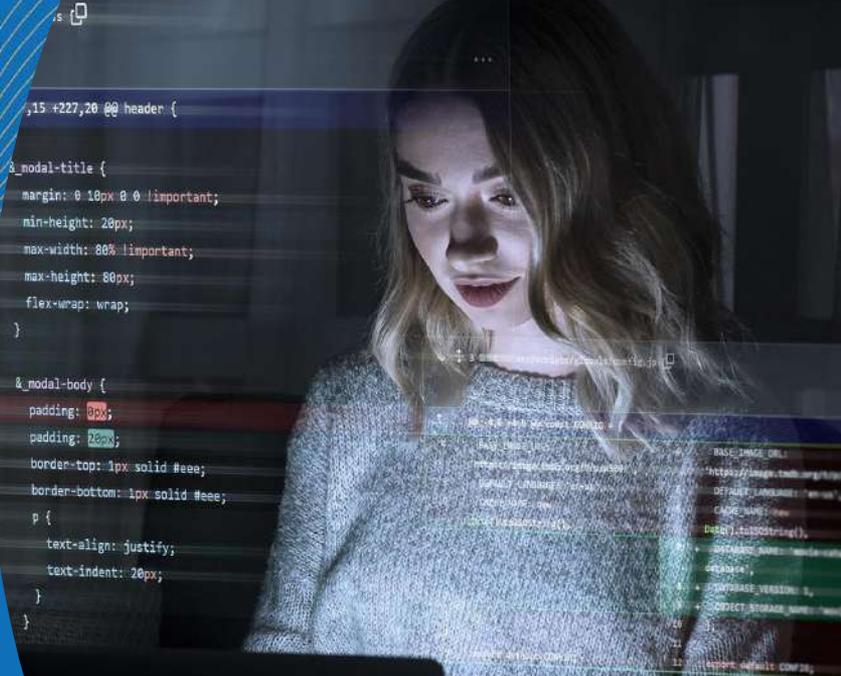
```
i = 250
while len(str(i)) > 72:
    i *= 2
else:
    i //= 2
print(i)
```

- A. 125
- B. 250
- C. 72
- D. 500



Python Institute

Plan de estudios



Python Professional 1 Course Series



General-Purpose
Programming



OpenEDG Python Institute Certification Roadmap
for the General-Purpose Programming track



Python Professional 1 Course Series



Python Essentials 1



Python Beginner

PYTHON ESSENTIALS 1
Python 101 (PE1)



Python Essentials 2



Python Associate

PYTHON ESSENTIALS 2
Python 102 (PE2)



Advanced OOP



Python Advanced

PYTHON ADVANCED 1
Python 201 (Advanced OOP)



Best Practices and Standardization



Python Advanced

PYTHON ADVANCED 2
Python 202 (PEPs)



GUI Programming



Python Advanced

PYTHON ADVANCED 3
Python 203 (GUI Programming)



Working with RESTful APIs



Python Advanced

PYTHON ADVANCED 4
Python 204 (RESTful APIs)



File Processing



Python Advanced

PYTHON ADVANCED 5
Python 205 (File Processing)



	Entry	Associate	Professional	
General-Purpose Programming	PCEP-30-xx Certified Entry-Level Python Programmer	PCAP-31-xx Certified Associate Python Programmer	PCPP-32-1xx Certified Professional Python Programmer Level 1	PCPP-32-2xx Certified Professional Python Programmer Level 2 <i>coming Q1/Q2 2024</i>
Data Science	PCED-30-xx Certified Entry-Level Data Analyst with Python <i>coming Q4 2023</i>	PCAD-31-xx Certified Associate Data Analyst with Python <i>coming Q4 2023/Q1 2024</i>		
Testing	PCET-30-xx Certified Entry-Level Tester with Python <i>coming Q4 2023/Q1 2024</i>	PCAT-31-xx Certified Associate Tester with Python <i>coming Q4 2023</i>	PCPT-32-xx Certified Professional Tester with Python <i>coming Q3/Q4 2024</i>	
Security	PCES-30-xx Certified Entry-Level Security Specialist with Python <i>coming Q1 2024</i>	PCAS-31-xx Certified Associate Security Specialist with Python <i>coming Q3 2024</i>		
Network Programming	PCEN-30-xx Certified Entry-Level Network Programmer with Python <i>coming Q2 2024</i>	PCAN-31-xx Certified Associate Network Programmer with Python <i>coming Q2/Q3 2024</i>		
Web Development	PCEW-30-xx Certified Entry-Level Web Developer with Python <i>coming Q1/Q2 2024</i>	PCAW-31-xx Certified Associate Web Developer with Python <i>coming Q3/Q4 2024</i>		

COMING SOON

Testing Essentials with Python

Beginner/Intermediate

Python Beginner/Intermediate

Testing Essentials with Python
Testing 201 (TEP)



**Facilitamos la formación
y certificación oficial en tecnología
al mundo educativo**

PUE Academy

Training & Certification Center (TCC) - Academy Support Center (ASC)

A través de nuestros servicios como TCC (Training & Certification Center) y ASC (Academy Support Center) nos ponemos a disposición de las instituciones académicas para acceder a este programa en las mejores condiciones, adaptándonos a sus necesidades y realidades.



Servicios de Formación

Dirigidos a la comunidad educativa global (staff técnico, docentes, alumnos y exalumnos)

[+ Info](#)



Servicios de certificación

Dirigidos a la comunidad educativa global (staff técnico, docentes, alumnos y exalumnos)

[+ Info](#)



Centros oficiales

PUE Academy ayuda a las instituciones educativas a convertirse en centros oficiales de formación y certificación de las organizaciones tecnológicas con más reconocimiento mundial.

[+ Info](#)

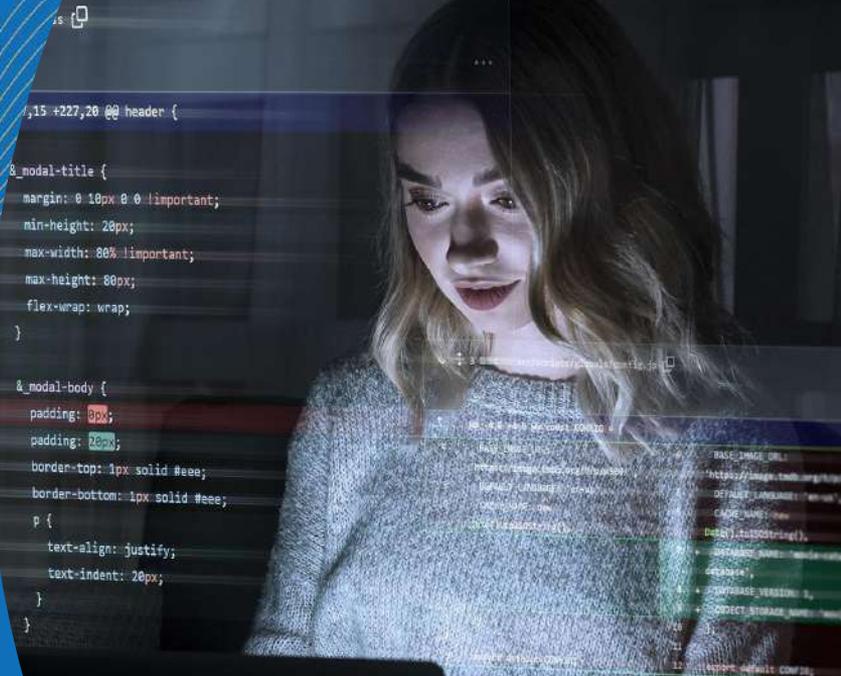


Soporte

Para la correcta implementación de las iniciativas Academy a las que se les dará acceso.

[+ Info](#)

Learn



```
15 +227,20 @@ header {
```

```
&_modal-title {  
  margin: 0 10px 0 0 !important;  
  min-height: 20px;  
  max-width: 80% !important;  
  max-height: 80px;  
  flex-wrap: wrap;  
}
```

```
&_modal-body {  
  padding: 8px;  
  padding: 20px;  
  border-top: 1px solid #eee;  
  border-bottom: 1px solid #eee;  
  p {  
    text-align: justify;  
    text-indent: 20px;  
  }  
}
```

```
BASE_IMAGE_URL  
"https://images.unsplash.com/  
DETAILS_AVAILABLE"  
CACHE_KEY  
Date().toLocaleString()  
APPBASE_NAME  
APPBASE_VERSION  
+ "DETECT_STORAGE_NAME"  
10  
11  
12 (Appbase default config)
```

Practice



Certify



Training & Certification Center

Formación oficial como garantía de calidad



**Cursos
oficiales**



**Instructores
de certificados**



**Learning
by doing**



**Orientados
a la certificación**



Presencial



Online



**Live
Virtual Class**

Training & Certification Center

Formación oficial como garantía de calidad

<p>Course</p> <p>Python Essentials 1</p> <p>Python INSTITUTE</p> <p>PYTHON ESSENTIALS 1 Python 101 (PE1)</p> <p>BEGINNER Aligned with PCEP-30-01/PCEP-30-02</p>	<p>Course</p> <p>Python Essentials 2</p> <p>Python INSTITUTE</p> <p>PYTHON ESSENTIALS 2 Python 102 (PE2)</p> <p>INTERMEDIATE Aligned with PCAP-31-02/PCAP-31-03</p>
--	--

30%
EN TODOS
NUESTROS CURSOS
OFICIALES



Certificación oficial
incluida para nuestro
colectivo PUE Alumni

NO acumulable con otras promociones

**+ Plazas de formación
incluidas en la suscripción**



21 may 2024 **CONFIRMADO** | Fundamentos de programación en Python - Curso oficial PCAP del Python Institute - Online
25 jun 2024 🕒 70 horas | 🗑️ ~~525,00€~~ 367,50 € | 📍 Online | 🗣️ Español | 🕒 Martes (16:00h - 17:30h)

03 jun 2024 **CONFIRMADO** | Fundamentos de programación en Python - Curso oficial PCAP del Python Institute
07 jun 2024 🕒 30 horas | 🗑️ ~~690,00€~~ 483,00 € | 📍 Live Virtual Class | 🗣️ Español | 🕒 De lunes a viernes (09:00h - 15:00h)

22 oct 2024 **CONFIRMADO** | Fundamentos de programación en Python - Curso oficial PCAP del Python Institute - Online
26 nov 2024 🕒 70 horas | 🗑️ ~~525,00€~~ 367,50 € | 📍 Online | 🗣️ Español | 🕒 Martes (16:00h - 17:30h)

23 oct 2024 **CONFIRMADO** | Programación avanzada en Python - Curso oficial PCPP1™ del Python Institute - Online
27 nov 2024 🕒 70 horas | 🗑️ ~~525,00€~~ 367,50 € | 📍 Online | 🗣️ Español | 🕒 Self-study

16 sep 2024 Fundamentos de programación en Python - Curso oficial PCAP del Python Institute
20 sep 2024 🕒 30 horas | 🗑️ ~~690,00€~~ 483,00 € | 📍 Live Virtual Class | 🗣️ Español | 🕒 De lunes a viernes (09:00h - 15:00h)

18 nov 2024 Fundamentos de programación en Python - Curso oficial PCAP del Python Institute
22 nov 2024 🕒 30 horas | 🗑️ ~~690,00€~~ 483,00 € | 📍 Live Virtual Class | 🗣️ Español | 🕒 De lunes a viernes (09:00h - 15:00h)

www.pue.es/pue-academy/cursos



PeopleCert ON BEHALF OF AXELOS



Training & Certification Center

PUE Certification Services



PARA CANDIDATOS

Compra y reserva individual de exámenes

Docentes, alumnos y staff técnico pueden reservar sus exámenes de forma individual y autónoma a través de nuestra [plataforma online](#) PUE Certification Services, facilitando al máximo el proceso de reserva.

PARA CENTROS

Convertirse en Centro Oficial de Certificación a través de PUE Academy

Los centros interesados en apostar por certificar a sus alumnos, como valor añadido para su futuro y prestigio para el centro, tienen la posibilidad de acreditarse como **Certiport Authorized Test Center (CATC)** mediante PUE Academy.

Acompañamiento en el proceso de certificación

PUE Academy ofrece apoyo y acompañamiento para que los centros certifiquen con éxito a sus alumnos.

Certification Credits

Vouchers de acceso a los exámenes de certificación a consumir según necesidad.

PUE Mobile Certification

Posibilidad de contratar jornadas de certificación en nuestras instalaciones o en las del centro.

Academy Support Center

Programas disponibles de PUE Academy Center



- Acceso al curso oficial Python Essentials, a través del programa Cisco Networking Academy.
- Acceso a plazas de formación para instructores incluidas según tu nivel de suscripción.
- Posibilidad de convertirse en Centro Certificador Autorizado para los exámenes IT Specialist.
- Certificación y Practice Tests de IT Specialist - Python **ilimitados**.
- Soporte de PUE Academy como ASC de Cisco para la correcta implementación del programa.

PUE Academy Award 2023

Mejor centro
Python Institute

reconoce a
Institutos Nebrija
como
Mejor centro Python Institute
Año 2023





¡Gracias!

Contáctanos. Estamos para ayudarte

 pueacademy@pue.es

 www.pue.es/pue-academy

SEDE BARCELONA

Avda. Diagonal, 98-100
08019 Barcelona
 93 206 02 49

SEDE MADRID

C/ Arregui y Aruej, 25-27
28007 Madrid
 91 162 06 69

Blog

in

X

▶