

Manual ECoS

¿Qué es la Plataforma ECoS?

La codificación es la nueva alfabetización. A medida que la tecnología va cobrando importancia en nuestra vida cotidiana, la programación se ha convertido en un elemento importante de nuestra sociedad y cada vez hay más puestos de trabajo que requieren conocimientos de programación. Los conocimientos básicos de programación también son muy útiles para comprender el funcionamiento de los ordenadores y los dispositivos tecnológicos de nueva generación, lo que supone una clara ventaja para los que tienen este conocimiento. Ya se observa una falta de programadores y esto podría convertirse en un problema en el futuro. Por eso, los conocimientos de programación son un área en la que se centran los promotores de la educación STEM.

Sin embargo, es importante apoyar el resto del plan de estudios, ya que a menudo en los cursos iniciales no hay mucho tiempo para añadir lecciones adicionales. La lengua, la historia y la cultura siguen siendo competencias importantes para los alumnos.

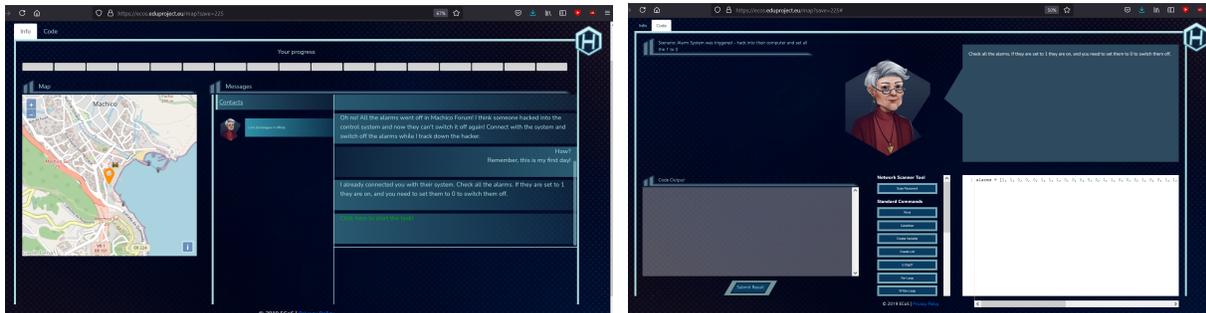
La Plataforma ECoS es una respuesta a este problema, ya que ofrece una divertida herramienta de aprendizaje basada en el juego que enseña codificación, fomenta el interés por las STEM y, al mismo tiempo, enseña la historia y la cultura regionales. Permite a los estudiantes explorar un mapa regional y revelar lugares en el mapa a los que pueden viajar a través de la codificación. Las tareas de codificación aumentan gradualmente en dificultad e introducen más y más conceptos a medida que los estudiantes avanzan en sus conocimientos de codificación. La línea de código correcta es importante y permite al jugador (los estudiantes) seguir avanzando en el juego. Este juego/plataforma de aprendizaje es, por lo tanto, capaz de enseñar habilidades de codificación a la vez que se repasa la historia y la cultura, de esta manera enseña cómo funcionan los ordenadores y, especialmente, cómo conseguir que hagan lo que se les dice, todo ello mientras se practica el pensamiento crítico y se aprende sobre la historia y la cultura regionales.

Así pues, la Plataforma ECoS (<https://ecos.eduproject.eu/>) es una herramienta de aprendizaje transversal para la lógica computacional, la codificación y la cultura local.

Mientras juegan, los alumnos se enfrentan a un entorno de aprendizaje desafiante mientras reciben el apoyo de su profesor. La narración es divertida, sencilla y atractiva para motivar a nuestros alumnos a jugar y aprender al mismo tiempo.

En la plataforma ECoS, el alumno, que es un agente de la ISA (Agencia de Seguridad Internacional), se embarca en diferentes misiones en su región de origen para evitar o resolver

ciberataques. Cada misión sólo puede resolverse mediante un código. La historia se desarrolla en el espacio de trabajo virtual del jugador, donde en la pestaña INFO interactúa con los personajes del juego y es llamado para las asignaciones asociadas a un punto en la pantalla del mapa. En la pestaña CÓDIGO, si el jugador resuelve la asignación con el código correcto, la situación se resuelve y la historia avanza. A medida que la historia avanza, surgen más situaciones que deben ser resueltas, entonces, más objetivos regionales están disponibles para que los jugadores aprendan sobre su historia e impacto cultural.



Espacio de trabajo del agente ISA.

Este juego cubre un vacío muy importante en los esfuerzos por aumentar las habilidades de codificación de los jóvenes estudiantes. Al convertirlo en un juego divertido, utiliza un tipo de aprendizaje sutil, que anima a los estudiantes a aprender mientras realizan una actividad divertida. La herramienta puede utilizarse fácilmente en clase porque se ejecuta en un navegador web y no requiere ninguna instalación. Los profesores tienen la libertad de dejar que sus alumnos aprendan a su propio ritmo y apoyarles en el juego. Esto también lo convierte en una herramienta ideal para que los alumnos aprendan por su cuenta y descubran más sobre la codificación fuera de clase.

Hay muchos proyectos a nivel nacional e internacional que intentan mejorar la educación STEM o hacerla más atractiva. Los proyectos que se centran en la tecnología se beneficiarán de esta plataforma y, en la mayoría de los casos, será muy complementaria a los nuevos enfoques de aprendizaje, los proyectos que desarrollan materiales didácticos y planes de estudio, y los proyectos que se centran en diferentes enfoques de la enseñanza de STEM.

¿Qué es la programación?

¿Quieres aprender a programar? En primer lugar, tienes que saber qué es. Una de las definiciones más completas –y a la vez más fáciles de entender– se puede encontrar en el lugar más típico, la Wikipedia:

"La programación informática es el proceso de diseñar y construir un programa informático ejecutable para lograr un resultado específico o realizar una tarea concreta. La programación implica tareas como: el análisis, la generación de algoritmos, la elaboración de perfiles de la precisión de los algoritmos y el consumo de recursos, y la implementación de algoritmos en un lenguaje de programación elegido (lo que comúnmente se denomina codificación). El código fuente de un programa se escribe en uno o varios lenguajes inteligibles para los programadores, en lugar de código máquina, que es ejecutado directamente por la unidad central de procesamiento. El objetivo de la programación es encontrar una secuencia de instrucciones que automatice una tarea en un ordenador, a menudo para resolver un problema determinado. Por lo tanto, una programación competente suele requerir conocimientos en varias materias diferentes, como el conocimiento del dominio de la aplicación, los algoritmos especializados y la lógica formal".

En resumen, la programación nos permite escribir una "receta" para que el ordenador la siga. Esta "receta" puede ser muchas cosas, como por ejemplo un juego, un procesador de textos, una tarea que necesitamos automatizar, etc., el ordenador pasará por esa "receta" ejecutándola exactamente como está escrita.

¿Qué es Python?

Python es un lenguaje de programación interpretado de alto nivel y de propósito general, cuya filosofía de diseño hace hincapié en la legibilidad del código con su notable uso de la sangría. Sus construcciones de lenguaje, así como su enfoque orientado a objetos, pretenden ayudar a los programadores a escribir un código claro y lógico para proyectos de pequeña y gran escala.

Esta filosofía de legibilidad del código está en su núcleo y se puede resumir de una manera muy elegante, de hecho se puede ver directamente en Python: solo tienes que instalar el tiempo de ejecución oficial de Python paratsu equipo, acceder a él y escribir el siguiente "import this" en la shell, para saludar con:

- Lo bonito es mejor que lo feo.
- Lo explícito es mejor que lo implícito.
- Lo simple es mejor que lo complejo.
- Lo complejo es mejor que lo complicado.
- Lo plano es mejor que lo superpuesto.
- Lo disperso es mejor que lo denso.
- La legibilidad cuenta.
- Los casos especiales no son tan especiales como para romper las reglas.
- Aunque la practicidad venza a la pureza.
- Los errores nunca deben ocurrir en silencio.
- A menos que se silencie explícitamente.
- Ante la ambigüedad, rechaza la tentación de adivinar.
- Debería haber una –y preferiblemente sólo una forma obvia de hacerlo.
- Aunque ese camino puede no ser obvio al principio, a menos que seas holandés.
- Ahora es mejor que nunca.
- Aunque nunca es mejor que *ahora mismo*.
- Si la aplicación es difícil de explicar, es una mala idea.
- Si la aplicación es fácil de explicar, puede ser una buena idea.
- Los espacios de nombres son una gran idea: ¡hagamos más de ellos!

Python fue diseñado para ser altamente extensible, por lo que, en lugar de tener toda su funcionalidad incorporada en su núcleo, utiliza módulos para aumentar sus capacidades. Esta modularidad lo ha hecho especialmente popular como medio para añadir interfaces programables a las aplicaciones existentes. Un objetivo importante de los desarrolladores de Python es que sea divertido de usar, algo que seguramente también ha ayudado a que gane fuerza en la educación. Esto se refleja en el nombre del lenguaje, que es un homenaje al grupo cómico británico Monty Python (y no al animal), este carácter lúdico también se puede ver en muchos tutoriales y materiales de referencia disponibles, donde se pueden encontrar alusiones al spam y a los huevos, de nuevo relacionadas con grupo cómico británico. También se puede ver en el Zen de los Python.

Todas las características y capacidades han permitido que Python sea constantemente clasificado como uno de los lenguajes de programación más populares.

¿Por qué Python?

Entonces, ¿por qué deberíamos elegir Python para nuestro proyecto?

Bueno, antes que nada deberías pensar en un lenguaje de programación como una herramienta que te permite crear algo, como tal deberías elegir la herramienta correcta para el trabajo que tienes entre manos. Esto también significa que Python puede no ser siempre la respuesta, por ejemplo, si estás desarrollando un juego AAA, con un fuerte uso de 3D, shaders, etc. Entonces probablemente no sea la opción más popular, aunque no es inaudito, por ejemplo el juego masivo en línea EVE Online utiliza Python.

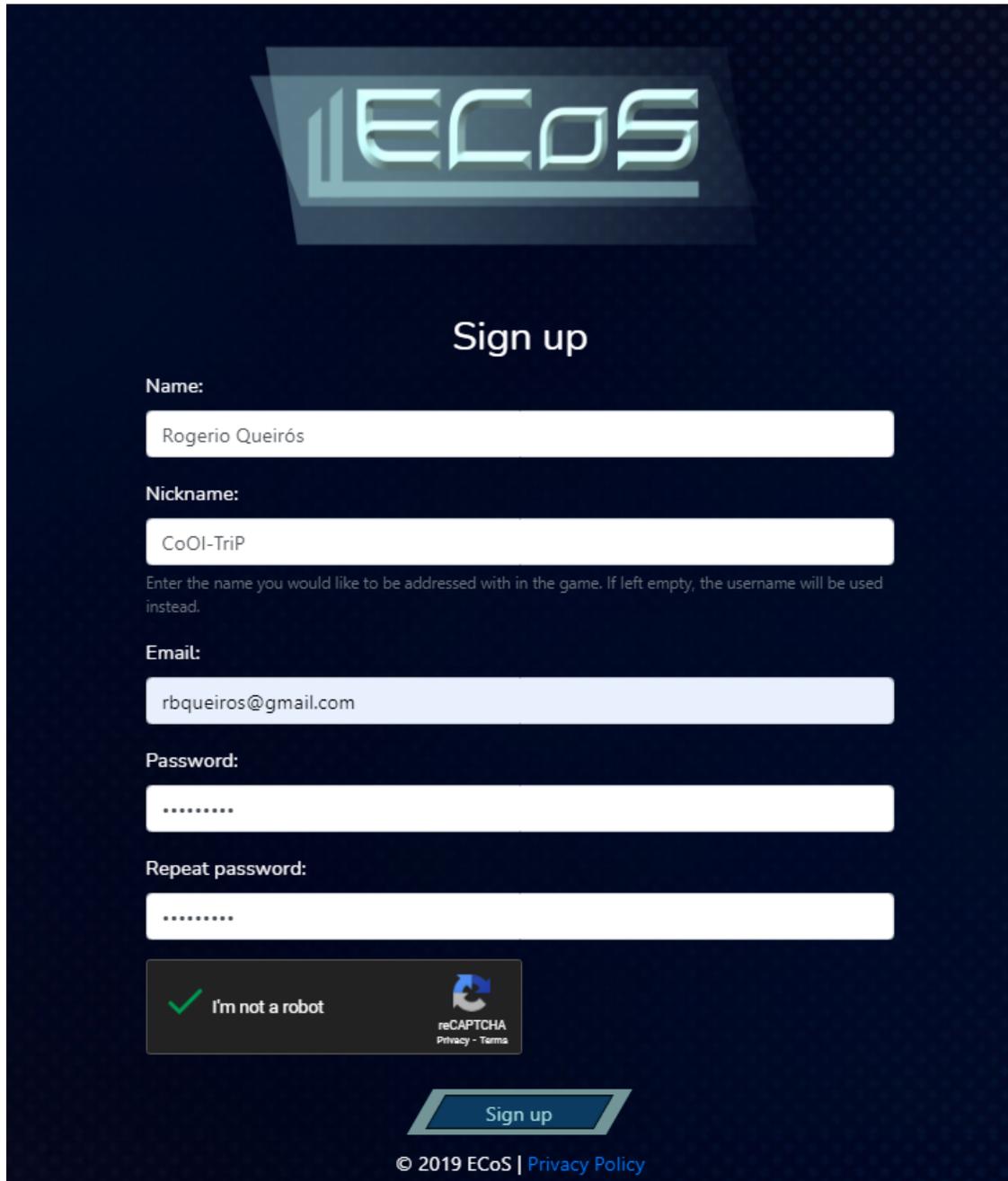
Así que vamos a repasar algunas de las razones que hacen de Python uno de los lenguajes más populares hoy en día:

1. es fácil de leer, escribir, aprender y utilizar;
2. es versátil, lo que significa que hay una gran variedad de proyectos que se pueden realizar con él;
3. es altamente flexible y extensible, desde juegos AAA hasta páginas web, puedes lograr casi cualquier cosa con él;
4. es uno de los lenguajes de programación de mayor crecimiento;
5. los programadores que lo dominan están muy solicitados;
6. la comunidad muy activa y solidaria;
7. tiene una biblioteca para cada necesidad, desde las matemáticas hasta la química, pasando por la astronomía. Python las tiene todas;
8. soporta múltiples paradigmas de programación, desde la programación orientada a objetos hasta la programación imperativa, todos pueden ser utilizados, incluso en el mismo proyecto, convirtiéndose así en una gran herramienta de enseñanza;
9. es el lenguaje de elegido para la Inteligencia Artificial y la Ciencia de Datos (extracción de datos, manipulación, etc.);
10. soporta múltiples plataformas, desde Windows, a Android, a Linux, incluso los sistemas empujados lo soportan. Puedes usar microcontroladores como el ESP32 y programarlos directamente usando Python, sin necesidad de flashearle un nuevo firmware cada vez que cambies el código;
11. es de fuente abierta;
12. Etc.

Como puedes ver, hay muchas razones para elegir Python y algunas de ellas pueden resultarte atractivas. A título personal, la legibilidad del código es algo que admiro, poder leer el código que puede haber sido escrito hace unos meses, o que un estudiante ha sometido a evaluación con poca dificultad es un gran logro de este lenguaje.

Tutorial

Registrarse en la plataforma



The image shows a registration form for the ECoS platform. At the top, the ECoS logo is displayed in a stylized, glowing font. Below the logo, the text "Sign up" is centered. The form consists of several input fields and a checkbox:

- Name:** A text input field containing "Rogerio Queirós".
- Nickname:** A text input field containing "CoOl-TriP". Below this field, there is a small note: "Enter the name you would like to be addressed with in the game. If left empty, the username will be used instead."
- Email:** A text input field containing "rbqueiros@gmail.com".
- Password:** A password input field with eight dots representing the characters.
- Repeat password:** A second password input field with eight dots.
- Verification:** A checkbox labeled "I'm not a robot" with a green checkmark, and a reCAPTCHA logo with links for "Privacy" and "Terms".

At the bottom of the form, there is a blue "Sign up" button. Below the button, the copyright notice "© 2019 ECoS | Privacy Policy" is visible.

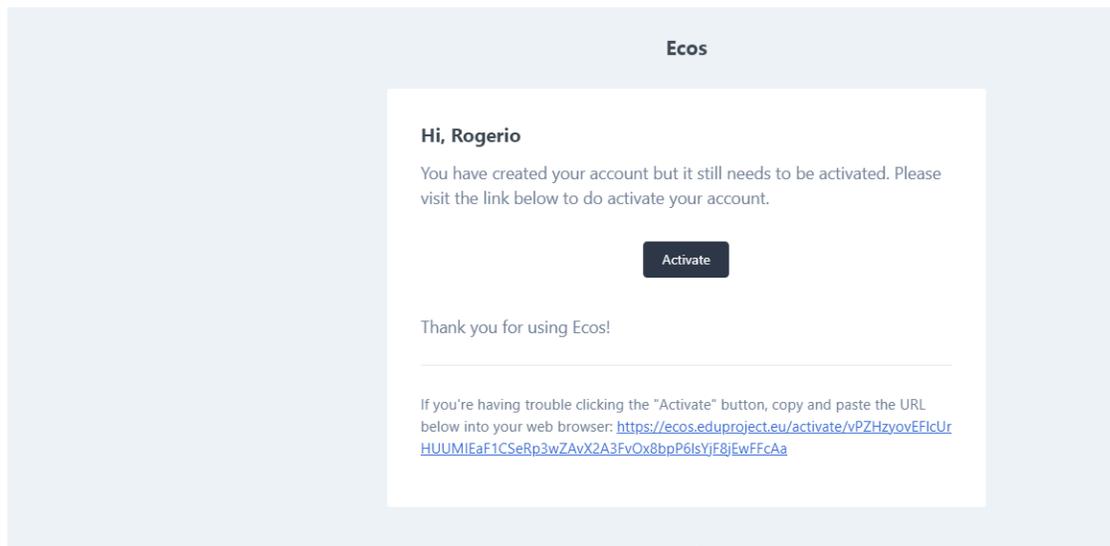
Cada jugador que se registre en la plataforma debe tener un correo electrónico válido, elegir un apodo y seleccionar una contraseña con al menos 8 caracteres

Validar la cuenta

Ecos - Activate your account Externo Caixa de entrada

Ecos System <no-reply@ecos.loc>
para mim

inglês > português Traduzir mensagem



Tras el proceso de registro, los jugadores deben activar sus cuentas entrando en su cuenta de correo electrónico y haciendo clic en "Activar" en el correo electrónico enviado por la Plataforma Ecos

Entrar en el sistema

Log in

Email:
rbqueiros@gmail.com

Password:
.....

Stay logged in
This keeps you logged in as long as you don't delete your cookies. Only use this on computers you're not sharing with anyone else.

Log in

Login using a session token

Session token:
.....

Stay logged in
This keeps you logged in as long as you don't delete your cookies. Only use this on computers you're not sharing with anyone else.

Log in

© 2019 ECoS | [Privacy Policy](#)

Los usuarios deben introducir su correo electrónico y contraseña y hacer clic en el botón "Iniciar sesión". Alternativamente, si tienen el "Token de Sesión" también pueden iniciar la sesión de esa manera.



Scenario Code

eyJpZCI6NCwibmFtZSI6IkN5YmVyIFNlY3VyaXR5IFNjZW5hcmlvIn0=

Start Scenario

There are no saved scenarios available

Continue Scenario

To access the Cyber-Security scenario, please enter this code:

eyJpZCI6NCwibmFtZSI6IkN5YmVyIFNlY3VyaXR5IFNjZW5hcmlvIn0=

You can enter this code again to delete the existing savegame and start over.

Después de iniciar la sesión, los usuarios deben copiar el código "Escenario de Ciber-Seguridad" en "Iniciar Ecenario"

Introduction

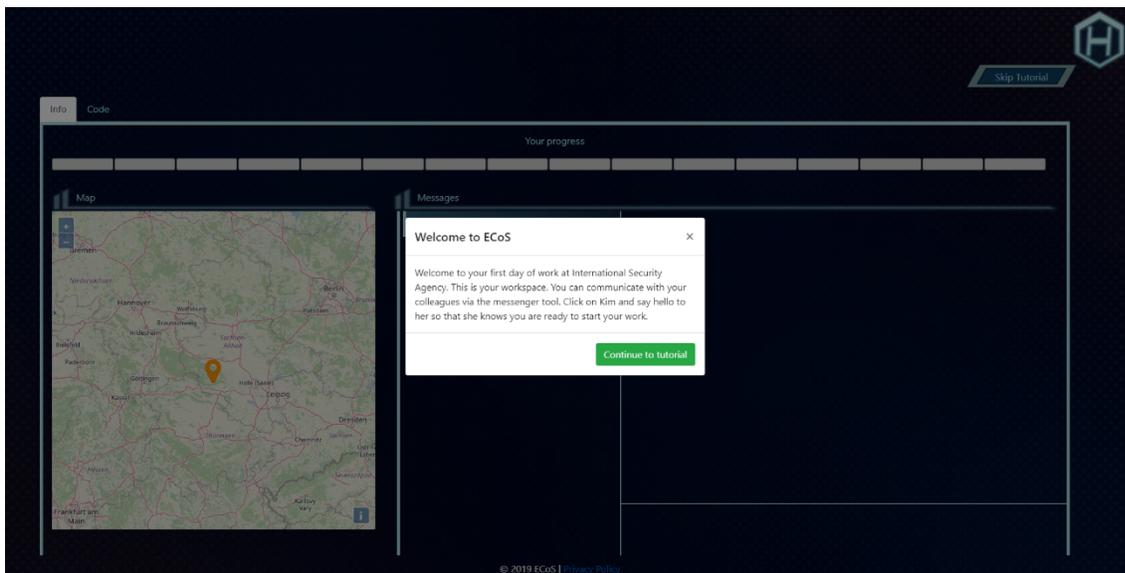
The following tutorial is going to introduce you to the basics of using ECoS as well as Python. If you feel like you do not need a tutorial, feel free to skip it using the button below.

Start

Skip Tutorial

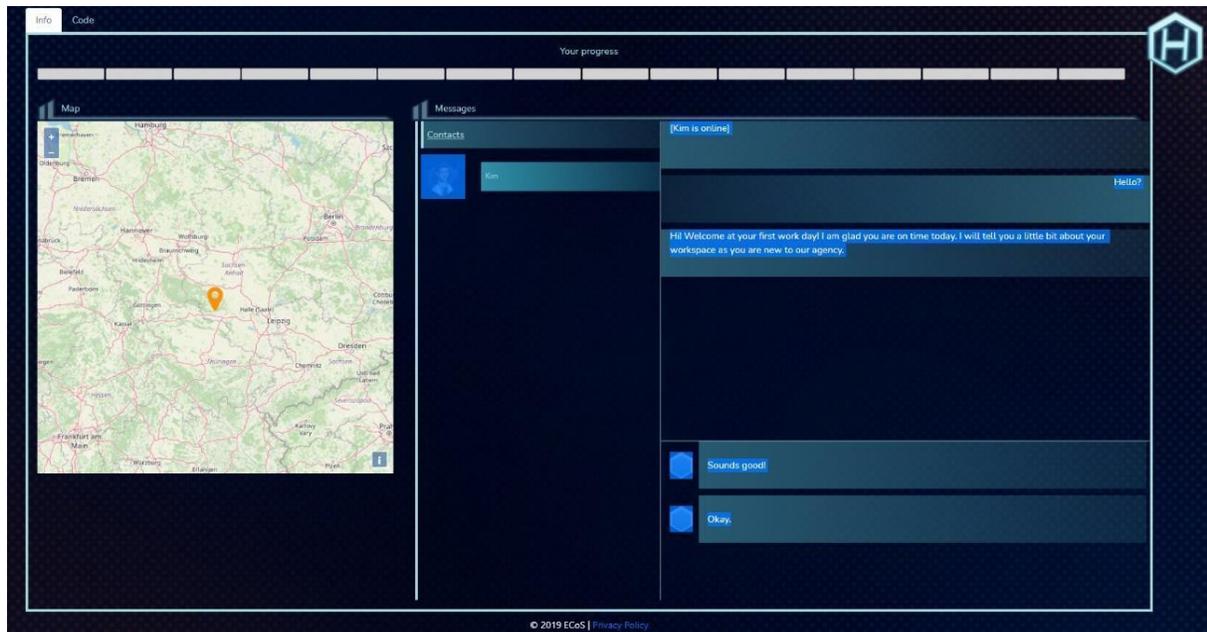
Después de iniciar la sesión por primera vez, los usuarios tendrán la posibilidad de ejecutar el tutorial del juego o empezar a jugar sin ayuda.

Escenario de la plataforma



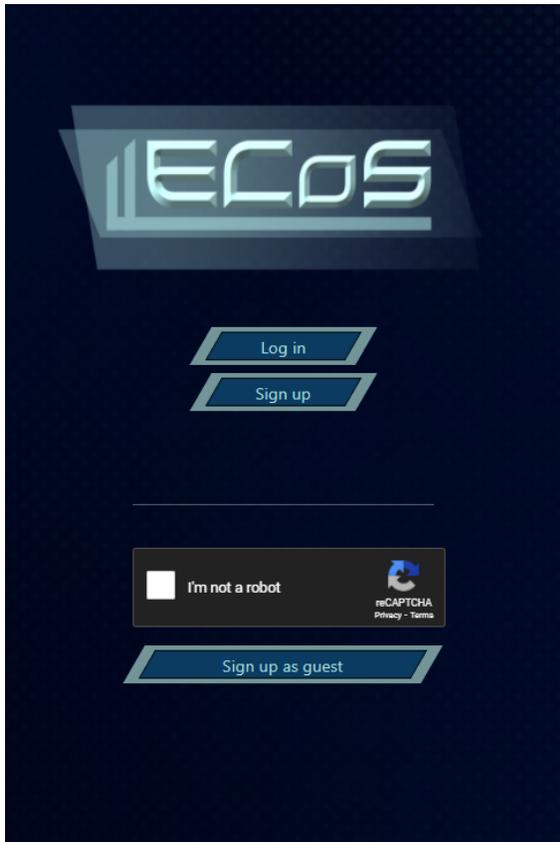
La primera vez que se accede al juego los usuarios serán recibidos con un mensaje de bienvenida

Agente KIM



Kim es la agente que te dará instrucciones a través del juego, desde el tutorial hasta los consejos para ayudar al jugador a completar sus tareas.

Registrarse como invitado



Si deseas explorar el juego sin registrarte, también puedes iniciar sesión como invitado haciendo clic en "Registrarse como invitado".



New Guest Account

Thanks for creating a new guest account. You can freely use ECoS with this account.

Below, you will see your session token. Using that token, you can log in to your guest account from any computer.

Keep in mind that if you lose this token, you will not be able to access your guest account anymore. We cannot recover guest accounts.

A randomly generated nickname was assigned to you. You can see and change it on the account settings page.

Your session token:

zJOnQBeWqrpwYzjFBqV8TKZutP51rgcL

 Copy

Make sure you save this token somewhere so you can access your account later on.

[Home](#)

Copia tu "Identificador de sesión" haciendo clic en "Copiar" y vuelve a "Inicio".



Scenario Code

zJOnQBeWqrpwYzjFBqV8TKZutP51rgcl|

Start Scenario

There are no saved scenarios available

Continue Scenario

To access the Cyber-Security scenario, please enter this code:

eyJpZCI6NCwibmFtZSI6IkN5YmVyaXR5IFNjZW5hcmlvln0=

You can enter this code again to delete the existing savegame and start over.

Pega tu "Código de Sesión" en "Código de Escenario" y haz clic en "Iniciar Escenario"

Escenarios ECOS

En esta sección te llevaremos a través de los escenarios para que sepas exactamente cuáles son las tareas y las soluciones. La sección está pensada como una guía para los profesores que quieran utilizar ECOS en su clase. Se mostrará el contenido de cada escenario junto con la solución para cada tarea de programación. Además, cada sección se comentará.

El escenario

En el juego ECOS, el jugador trabajará para la ISA, una agencia de seguridad que supervisa la seguridad de la red de la ciudad. ISA es, por supuesto, una agencia ficticia que no tiene nada que ver con el funcionamiento real de la seguridad de la red. Los escenarios se consideran una forma desenfadada y divertida de abordar este tema y el propósito es hacer que el alumno se sienta importante a pesar de ser un principiante en la codificación.

Personajes

Los personajes con los que interactúan los jugadores son:

Michael: Michael es el jefe del jugador. Está muy ocupado con las tareas de jefe por lo que no se involucrará en la aventura muy a menudo.

Piedro: Toda buena agencia necesita agentes de campo que hagan el trabajo sucio. Piedro es el agente secreto que sale a hacer ese trabajo.

Livia: Como agente de seguridad con experiencia, Livia estará atenta a la actualidad y dará tareas al jugador. Es una colega muy útil.

Escenarios

El contenido de texto en los escenarios siempre se muestra como mensajes de chat en el juego. Cualquier contenido que esté entre corchetes ([]) es texto dinámico que será sustituido por el contenido apropiado. Puede ser el nombre del jugador o el nombre de una localización. Cualquier contenido entre corchetes ({ }) indica una opción que tiene el jugador.

Introducción

Una visión general rápida:

Esto es sólo una introducción para el jugador que lanza el primer escenario de inmediato.

Texto del chat:

Livia: Hola [Nombre]. He oído que hoy es tu primer día aquí. Kim me ha dicho que ya te ha presentado nuestro espacio de trabajo. Eso es genial, significa que puedes encargarte de algunas de mis tareas.

{Wow, realmente parece ser directo}

{Seguro, ¿qué quieres que haga? }

Livia: Como sabes, somos responsables de la ciberseguridad de los lugares que se encuentran aquí. Echa un vistazo al mapa. Estos son los lugares de los que nos ocupamos. Normalmente, no hay mucho que hacer, así que coge un bocadillo y...

Livia: Oh, espera... ¡acabamos de recibir una alarma!

Objetivo:

El motivo de que la introducción sea tan breve es que el jugador se involucre en la codificación de inmediato. La introducción da a los jugadores una idea rápida de dónde están y qué se espera de ellos.

Escenario 1: Edificio público (ayuntamiento)

Una visión general rápida:

Este escenario presenta una tarea fácil en la que el alumno tiene que comparar un valor en una matriz y cambiarlo a 0 si es 1.

Texto del chat:

Livia: ¡Oh, no! Todas las alarmas han saltado en [el ayuntamiento]. Creo que alguien ha hackeado el sistema de control y ahora no pueden volver a apagarlas. Conecta con el sistema y apaga las alarmas mientras yo localizo al hacker.

{¿Cómo?}

{Recuerda, este es mi primer día}

Livia: Ya te conecté con su sistema. Comprueba todas las alarmas. Si están puestas a 1 están encendidas, y tienes que ponerlas a 0 para apagarlas.

Objetivo:

El escenario introduce una tarea fácil para que el alumno se meta rápidamente en el escenario.

Tarea

Comprueba todas las alarmas. Si están configuradas en 1, están activadas y es necesario ponerlas en 0 para desactivarlas.

```
alarmas = [1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0]
```

Solución

```
for i in range(0, len(alarmas)):  
    if alarmas[i] == 1:  
        alarmas[i] = 0
```

Escenario 2: Institución educativa

Una visión general rápida:

Esta tarea es un poco más complicada porque se basa en un bucle que recorre todos los elementos de una matriz y los convierte si no son espacios vacíos.

Texto del chat:

Livia: ¡Bien hecho! Todavía estoy rastreando al hacker. Ya sé que se conectó desde Asia. Mientras estoy ocupada, ¿podrías encargarte de la universidad local? Un amigo mío, Marcus, es el administrador del sistema allí. Vamos a enviarle un mensaje encriptado.

{¿Por qué encriptado? }
{¿Para qué? }

Livia: La [universidad] ha sido hackeada antes y quiero asegurarme de que están alerta. Sin embargo, si ya han sido hackeados, no quiero que los hackers puedan leer el mensaje. Así que encripta el mensaje, Marcus será capaz de descifrarlo.

{¿Cómo lo codifico?}
{¿Cuál es el mensaje?}

Livia: Sólo tienes que desplazar las letras del alfabeto en 2. Una A se convierte en una C, una B en una D, una C en una E, y así sucesivamente. Cuando llegues al final, vuelve a empezar por el principio. La Y se convierte en una A, la Z se convierte en una B. Envía el mensaje: "¡Alerta de hacker! Compruebe sus sistemas".

Objetivo:

Esta tarea es lo suficientemente complicada como para hacer que el alumno piense un poco. Lo ideal es que el alumno divida la tarea en partes más pequeñas y las resuelva una tras otra.

Tarea

Cifra el mensaje que se asigna en la variable "toEncrypt" e imprímelo utilizando el cifrado César. Una A se convierte en una C, una B en una D, una C en una E, y así sucesivamente. Al llegar al final se vuelve a empezar por el principio. La Y se convierte en una A, la Z se convierte en una B. Cuando el carácter sea un espacio (" "), no cambies nada de él.

```
alfabeto = ["a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z"]
```

```
toEncrypt = "alerta de hacker comprueba tus sistemas"
```

```
encryptedMessage = ""
```

Solución

```
for i in toEncrypt:
    if i != " ":
        position = alphabet.index(i)
        position += 2
        if position >= len(alphabet):
            position = 0 + position - len(alphabet)

        encryptedMessage += alphabet[position]
    else:
        encryptedMessage += i

print(encryptedMessage)
```

Livia: ¡Buen trabajo! Creo que nuestro trabajo por hoy ha terminado, ¡hasta mañana!

Escenario 3: Biblioteca

Una visión general rápida:

Este escenario es un poco más complejo y consiste en varias tareas encadenadas. El gran reto aquí es incluir un módulo diferente utilizando el comando import.

Texto del chat:

Contacto desconocido: Hola? ¿Es la ISA?

{Sí, ¿quién eres? }

{¿Cómo podemos ayudarle?}

Contacto desconocido: Este es el Escuadrón de Infiltración Lazer. Pague 1.000.000 € o se meterá en un gran problema. Esta vez no podrá defenderse de nuestros ataques.

{Sí, sí, claro}

{Déjame preguntar a mis colegas}

Contacto desconocido: Esto no es una broma.

Livia: ¡Hola! ¿Todo bien? ¿Cómo va tu trabajo?

{Acabo de recibir un mensaje de un contacto desconocido}

Livia: ¿Qué? ¿En nuestro sistema de mensajes interno? Eso no debería ser posible. Dame un momento....

Livia: Sí, ya lo veo. Hay un nuevo contacto en nuestra base de datos. ¿Cómo ha conseguido crear esa entrada de contacto?

{¿Cómo debo responder?}

Livia: De acuerdo. Primer paso en situaciones tan graves: Tratar de averiguar su dirección IP. ¿Sabes lo que es una dirección IP?

{No realmente}

{¿Qué es eso?}

Livia: Todo el mundo que navega por Internet tiene una dirección IP. Las direcciones IP pueden tener diferentes esquemas. Una dirección IPv4 tradicional consiste en 4 números con un máximo de 3 dígitos, todos separados por un punto.

Livia: Por ejemplo: 135.235.86.23. Vamos a asegurarnos de que has entendido esto. Por favor, define una variable que contenga una dirección IP válida e imprímela para mí.

Tarea

DefinE una variable que contenga una dirección IPv4 en forma de cadena e imprímala. Recuerda que consta de 4 números con un máximo de 3 dígitos, todos separados por puntos.

Solución

```
ipAddress ="132.168.23.21"  
print(ipAddress)
```

Texto del chat:

Cuando se dispone de las herramientas adecuadas y de la dirección IP de alguien, se puede rastrear a las personas y averiguar desde dónde están conectadas. Por ejemplo, la policía y nosotros podemos averiguar una posición más o menos exacta del usuario.

{¿Cómo puedo hacer eso?}

Livia: Tenemos todo tipo de herramientas secretas. Una de ellas es el dispositivo de rastreo de IP. Tenemos el nombre de usuario, él está usando "nightwolf". Extraño nombre de usuario. Ese tipo debe ser un auténtico bicho raro.

Livia: Utiliza la función "read_ip(string ip)" del dispositivo de seguimiento ip e introduce el nombre de usuario exacto como parámetro. El dispositivo de seguimiento comprobará nuestros registros y devolverá su IP.

Tarea

Define una variable que contenga el nombre de usuario del hacker "nightwolf". Llama a la función read_ip() del IP-Tracking-Device. Revisa las opciones a tu izquierda. Ahí puedes seleccionar fácilmente la función read_ip(). Todo lo que necesitas hacer es introducir el nombre de tu variable como parámetro.

Imprime la IP que devuelve el dispositivo de seguimiento.

Solución

```
variable_name = "nightwolf"

print(ip_tracking_device.read_ip(variable_name));
```

(Importante aquí: Definir la variable e imprimir el resultado del rastreador, que debe ser correcto. El resultado que devuelve en "nightwolf" es "183.42.58.12".

Texto del chat:

Livia: ¿Pudiste conseguir la IP?

{Sí. La IP es: 183.42.58.12}

Livia: De acuerdo. ¡Bien! Parece que todavía está conectado. Parece estar en la [BIBLIOTECA]. Tratemos de hackearlo rápido antes de que se desconecte de su red.

{Bien. ¿Cómo lo harás?}

Livia: Tú lo harás. Usa el módulo de comprobación de puertos. Recorre todo tipo de puertos diferentes para asegurarte de que encuentras uno que podamos usar para entrar en su ordenador. Lo usaré y colocaré un troyano allí.

Tarea

Inicia el módulo Port-Checker. Llama a su función `set_ip(ip)` y pásale la IP que hayas resuelto. Recuerda: La IP debe ser proporcionada como una cadena. Recuerda: La IP es: 183.42.58.12

Utiliza un bucle `for` para recorrer los números del 1 al 99999. Pasa cada número al módulo de comprobación de puertos `is_port_vulnerable(port)`. Imprime el número si es un puerto vulnerable.

Código oculto:

```
import random
class PortCheckerModule:
    def __init__(self):
        self.ip = ""
    def set_ip(self, ip):
        self.ip = ip
    def is_port_vulnerable(self, port):
```

```
if self.ip == "183.42.58.12":
    if port == 553 or port == 145 or port == 86744:
        return True
    else:
        return False
elif self.ip=="":
    return "Error: No IP set"
else:
    rand = random.randint(1, 40)
    if rand == 5:
        return True
    else:
        return False
```

Solución

```
port_checker_module = PortCheckerModule()
port_checker_module.set_ip("183.42.58.12")
for i in range(1, 99999):
    if port_checker_module.is_port_vulnerable(i):
        print(i)
```

Texto del chat:

Livia: ¿Has encontrado algunos puertos vulnerables que pueda utilizar?

{Sí. 553, 145 y 86744}

Perfecto. He colocado un código en su ordenador. Por favor, hazme un favor. Ahora deberías poder utilizar la funcionalidad del troyano. Tiene una función llamada `get_recent_files()`. Por favor, deja que liste todos los archivos del escritorio para ti e imprime sus nombres.

Tarea

Ahora que el troyano está instalado en el ordenador del hacker, puedes utilizar sus funciones para acceder al ordenador. Utiliza el comando `get_recent_files()` del troyano, que devuelve una lista y la asigna a una variable. Haz un bucle a través de la lista e imprime todos los nombres de archivos de la lista.

Código oculto

`#Trojan`

```
import random
class Trojan:
    def get_recent_files(self):
        files = ["secretTarget.txt", "girlsnightout.mp4", "frozen2.mp4", "iwannacry.mp4",
"how_to_apply_blocks.pdf", "how_to_get_a_girlfriend.pdf"]
        return files
trojan = Trojan()
files = trojan.get_recent_files()
for i in files:
    print(i)
```

Solución

```
recentFiles = trojan.get_recent_files()
for i in recentFiles:
    print(recentFiles[i])
```

FILES

```
secretTarget.txt
girlsnightout.mp4
frozen2.mp4
iwannacry.mp4
how_to_apply_blocks.pdf
how_to_get_a_girlfriend.pdf
```

Objetivo:

La razón por la que esta sección está dividida en varias tareas es que se debe introducir al alumno en el concepto de llamar a una clase diferente de forma gradual. La idea es también que el alumno repita esta tarea para adquirir más confianza en su uso.

Escenario 4: Instalación deportiva

Una visión general rápida:

Esta tarea vuelve a ser sencilla. Lo único que tiene que hacer el alumno es filtrar el código no deseado de una cadena.

Texto del chat:

¿Funcionó?

{Sí, encontré algunos archivos muy extraños. Hay un archivo llamado secretTarget.txt}

Livia: Oh. Eso no puede ser bueno. Déjame comprobar el archivo.... oh....

{¿Qué?}

{¿Malas noticias?}

Livia: Su próximo objetivo es [Instalación deportiva]

Livia: Deberíamos ponerles un cortafuegos para asegurarnos de que nadie pueda hackearlos.

Tarea

Parece que el hacker quiere atacar [Instalación deportiva]. Es hora de establecer una protección de cortafuegos para ellos. Escribe un algoritmo que elimine todo lo que no sean ceros y unos de la cadena de entrada. Cada vez que lo pruebes, la cadena tendrá un valor diferente, así que asegúrate de que tu programa puede manejar eso. Declara una segunda variable que contenga el valor de la primera variable, pero sin nada excepto 0 y 1. Imprime esa cadena.

Solución

```
request =  
"100101010001010worm?hack0101010101101malware0010110100110101virus101011010010  
100101010001111010101101010011"
```

```
result = ""
```

```
for c in request:
```

```
    if c == "0" or c == "1":
```

```
        result += c
```

```
print(result)
```

Objetivo:

Se trata de una tarea alentadora que el alumno puede resolver fácil y rápidamente. Se coloca después del escenario 3, más complejo, para animar de nuevo al alumno.

Escenario 5: Iglesia

Una visión general rápida:

Este escenario retoma la tarea del escenario 1 y la hace ligeramente más compleja.

Texto del chat:

Gracias. Espero que sean lo suficientemente seguros como para resistir sus ataques.

Livia: Acabo de recibir una llamada de la iglesia. Sus campanas han empezado a sonar y el sistema electrónico que las controla está atascado. Debe haber sido el hacker de nuevo, me pregunto qué está tratando de hacer.

{¿Tal vez sólo quiere molestar a la gente?}

{¿Tal vez nos está probando?}

Tal vez. Pero debemos actuar y poner fin a esto. Conecta con el sistema de control y apaga las campanas.

Tarea

Comprueba la lista de dispositivos conectados. Si el dispositivo es una campana de iglesia, apágalo en la lista deviceStates. 0 en la lista deviceStates significa apagado, 1 en la lista deviceStates significa encendido.

```
connectedDevices = ["churchbell", "toaster", "fridge", "churchbell", "speaker", "churchbell"]
deviceStates = [1, 1, 1, 1, 0, 1]
```

Solución

```
For i in connectedDevices:
```

```
    If connectedDevices[i] == "churchbell":
```

```
        If deviceStates[i] == 1:
```

```
            deviceStates[i] = 0
```

```
        deviceStates[i] = 0
```

Objetivo:

La idea es desafiar al alumno tomando algo con lo que ya debería estar familiarizados y elevando un poco el nivel de dificultad.

Escenario 6: Economía / Banco

Una visión general rápida:

Esta tarea consiste en comparar dos matrices y calcular la diferencia de valor.

Texto del chat:

Michael (Jefe): [Nombre]! ¿Qué está pasando? ¡El banco ha sido hackeado! ¿Por qué no hiciste nada para evitarlo?

{Perdón, estaba ocupado apagando las campanas de la iglesia...}

Michael (Jefe): ¿Por qué te centras en la iglesia? A nadie le importa que suenen las campanas, a no ser que sea el domingo a las 10 de la mañana y tú estés todavía en la cama.

Livia: [Nombre], parece que el hacker nos tendió una trampa para distraernos. Al parecer, nunca se planeó que [instalación deportiva] fuera el objetivo... y el momento de las campanas de la iglesia no fue ninguna coincidencia.

{Demonios...}

{¿Qué debemos hacer ahora?}

Livia: Creo que el hackeo logró realizarse. El hacker parece haber robado mucho dinero. Creo que es un gran experto. Por favor, ayuda al banco a averiguar cuánto dinero ha robado. Al menos deberíamos saberlo, ¡y luego deberíamos intentar atraparlo!

Tarea

Calcula la suma del dinero que falta en los diferentes grupos de cuentas del banco. Compara las listas `afterHack` y `beforeHack` y sus números en las posiciones individuales. Si los números en la misma posición de ambas listas son diferentes, suma la diferencia a la suma de dinero faltante. Imprime la suma de dinero que falta.

Solución

```
afterHack = [15500, 2400, 10000]
```

```
beforeHack = [21000, 3020, 13040]
```

```
sum = 0
```

```
for i in range(0, afterHack.length):
```

```
    if(beforeHack[i] != afterHack[i]):
```

```
        sum += beforeHack[i] - afterHack[i]
```

```
print(sum)
```

Objetivo:

Esta tarea introduce cálculos con variables que son un poco complejos, pero que siguen siendo fáciles. El alumno no debería asustarse por nada de esto.

Escenario 7: Transporte

Una visión general rápida:

Una vez más, la tarea consiste en recorrer una matriz y hacer una comparación.

Texto del chat:

Livia: Eso es bastante dinero. Tenemos que atrapar a esta persona, o al menos conseguir más información.

Livia: He analizado un poco más el ordenador del hacker y he averiguado su nombre. Además, descubrí que aparentemente entre el ataque a [edificio público] y el ataque a [banco] viajó a esta ciudad.

Se llama Mario.

{¡Oh, eso es bueno! }

{Genial. Como Super Mario}

Livia: Yeaah.... Sin embargo, mi idea. Ya que llegó aquí hace poco, podríamos ir a [Transporte] y buscar su nombre en las últimas listas de pasajeros. Probablemente uno de los Marios sea él. Eso nos dará más información sobre él.

{Wow! Eres muy inteligente.}

{¡Hagamos esto! }

Tarea

Recorre la lista de pasajeros y comprueba si el nombre es "Mario". En caso afirmativo, comprueba la misma posición de la lista de pasaportes e imprímela.

```
passengersList = ["Mario", "Luigi", "Carlo", "Isabella", "Liseth", "Maja", "Gabriel", "Santos",  
"Cristina"]
```

```
passportList = ["A92fKsk2", "Kfask201", "kf0kak2D", "kbcA12d9", "Ake012F34", "Yke92A1f",  
"Rks18dA1"]
```

Solución

```
For i in range(0, passengersList.len):  
    If passengersList[i] == "Mario":  
        print(passportList[i]);
```

Objetivo:

El alumno

Escenario 8: Parques / naturaleza

Una visión general rápida:

En este escenario, el alumno tendrá que trabajar con posiciones en una matriz para descifrar un mensaje cifrado.

Texto del chat:

Livia: Bien. Ahí tenemos a nuestro Mario...

Creo que nuestro objetivo acaba de conectarse al Wi-Fi público del parque.

{¡Ni siquiera sabía que teníamos Wi-Fi público en el parque!}

{Menos mal que no usó una VPN.}

Livia: Sí, no suelo sugerir el uso del Wi-Fi del parque, no es muy seguro, como puedes ver. Puedes controlar el tráfico y escuchar sus mensajes?

{Veré lo que puedo hacer}

Tarea

Nightwolf utiliza el wifi público de [Park Location] para enviar mensajes encriptados a alguien. Parece que ha utilizado el cifrado César que también habíamos utilizado antes cuando enviamos nuestros avisos a nuestro amigo. Esta vez, sin embargo, es al revés: el mensaje no puede leerse hasta que cambiemos todas las letras del mensaje por un número determinado. Te sugiero que pruebes diferentes números y veas cuál funciona para desenscriptar el mensaje. Imprime el mensaje después de desenscriptarlo

encryptedMessage = "Fcw dpgclbq! G ugjj zc fyaigle JMAYRGML gl hsqr y dcu fmspq! Qryw
rslcb dmp qmkc pcyj yargml. Mf, ylb zw rfc uyw, G zmsefr y rgaicr dmp Kyazcrf yr RFCYRCP
dmp rmlgefr. G fmnc rfcw nsr ml y emmb npmbsargml."

Solución

```
Alphabet = ["a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v",  
"w", "x", "y", "z"]
```

```
decryptedMessage = ""
```

```
For i in encryptedMessage:
```

```
    If i != "!" && i != " ":
```

```
        Position = alphabet.index(i)
```

```
        Position -= 2
```

```
        If position < 0:
```

```
            Position = alphabet.length() + position
```

```
        decryptedMessage.push(alphabet[position])
```

```
    Else:
```

```
        decryptedMessage.push(i)
```

```
print(decryptedMessage)
```

```
print(decryptedMessage)
```

Texto del chat:

Livia: ¿Pudiste descifrar el mensaje?

```
{ "¡Hola amigos! Voy a hackear la LOCACIÓN en sólo unas horas! Estad atentos para ver algo  
de acción real. Ah, y por cierto, he comprado una entrada para Macbeth en el TEATRO para  
esta noche. Espero que hagan una buena producción". }
```

Objetivo:

Esta tarea es un rompecabezas muy complicado para los alumnos. Se basa en algo conocido para que el alumno tenga algún tipo de orientación. Sin embargo, el descifrado es todo un reto y el alumno puede necesitar ayuda en este aspecto. El objetivo de este ejercicio es llevar a los alumnos al límite y desafiarlos

Escenario 9: Teatro

Una visión general rápida:

Este es un problema de lógica fácil que requiere un pensamiento lógico.

Texto del chat:

Livia: ¡Oh, no, son muy malas noticias! Pero al menos hemos descubierto sus planes para esta noche. Ah, tengo una gran idea. Enviaremos a un agente a sentarse en el asiento de al lado para ponerle un dispositivo de seguimiento. Sólo tenemos que reservar a nuestro agente un asiento junto a él.

{¿Cómo lo hacemos? }

{¡Estoy en ello! }

Livia: Conéctate al sistema de reservas del teatro. Nightwolf está en el asiento 16. Cambia la reserva del asiento 15 a "Lisa Fernandez". Esa es nuestra agente.

Tarea

Cambia la reserva del asiento 15 a nombre de nuestro agente en [Teatro]. Ya te he registrado en su sistema de reservas. Si ya hay una persona asignada al asiento 15, por favor, asigna esa persona a otro asiento que esté libre: somos hackers, pero no somos gente mala.

```
Seats = ["Tim Knopf", "Harry Hopp", "Gustavo Salerno", "Elder Ferreira", "Pio Cerqueira", "Edgar Simões", "free", "Alfonso del Valle", "Miguel Angel Alonso", "free", "Sergio Villarreal", "Jiří Čech", "Rudolf Král", "free", "Sebastian Souček", "Mario Whitehead", "Felix Schmidt", "free", "free"]
```

Solución

```
occupiedName = ""
```

```
If seats[14] != "free":
```

```
    occupiedName = seats[14]
```

```
    Seats[14] = "Lisa Fernandez"
```

```
If occupiedName != "":
```

```
    For i in seats.length:
```

```
        If seats[i] == "free":
```

```
            Seats[i] = occupiedName
```

Objetivo:

El propósito de este ejercicio es desafiar las habilidades de pensamiento lógico.

Escenario 10: Mercado

Una visión general rápida:

Este ejercicio introduce las funciones y se centra en la comparación de datos.

Texto del chat:

Livia: [¡Nombre!] Nuestro dispositivo de rastreo nos muestra que Nightwolf está actualmente en el mercado. Pedro fue allí para observarlo, y puede ver que está hablando con alguna otra persona. Sin embargo, no puede ver el aspecto de esa persona, ¡porque está demasiado lejos!

Livia: Afortunadamente tenemos cámaras de seguridad de alta tecnología que pueden utilizarse para identificar a esa persona. Por favor, entra en la cámara de seguridad X23M1 y utiliza sus funciones de identificación para obtener más datos.

{¿Cómo lo hago? }

{¿Crees que ya estoy tan cómodo con tus herramientas de alta tecnología?}

Livia: Te mostraré cómo hacerlo.

Tarea

Accede a la cámara de seguridad "X23M1" utilizando el comando `security_cameras.connect_to("camera id")`. Utiliza el comando `security_cameras.get_target_feature("feature_type")` para obtener las características específicas de la persona con la que se encuentra el hacker. Las características posibles son "hair_color", "eye_color", "hair_type", "skin_color" y "features". Compara estos rasgos con nuestra base de datos de delincuentes. Si una de las posiciones de la base de datos tiene todas las características introducidas, imprime su nombre.

```
Known_criminals_name = ["John", "Gabriela", "Lissy"]
Known_criminals_hair_color = ["red", "blonde", "brown"]
Known_criminals_eye_color = ["green", "blue", "brown"]
Known_criminals_skin_color = ["dark", "light", "light"]
Known_criminals_features = ["none", "skar", "none"]
```

Objetivo:

Este ejercicio permite al alumno utilizar funciones...

Escenario 11: Suministro de energía y agua

Una visión general rápida:

En esta sección el alumno creará una clase y la llamará en el código principal.

Texto del chat:

Livia: Deberíamos crear una herramienta de alarma para [Power Supply], ya que no sabemos la hora exacta en la que ataca. Como tenemos nuestro dispositivo de seguimiento en Nightwolf, debería ser bastante fácil para nosotros hacerlo.

{Bien, me encargaré de esto}

Livia: Suena bien. Comprueba las funciones que están disponibles para el dispositivo de seguimiento. Deberían ser suficiente ayuda para crear una herramienta. Asegúrate de activar la alarma cuando esté a menos de 500 metros de [Fuente de alimentación] y enviaremos una unidad policial especial.

Tarea

Creará una herramienta de alarma basada en el dispositivo de seguimiento. La herramienta de alarma necesita ser una clase con la función "check_status" que se asegure de que Nightwolf no está a menos de 500 metros de distancia de la LOCALIZACIÓN. Si está dentro del rango de 500 metros, devuelve 1. Si no está dentro del rango de 500 metros, devuelve 0. Ya he preparado una clase vacía para ti. Introduce el nombre de la ubicación, "LOCATION", como parámetro para la función check_distance_to() del dispositivo de seguimiento. Imprímelo cuando se llame a "check_status". Llama a la función fuera de la clase para probar si funciona.

Solución:

```
class Alarm(object):
    def __init__(self, ipTracker):
        self.ipTracker = ipTracker
    def check_status():
        // Enter Code here
```

```
alarm = Alarm.new(ipTracker)
```

Objetivo:

Este ejercicio permite al usuario crear una clase y, por tanto, introduce una parte avanzada de la codificación.

Escenario 12: Festival / Eventos

Una visión general rápida:

En este ejercicio, el alumno utilizará el método de hackeo por fuerza bruta y escribirá un bucle sencillo.

Texto del chat:

Livia: ¡Nightwolf ha atacado de nuevo! Ha pirateado los altavoces del festival y exige un millón de euros o arruinará definitivamente el [monumento]. Todo el mundo puede oír su voz, y están bastante asustados. Tenemos que volver a controlar los altavoces para que no pueda asustar a más gente.

{Me pondré a apagarlas.}

{¡Maldición! Esperaba poder dormir una siesta. Oh, bueno.}

Tarea

Escribe un algoritmo que averigüe el código correcto para acceder de nuevo al altavoz. El código es un número de 4 dígitos. Imprime todas las combinaciones posibles.

Código oculto

```
# Simple Password Generator
def generate_simple_passcode():
    import random
    number_range_min = 1
    number_range_max = 10000
    random_password = random.randint(number_range_min, number_range_max)
    return random_password
password = generate_simple_passcode()
print(password)
```

Solución

```
for i in range(1, 10000):
    print(i);
```

Objetivo:

Esto debería resultar fácil para el alumno. La tarea es una introducción al hacking por fuerza bruta.

Escenario 13: Monumento

Una visión general rápida:
Este ejercicio exige una estructura if/elif ampliada.

Texto del chat:

Livia: Bien, mientras apagabas los altavoces he avisado a la policía, que ya ha llegado al monumento.

Livia: El problema es que parece que no pueden acceder a los controles de la bomba. Sin embargo, he conseguido conectar con la bomba, ya que está conectada a Internet. Desgraciadamente no puedo entender su mecanismo. ¿Tal vez tengas una idea de cómo funciona?

{Haré todo lo posible, jefe.}
{Será una bomba que hay que desctivar}

Tarea

¡Alguien puso una bomba para nosotros! Desactiva la bomba. Asegúrate de imprimir los colores correctos en el orden correcto. La bomba tiene una secuencia de colores que debe ser contestada con la respuesta correcta. Hay 4 variables que definen la respuesta para cada color. Imprime "rojo" si la secuencia es amarilla. Imprime "amarillo" si la secuencia es azul. Imprime "verde" si la secuencia es roja. Imprime "azul" si la secuencia es verde.

```
sequences = ["yellow", "red", "red", "green", "blue", "red", "yellow", "red", "yellow"]
```

Solución

```
for i in sequences:  
    if i == "yellow":  
        print("red");  
    elif i == "blue":  
        print("yellow");  
    elif i == "red":  
        print("green");  
    elif i == "green":
```

```
print("blue");
```

Objetivo:

Desde el punto de vista del código, este ejercicio debería ser muy sencillo para el alumno. Es sobre todo un ejercicio de pensamiento lógico.

Escenario 14: Museo

Una visión general rápida:

En esta última tarea, el alumno tiene que enfrentarse a una versión ampliada del método de hackeo por fuerza bruta.

Texto del chat:

¡Increíble! La policía no pudo atrapar a Nightwolf antes de que notara el dispositivo de rastreo. Huyó al museo y consiguió encerrarse dentro. Todas las puertas están selladas electrónicamente. Tendrás que encontrar el código de las cerraduras para que la policía pueda entrar y detenerlo. ¡Sé que puedes hacerlo! Las cerraduras usan un código de 4 dígitos. ¡Deprisa!

(Este tipo realmente me está haciendo trabajar duro.)

(¿Podemos ir a tomar un helado después?)

Tarea

Escribe un algoritmo que averigüe el código correcto para las puertas del museo. El hacker parece haber mejorado su método de contraseña. Ahora utiliza el siguiente formato: Number(1-9) Number(1-9) Letter (a-z). Escribe un algoritmo que compruebe todas las versiones posibles de la contraseña para encontrar la correcta y abrir la puerta

Código oculto:

```
def generate_improved_passcode():  
    import random  
    letters = ["a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t", "u",  
              "v", "w", "x", "y", "z"]  
    number_range_min = 1  
    number_range_max = 99  
    random_password = str(random.randint(number_range_min, number_range_max)) +  
    letters[random.randint(0, len(letters)-1)]  
    return random_password
```

```
password = generate_improved_passcode()
print(password)
```

Solución

```
letters = ["a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l",
           "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z"]
```

```
for a in letters:
    for b in range(1, 99):
        print(str(b) + a)
```

Texto del chat:

Livia: ¡Uf, eso estuvo cerca! Gracias a tu duro trabajo, la policía ha entrado en el museo y ha arrestado al alborotador Nightwolf y a sus compañeros hackers. Se acabó el internet para ellos allí donde van! Lo has hecho muy bien. Menos mal que estás de nuestro lado.

{Todo en un día de trabajo.}
{¿Así que eso es un "sí" al helado?}

Objetivo:

Esta tarea pone fin al escenario. El ejercicio se basa en el método de hacking de fuerza bruta utilizado anteriormente y lo amplía a un escenario más complicado. El propósito aquí es desafiar al alumno de nuevo a un rompecabezas de pensamiento lógico y al mismo tiempo proporcionar un elemento familiar.

FIN

	Tarea	Contexto
Edificios públicos (ayuntamiento, etc.)	El sistema de alarma se activó; hackear su ordenador y poner todos los 1 a 0	Un hacker está poniendo a prueba los mecanismos de defensa de la ciudad
Institución educativa	Enviar aviso encriptado (cifrado César)	Tememos que la institución educativa también fue hackeada y queremos advertir al administrador del sistema
Biblioteca	Hackear el ordenador de un hacker por primera vez Buscar en el catálogo de libros y encontrar el libro que el hacker sacó	El hacker se pone en contacto con la agencia secreta y quiere dinero. Sin embargo, en los registros internos se puede rastrear su IP. Hackeamos al hacker usando nuestro IP-Checker.
Instalación deportiva	Establecer un cortafuegos	Al escanear el ordenador de los hackers, la agencia encuentra los archivos privados del hacker. Uno de ellos es "next_target.txt". En ese documento, la instalación deportiva aparece como objetivo. La

		agencia decide que debe establecer un cortafuegos para la instalación deportiva.
Iglesias	Apagar los timbres.	El hacker utiliza el hackeo de la iglesia como distracción. Mientras los agentes se ocupan de las campanas, el hacker ataca el banco.
Economía / Banco	Comprueba cuánto dinero ha robado el hacker	El hacker roba el banco en lugar de los dos primeros objetivos
Transporte	Compruebe la lista de pasadores para el nombre del hacker	Como descubrimos que el hacker se llama Mario, comprobamos todas las listas de pasajeros del último vuelo en busca del nombre Mario.
Parques / naturaleza	El hacker utiliza una wifi pública y envía mensajes cifrados a su organización. Descifra los mensajes	Descubrimos que el hacker planea sabotear infraestructuras críticas. Subproducto: Notamos que el hacker compró entradas para el teatro por la noche.
Teatro	Buscar el número de asiento	Un agente de campo pone un dispositivo de seguimiento a un hacker
Mercado	Cámaras de seguridad: obtienen diferentes rasgos de una persona y los cotejan con la base de datos de delincuentes	El hacker se reúne con sus cómplices, los identificamos
Suministro de energía y agua	Alerta de proximidad; crear una herramienta de alarma	Estamos alertados y tenemos que proteger las infraestructuras críticas. El hackeo se acerca pero nuestro sistema de alarma funciona y la policía es enviada.
Fiestas / Eventos	Recuperar el acceso a una lista de ponentes pirateados en el evento	El hacker piratea los altavoces de una fiesta para hablar con el público, exigiendo dinero y amenazando con que volará el monumento local con una bomba si no consigue lo que quiere.
Monumento	Señales de retorno basadas en las instrucciones para desactivar una bomba que alguien preparó para nosotros	El grupo de hackers recurre a bombardear el monumento. Nos damos cuenta y desactivamos la bomba.
Museo	El hacker bloquea todas las puertas - código de 4 dígitos para averiguarlo	El hacker se vuelve loco y se da cuenta del dispositivo de seguimiento. Antes de que la policía pueda atraparlo, huye al museo y sella todas las puertas. Hay que volver a abrirlas para ayudar a la policía a atraparlo.

```

class IpTrackingDevice:
    def read_ip(self, parameter):
        if parameter == "nightwolf":
            return "183.42.58.12"
        else:
            return "Couldn't find ip"

```


¿Cómo utilizar ECOS en el aprendizaje formal, no formal e informal?

La plataforma ECOS no sólo puede utilizarse en un aula o contexto escolar formal, sino también en la educación no formal e informal.

¿Cuál es la diferencia entre educación formal, no formal e informal?

La educación formal es la que está regulada y reconocida, tiene pruebas que garantizan la competencia y los conocimientos y tiene una estructura vertical. Es la educación obligatoria, la EFP y la educación universitaria, por ejemplo.

La educación no formal está estructurada, es intencional y también tiene objetivos de aprendizaje y evaluaciones, pero en este caso es voluntaria y cada participante decide hasta qué punto se involucra y también se encarga de su propia evaluación.

La educación informal es la que se produce de forma involuntaria, sin que la persona sea consciente de que está aprendiendo. Cuando jugamos estamos aprendiendo reglas y estrategias, y a menudo ni siquiera somos conscientes de ello.

En la educación informal, podemos, por ejemplo, ofrecer a los jóvenes de un centro juvenil la posibilidad de jugar a ECOS. De este modo, les permitimos aprender por su cuenta. Sin embargo, es aconsejable estar disponible en caso de que tengan preguntas o necesiten algún apoyo. Lo importante es esperar a que pidan ayuda en lugar de dirigir su aprendizaje.

En la educación no formal, el juego ECOS puede utilizarse por sí solo o también en el contexto de un aprendizaje más amplio (por ejemplo, dentro de un intercambio de jóvenes en el marco del programa Erasmus+).

-El facilitador pensará en cómo quiere utilizar el juego, si utilizar todo el juego o sólo una parte.

Si los participantes no saben nada de programación, es aconsejable empezar con el tutorial.

-Puedes proponer un aprendizaje individual o intentar resolver las tareas en parejas o grupos.

-Puedes añadir un elemento competitivo entre los grupos, como puntos o premios por completar las tareas, pero no es necesario.

Al final del proceso es útil hacer un momento de reflexión con los participantes para compartir cómo fue la experiencia del juego. Algunas preguntas para el informe podrían ser:

-¿Qué dificultades encontraste y cómo las afrontaste?

-¿Qué partes fueron las más motivadoras y cuáles las más frustrantes? ¿Cómo has afrontado esa frustración?

-Cuando colaborabas en grupo, ¿qué papel tenías dentro del grupo, había un líder, cómo se tomaban las decisiones?

-Por último, debes terminar con una evaluación, ya sea de la sesión de ECOS o de todo el programa de formación, y también con una reflexión sobre lo que han aprendido de esta experiencia.

¿Cómo utilizar la ECoS en el aprendizaje formal?

Sin duda, todas las organizaciones de educación formal, como las escuelas, los institutos y las universidades de distintos países, tienen varios niveles de flexibilidad en sus planes de estudio. Todo depende de la organización y del profesor. Sin embargo, podría haber dos opciones, entre otras, de cómo incluir el ECoS en su sistema:

- Mira desde el punto de vista de los resultados de aprendizaje de la unidad y concéntrate en las habilidades que se desarrollan al jugar al juego ECoS. Puedes introducir este juego en la clase de informática, en las TIC, en las clases de programación, mientras se aprende lógica o incluso en los aspectos de descubrimiento cultural mientras se resuelve el misterio. Puedes planificar un par de lecciones con una tarea y una presentación sobre el viaje, los retos y las soluciones que tomaron, porque en este juego puede haber diferentes soluciones al problema. Los alumnos pueden presentar su forma de pensar.
- Invita a los alumnos a obtener créditos extra jugando a este juego en su tiempo libre como actividades extraescolares (clubes especiales).

Habilidades desarrolladas:

- Digital
- Ordenador
- Programación / codificación
- Lógica
- Investigar
- Resolución de problemas
- Toma de decisiones
- Descubrimiento multicultural
- Lengua extranjera

Lo mejor es que lo pruebes tú mismo y veas las soluciones más adecuadas para ti y tus alumnos.

PREGUNTAS FRECUENTES

¿Dónde puedo jugar a la Plataforma ECoS?

En cualquier navegador web en la siguiente URL: <https://ecos.eduproject.eu/>

¿Qué pasa si mi aula no tiene suficientes dispositivos (ordenadores y tabletas)?

Incluso un dispositivo puede hacer posible el uso de la Plataforma ECoS en su aula. Por ejemplo, puedes proyectar la pantalla y jugar con la clase.

Aplicando la estrategia de "aprender compartiendo" puedes crear un grupo para la clase si tienes acceso a dispositivos limitados.

Recuerda preguntar y buscar en nuestra web (<http://earlycodinginschools.eu>) cuando busques ideas. Además, nos encantaría que nos contaras si quieres compartir algo sorprendente que se te haya ocurrido.

No me siento cómodo con mis conocimientos de programación, pero me gustaría utilizar la Plataforma ECoS en mi aula.

El tutorial de la Plataforma ECoS es una forma de practicar sus habilidades de codificación empezando desde cero. Te introduce en los fundamentos del uso de ECoS Platform así como de Python. Sigue las instrucciones de Kim y lo lograrás.

¿Puedo pasar directamente al juego de ECoS sin hacer primero el tutorial?

Sí. Siéntete libre de hacerlo y desafía tus habilidades de codificación como agente de la ISA.

Al codificar, ¿los espacios son diferentes de las tabulaciones?

Cuando uses la indentación, ten cuidado de usar siempre una u otra, nunca las mezcles en tu proyecto. La convención estándar establece que se deben utilizar 4 espacios para la indentación y la mayoría de los editores de python incluso llegan a sustituir automáticamente los tabuladores por 4 espacios al hacer la indentación.

Fallo. Mi código no es correcto aunque la ventana de la consola dice "El programa se ha ejecutado correctamente".

Este mensaje significa que tu código es correcto, sin embargo no resuelve la tarea. Vuelve a leer el problema y comprueba si tu código tiene sentido.

¿Se evalúa a los alumnos mientras juegan con la plataforma ECoS?

No. A medida que los alumnos resuelven sus tareas, avanzan en la narración, aprendiendo más sobre la codificación y la historia y cultura locales.

Se puede realizar una evaluación informal comprobando su barra de progreso en el espacio de trabajo del juego, que muestra cuántas tareas ha resuelto el alumno.

He encontrado un error en la Plataforma ECoS. ¿Dónde puedo informar?

Por favor, escríbenos un correo electrónico a support@ingeniousknowledge.com

O ponte en contacto con nosotros a través de nuestra página web:

<http://earlycodinginschools.eu>

Tu contribución será muy apreciada.

Glosario

“>>>”

- El prompt de Python por defecto del shell interactivo. A menudo se ve para los ejemplos de código que se pueden ejecutar de forma interactiva en el intérprete.

abs

- Devuelve el valor absoluto de un número.

argument

- Información adicional que el ordenador utiliza para ejecutar comandos

assignment

- Dar un valor a una variable.

block

- Sección del código que se agrupa

break

- Se utiliza para salir de un bucle for o de un bucle while.

class

- Una plantilla para crear objetos definidos por el usuario.

compiler

- Traduce un programa escrito en un lenguaje de alto nivel a un lenguaje de bajo nivel.

debugging

- El proceso de encontrar y eliminar errores de programación.

def

- Define una función o método

dictionary

- Una matriz (array) asociativa mutable (o diccionario) de pares de claves y valores. Puede contener tipos mixtos (claves y valores). Las claves deben ser de tipo hashable.

evaluation order

- Python evalúa las expresiones de izquierda a derecha. Observa que al evaluar una asignación, el lado derecho se evalúa antes que el lado izquierdo.

for

- Itera sobre un objeto iterable, capturando cada elemento a una variable local para su uso por el bloque adjunto

function

- Una secuencia parametrizada de declaraciones.

function call

- Una invocación de la función con argumentos.

high level language

- Diseñado para ser fácil de leer y escribir para los humanos.

IDLE

- Entorno de desarrollo integrado

if statement

- Ejecuta condicionalmente un bloque de código, junto con else y elif (una contracción de else-if).

immutable

- No se puede cambiar después de su creación.

import

- Se utiliza para importar módulos cuyas funciones o variables pueden utilizarse en el programa actual.

indentation

- Python utiliza la sangría de espacio en blanco, en lugar de llaves o palabras clave, para delimitar los bloques.

interpret

- Ejecuta un programa traduciéndolo línea a línea.

iterable

- Un objeto capaz de devolver sus miembros de uno en uno.

list

- Lista mutable, puede contener tipos mixtos.

methods

- Un método es como una función, pero se ejecuta "sobre" un objeto.

object

- Cualquier dato con estado (atributos o valor) y comportamiento definido (métodos).

object-oriented

- permite a los usuarios manipular estructuras de datos llamadas objetos para construir y ejecutar programas.

PEP 8

- Un conjunto de recomendaciones sobre cómo escribir código Python.

slice

- Subpartes de las secuencias

strings

- Pueden incluir números, letras y varios símbolos y estar encerradas entre comillas dobles o simples, aunque las comillas simples son más utilizadas.

variables

- Marcador de posición para textos y números. El signo de igualdad (=) se utiliza para asignar valores a las variables.

while

- Ejecuta un bloque de código siempre que su condición sea verdadera.