

ECoS Handbook

O que é a plataforma ECoS?

Programação é a nova literacia. À medida que a tecnologia tem ganhado importância no nosso quotidiano, a programação tornou-se mais relevante na nossa sociedade e como consequência, tem aumentado o número de trabalhos que requer competências de programação.

Compreender o funcionamento dos computadores e da próxima geração de dispositivos tecnológicos requer conhecimentos básicos de programação, o que concede uma distinta vantagem a quem os detém. Atualmente, o mercado de trabalho tem falta de programadores e este facto pode tornar-se um problema no futuro. Para contornar esta situação, os agentes educativos das STEM (Ciência, Tecnologia, Engenharia e Matemática) têm concentrado parte das suas ações na promoção de competências em programação. Sem deixar de lado as outras áreas disciplinares, tais como as línguas, a história e a cultura, que possuem competências importantes para os nossos alunos.

A plataforma ECoS apresenta-se como uma resposta a esta problemática porque oferece um divertido jogo didático que ensina a programar, encoraja o interesse nas STEM, desenvolve a prática do pensamento crítico e ao mesmo tempo, ensina história e cultura regional. Permite ao jogador (aluno) explorar um mapa regional e revelar localizações no mapa para onde poderão viajar através da programação. As tarefas de programação aumentam gradualmente de dificuldade e introduzem mais conceitos à medida que os alunos avançam na sua aprendizagem de programação. A linha de código correta é importante, pois só assim se avança no jogo. Esta plataforma de aprendizagem é, por isso, capaz de ensinar competências de programação com história e cultura local, assim como ensinar como funcionam os computadores e especialmente, como conseguir que estes façam aquilo que queremos.

Assim sendo, a plataforma ECoS (<https://ecos.eduproject.eu/>) constitui-se como uma ferramenta de aprendizagem interdisciplinar para o ensino de lógica computacional, programação e cultura local.

À medida que jogam, os alunos irão enfrentar um ambiente de aprendizagem desafiante que deverá ser apoiado pelo professor. A narrativa é divertida, simples e envolvente, capaz de motivar os nossos alunos a jogar e aprender ao mesmo tempo.

Na plataforma ECoS, o aluno assume o papel de um agente ISA (International Security Agency - Agência Internacional de Segurança) que embarca em diferentes missões na sua região com o objetivo de evitar, ou resolver, ciberataques. A história desenrola-se no espaço de trabalho virtual do jogador, onde no separador INFO, interage com as personagens do jogo e recebe as missões associadas a um local no mapa. No separador CODE, o jogador deve introduzir o

código correto que resolve a missão para poder avançar na história. À medida que a história avança, surgem mais situações que têm de ser resolvidas, ao mesmo tempo que mais locais regionais ficam disponíveis para os jogadores aprenderem sobre a sua história e respetivo impacto cultural.



Espaço de trabalho do agente ISA.

Este jogo preenche um nicho importante no sentido de ensinar competências de programação nos alunos mais jovens. Ao torná-lo um jogo divertido, que utiliza uma aprendizagem “disfarçada”, encoraja-os a aprender, ao mesmo tempo que se envolvem numa atividade divertida. A plataforma ECoS pode ser facilmente utilizada dentro da sala de aula (computador ou tablet), pois corre num navegador de internet e não requer qualquer instalação. Os professores têm a liberdade para deixar que os seus alunos aprendam ao seu próprio ritmo e apoiá-los através do jogo, sempre que precisarem. Esta característica torna a plataforma ECoS numa ferramenta ideal para a aprendizagem autónoma e descobrir mais sobre programação fora do contexto letivo.

Existem muitos projetos nacionais e internacionais com o objetivo de melhorar a educação nas STEM ou de tornar a educação nas STEM mais atrativa. Aqueles projetos com foco em tecnologia beneficiarão desta plataforma. E na maioria dos casos, será complementar a novas abordagens de aprendizagem, a projetos que desenvolvam materiais de apoio ao ensino e ao currículo, bem como a projetos que se concentram nas diferentes abordagens ao ensino das STEM.

O que é a programação?

Com que então queres aprender a programar? Primeiro, tens que saber o que é programação. Uma das definições mais completas e ainda assim, a mais fácil de aprender, pode ser encontrada no local mais provável de todos, a Wikipédia:

“Programação é o processo de desenhar e construir um programa executável para atingir um resultado específico ou executar uma tarefa específica. A programação envolve tarefas tais como: análise, geração de algoritmos, análise da precisão e consumo de recursos de algoritmos e a implementação destes numa linguagem de programação. O código fonte de um programa é escrito numa ou mais linguagens inteligíveis para os programas, ao invés de linguagem máquina, que é executada diretamente pela unidade central de processamento. O propósito da programação é encontrar a sequência de instruções que irá automatizar uma tarefa num computador, geralmente resolvendo um problema particular. A programação proficiente, portanto, geralmente requer experiência em diferentes temas, incluindo conhecimento do domínio da aplicação, algoritmos especializados e lógica formal.”

Em resumo, a programação permite escrever uma “receita” para o computador seguir. Esta “receita” pode ser muitas coisas, por exemplo: um jogo, um processador de texto, uma tarefa que necessitamos de automatizar, etc., o computador irá então seguir essa “receita”, executando-a tal e qual como está escrita.

O que é o Python?

O Python é uma linguagem de programação de alto nível, de âmbito genérico e interpretada, com uma filosofia de design que enfatiza a legibilidade do código escrito, fazendo um uso notável da indentação. Os seus constructos de linguagem e a sua abordagem orientada a objetos, tem como objetivo ajudar os programadores a escrever código claro e lógico para projetos de pequena e de grande dimensão.

Esta filosofia de legibilidade do código está no seu cerne e pode ser resumida de uma forma muito elegante. De facto, pode vê-la diretamente no Python, bastando para tal, instalar o interpretador oficial de Python no seu computador e escrever na linha de comandos, o seguinte comando: "import this". Depois, ser-lhe-á apresentado o seguinte texto:

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Flat is better than nested.
- Sparse is better than dense.
- Readability counts.
- Special cases aren't special enough to break the rules.
- Although practicality beats purity.
- Errors should never pass silently.
- Unless explicitly silenced.
- In the face of ambiguity, refuse the temptation to guess.
- There should be one-- and preferably only one --obvious way to do it.
- Although that way may not be obvious at first unless you're Dutch.
- Now is better than never.
- Although never is often better than *right* now.
- If the implementation is hard to explain, it's a bad idea.
- If the implementation is easy to explain, it may be a good idea.
- Namespaces are one honking great idea -- let's do more of those!

O Python foi desenhado para ser altamente extensível, como tal, ao invés de ter todas as suas funcionalidades implementadas no interpretador, este utiliza módulos para aumentar as suas capacidades. Esta modularidade torna-o particularmente popular como um meio de adicionar interfaces de programação a aplicações existentes. Um objetivo importante dos responsáveis pelo desenvolvimento do Python é mantê-lo divertido de usar, algo que certamente ajudou-o a ganhar aderência na educação. Isto reflete-se no nome da linguagem, que é um tributo ao grupo de comédia britânico Monty Python (e não ao animal), esta atitude divertida também pode ser vista em vários tutoriais e materiais de referência disponibilizados, onde pode encontrar referências a spam e ovos, uma vez mais referências ao grupo de comédia britânico. Também pode vê-la no Zen do Python.

Todas as funcionalidades e capacidades permitiram ao Python ser consistentemente classificado como uma das linguagens de programação mais populares.

Porquê o Python?

Então porque deveríamos escolher o Python para o nosso projeto??

Bem, antes de mais deveremos pensar numa linguagem de programação como uma ferramenta que nos permite criar alguma coisa, como tal deve escolher a ferramenta correta para o trabalho que tem entre mãos. Isto também significa que o Python pode não ser a resposta, por exemplo se estiver a desenvolver um jogo AAA, que faz uso intensivo de 3D, shaders, etc. Neste caso não será a escolha mais comum, no entanto não seria inédito, por exemplo o jogo online massivo que é o EVE Online utiliza o Python.

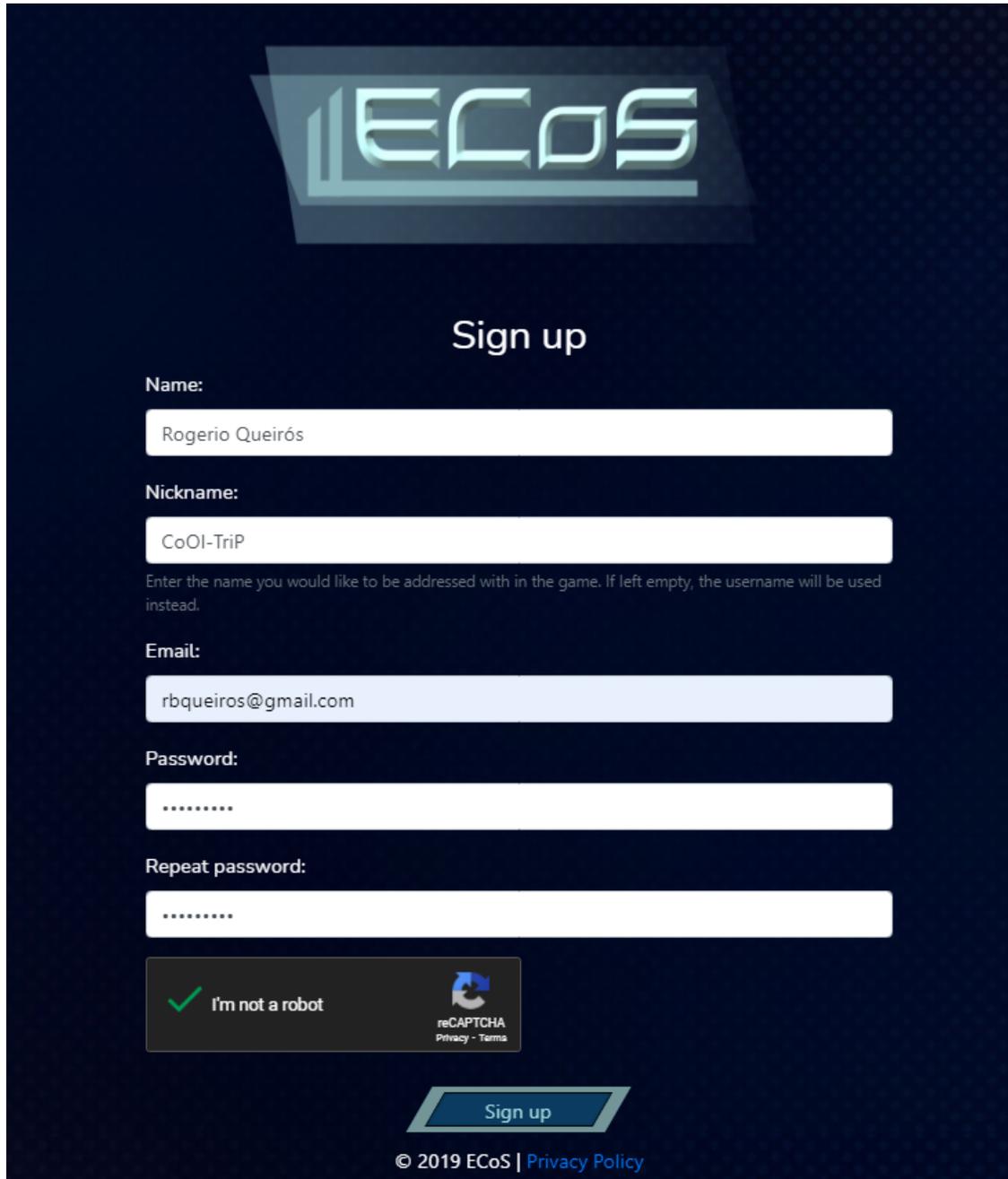
Então vamos passar por algumas das razões que fazem do Python uma das linguagens mais populares no dia de hoje:

1. É fácil de ler, escrever, aprender e utilizar;
2. Versátil, o que significa que pode utilizá-la para completar uma grande variedade de projetos;
3. Altamente flexível e extensível, de jogos AAA a páginas web, pode criar quase tudo com ele;
4. Uma das linguagens em mais rápido crescimento;
5. Existe uma grande necessidade de programadores fluentes em Python;
6. Uma comunidade muito ativa e solidária;
7. Tem uma biblioteca para todas as necessidades, da matemática à química, passando pela astronomia o Python tem-nas todas;
8. Suporta múltiplos paradigmas de programação, desde programação orientada a objetos até á programação imperativa, todos podem ser utilizados, inclusivamente no mesmo projetos, tornando-o numa excelente ferramenta de ensino;
9. É a linguagem de escolha para Inteligência Artificial e Ciência de Dados (extração de dados, manipulação, etc.);
10. Suporta múltiplas plataformas, desde o Windows, Android, Linux, até mesmo sistemas embebidos. Pode utilizar microcontroladores como o ESP32 e programá-los diretamente utilizando o Python, sem a necessidade de flashar novo firmware sempre que muda o seu código;
11. Código aberto;
12. Etc.

Como pode ver, existem muitas razões para escolher o Python e algumas delas podem agradar-lhe. Numa nota pessoal, a legibilidade do código é aquela que admiro, ser capaz de ler código que pode ter sido escrito há alguns meses atrás ou que um aluno submeteu para avaliação com pouca ou nenhuma dificuldade é um grande feito desta linguagem.

Tutorial

Registo na 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 reCAPTCHA widget. The "Name" field contains "Rogerio Queirós". The "Nickname" field contains "CoOI-TriP", with a small note below it stating: "Enter the name you would like to be addressed with in the game. If left empty, the username will be used instead." The "Email" field contains "rbqueiros@gmail.com". The "Password" and "Repeat password" fields are both filled with eight dots. At the bottom left, there is a reCAPTCHA widget with a green checkmark and the text "I'm not a robot". To the right of the reCAPTCHA is a small logo for reCAPTCHA with the text "reCAPTCHA Privacy - Terms". At the bottom center, there is a "Sign up" button. At the very bottom, the copyright notice "© 2019 ECOS | Privacy Policy" is visible.

ECOS

Sign up

Name:
Rogerio Queirós

Nickname:
CoOI-TriP
Enter the name you would like to be addressed with in the game. If left empty, the username will be used instead.

Email:
rbqueiros@gmail.com

Password:
.....

Repeat password:
.....

I'm not a robot

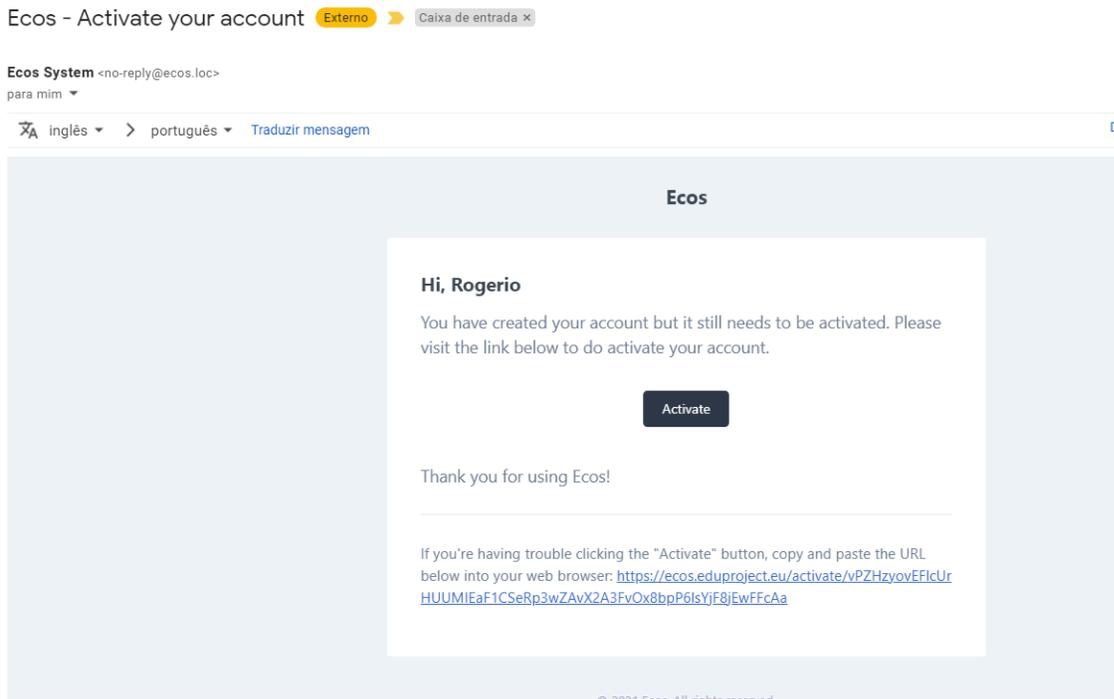
reCAPTCHA
Privacy - Terms

Sign up

© 2019 ECOS | [Privacy Policy](#)

Cada jogador que se registe na plataforma deverá ter um endereço de email válido, escolher uma alcunha e uma senha com pelo menos 8 caracteres.

Validar conta



Após o processo de registo, os jogadores deverão ativar as suas contas, para tal deverão entrar nas suas contas de email e clicar em "Ativar" no email enviado pela plataforma ECoS.

Entrada

ECoS

Log in

Email:

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](#)

Os utilizadores deverão introduzir o seu email e senha e clicar no botão de entrada. Em alternativa, se tiverem um “Session Token” também poderão autenticar-se dessa forma.



Scenario Code

eyJpZCI6NCwibmFtZSI6IkN5YmVyaXR5IFNjZW5hcmlvIn0=

Start Scenario

There are no saved scenarios available

Continue Scenario

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

eyJpZCI6NCwibmFtZSI6IkN5YmVyaXR5IFNjZW5hcmlvIn0=

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

Após a autenticação os utilizadores deverão copiar o código do cenário de Cibersegurança para e clicar em iniciar cenário.

Introduction

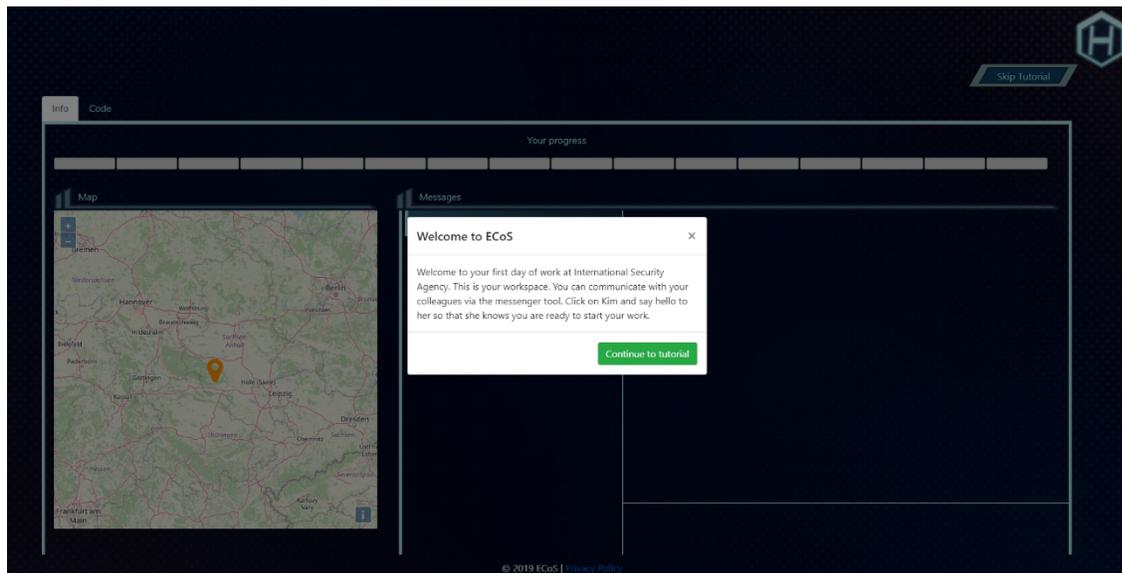
The following tutorial is going to introduce you to the basics of using ECoS as well as Python. If you feel like you do not need a tutorial, feel free to skip it using the button below.

Start

Skip Tutorial

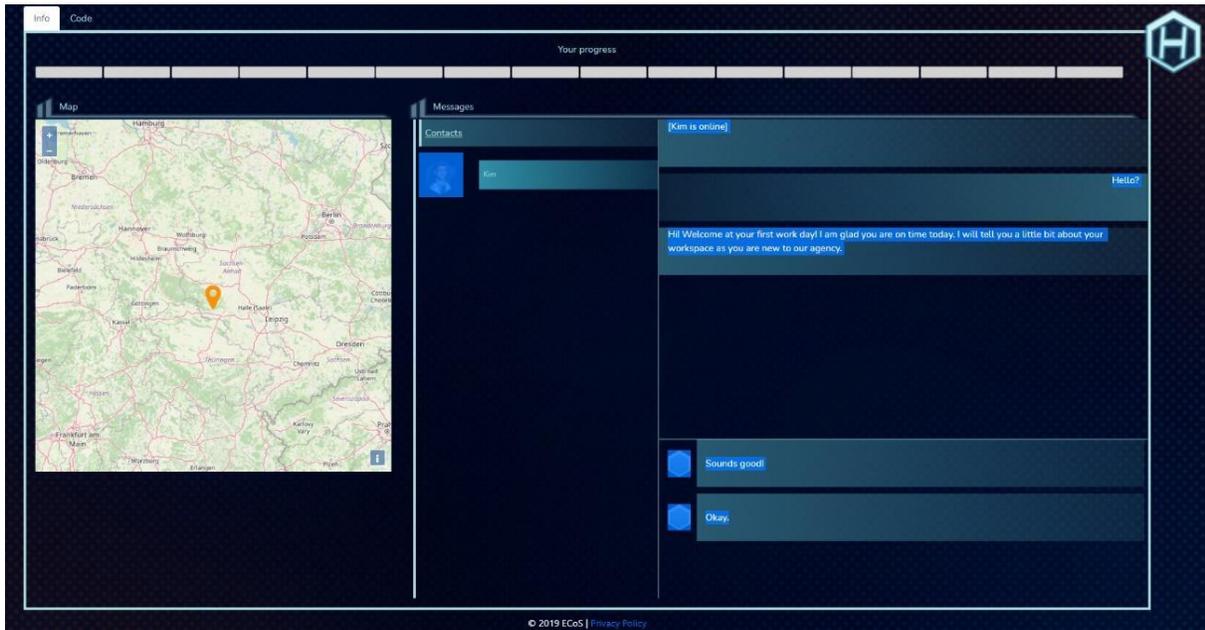
Após autenticar-se pela primeira vez, os utilizadores terão a possibilidade de executar o tutorial ou simplesmente começar a jogar sem ajuda.

Cenário



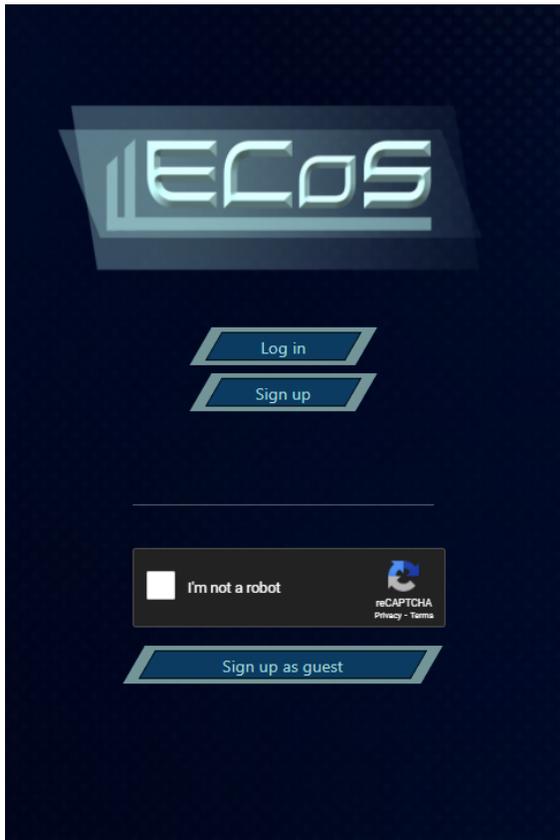
Na primeira entrada os jogadores serão cumprimentados com uma mensagem de boas vindas.

Agente KIM



A Kim é a agente que dará instruções no decorrer do jogo, desde o tutorial até a dicas para ajudar o jogador a completar as suas tarefas.

Entrada como convidado



Se desejar explorar o jogo sem se registrar, pode também entrar como convidado bastando para tal entrar como convidado.



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](#)

Copie o seu código de sessão e clique para regressar à entrada.



Scenario Code

Start Scenario

There are no saved scenarios available

Continue Scenario

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

eyJpZCI6NCwibmFtZSI6IkN5YmVyIFNlY3VyaXR5IFNjZW5hcmlvln0=

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

Cole o seu código de sessão e clique em iniciar o cenário.

Cenários ECOS

Nesta seção iremos guiá-lo pelos cenários para que saiba exatamente quais as tarefas e soluções. Esta pretende ser um guia para professores que queiram utilizar a plataforma ECoS na sua sala de aula. O conteúdo de cada cenário será exibido juntamente com a solução para cada tarefa de programação. Cada seção terá também comentários.

O Cenário

No ECoS o jogador irá trabalhar para a ISA, a agência de segurança que monitoriza a segurança da rede da cidade. A ISA é, claro está, uma agência ficcional que nada tem a ver com a forma como a segurança de redes funciona. Os cenários devem ser vistos como uma forma leve e divertida de abordar este tópico e o seu propósito é fazer o aluno sentir-se importante apesar de ser um principiante na programação.

Personagens

As personagens que os jogadores podem interagir com, são:

Michael – O Michael é o patrão do jogador. Ele é muito ocupado e não irá envolver-se muito frequentemente.

Piedro – Qualquer boa agência precisa de agentes de campo que façam o trabalho sujo. Piedro é o agente secreto que sai e que faz o trabalho.

Livia – Como uma agente de segurança com experiência, a Livia irá manter-se atenta aos eventos atuais e dar tarefas ao jogador. Ela é uma colega prestativa.

Cenários

O conteúdo de texto nos cenários será sempre exibido como mensagens no sistema de chat do jogo. Qualquer conteúdo dentro de parênteses retos ([]) é texto dinâmico que será substituído pelo conteúdo apropriado. Este poderá ser o nome do jogador ou o nome de uma localização. Qualquer conteúdo dentro de chavetas ({ }) indica uma escolha que o jogador tem.

Introdução

Visão geral:

Esta é apenas uma introdução para o jogador que lança o 1º cenário imediatamente.

Texto do Chat:

Livia: Olá [nome]. Ouvi dizer que hoje é o teu primeiro dia cá. A Kim disse-me que ela já te introduziu ao nosso espaço de trabalho. Ainda bem, agora já posso passar-te algumas das minhas tarefas..

{Ena, parece ser bem direto}
{Claro, o que quer que eu faça?}

Livia: Como sabes, somos os responsáveis pela cibersegurança das localizações indicadas. Dá uma vista de olhos ao mapa. Estes são os locais que mantemos. Geralmente, não existe muito para fazer, por isso tira um pouco de café e...

Livia: Oh, espera... acabamos de receber um alarme!

Objetivo:

A razão para manter a introdução tão curta é para envolver o jogador imediatamente. A introdução dá uma ideia rápida de onde eles estão e o que é esperado.

Cenário 1: Edifício Público (Câmara Municipal)

Visão geral:

Este cenário introduz uma tarefa simples, na qual o aluno terá que comparar um valor num array e alterá-lo para 0 se for 1.

Texto do chat:

Livia: Oh não! Todos os alarmes estão a tocar no [City Hall] Acho que alguém entrou no sistema de controlo e agora eles não conseguem desligá-los! Liga-te ao sistema e desliga os alarmes enquanto eu descubro o hacker..

{Como?}

{Lembre-se, este é o meu primeiro dia!}

Livia: Eu já te liguei ao sistema deles. Verifica todos os alarmes. Se estão marcados com 1 estão ligados, tens que os alterar para 0 para os desligar..

Objetivo:

Este cenário introduz uma tarefa fácil para introduzir o aluno rapidamente ao cenário.

Tarefa

Verifica todos os alarmes. Aqueles que estiverem marcados com 1 estão ligados, tens que os alterar para 0 para os desligar.

alarms = [1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0]

Solução

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

Cenário 2: Instituição Educacional

Visão geral:

This task is a little more tricky because it relies on a loop which runs through all elements in an array and converts them if they are not empty spaces.

Texto do Chat:

Livia: Bom trabalho! Ainda estou à procura do hacker. Já descobri que ele estabeleceu ligação a partir da Ásia. Enquanto estou ocupado, podes tratar do [Universidade local]. Um amigo meu, Marcus, é o administrador de sistemas lá, vamos enviar-lhe uma mensagem encriptada..

{Porquê encriptado?}

{Para quê?}

Livia: O [Universidade local] foi atacado anteriormente e eu quero garantir que eles estão alerta. No entanto, se eles já foram atacados não quero que os hackers consigam ler a mensagem. Por isso, encripta a mensagem, que o Marcus conseguirá decifrá-la..

{Como faço para a encriptar?}

{Qual é a mensagem?}

Livia: “Salta” as letras do alfabeto em 2 posições. Um A transforma-se num C, um B transforma-se num D, um C num E, e assim sucessivamente. Um Y transforma-se num A, o Z num B. Envia a mensagem “Alerta hackers! Verifiquem os vossos sistemas!”.

Objetivo:

Esta tarefa é suficientemente complicada para fazer o aluno pensar um pouco. Idealmente o aluno divide a tarefa em partes mais pequenas e resolve-as sucessivamente.

Tarefa

Encripta a mensagem que está atribuída na variável "toEncrypt" e imprime-a usando a Cifra de César. Um A torna-se um C, um B torna-se um D, um C torna-se um E, e assim sucessivamente. Quando se atinge o final do abecedário, continuamos desde o seu início: Um

Y torna-se um A, um Z torna-se um B. Quando um caractere for um espaço (" "), não é necessário fazer qualquer ação.

```
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"]
```

```
toEncrypt = "hacker alert check your systems"
```

```
encryptedMessage = ""
```

Solução

```
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: Bom trabalho! Acho que por hoje o nosso trabalho está terminado, vemo-nos novamente amanhã!

Cenário 3: Biblioteca

Visão geral:

Este cenário é um pouco mais complexo e consiste em várias tarefas que estão ligadas entre si. O grande desafio é incluir um módulo diferente através do comando import

Texto do chat:

Contacto desconhecido:Olá? Isto é a ISA?

{Sim, quem és?}

{Como podemos ajudá-lo?}

Contacto desconhecido: Este é o Esquadrão de Infiltração Lazer. Pague 1.000.000 € ou vai ficar em apuros. Desta vez não conseguirá defender-se dos nossos ataques.

{Sim, sim, certo}

{Deixa-me perguntar aos meus colegas}

Contacto desconhecido: Isto não é uma piada..

Livia: Olá! Está tudo bem? Como estás a progredir com o teu trabalho?

{Acabei de receber uma mensagem de um contacto desconhecido}

Livia: O quê? No nosso sistema de mensagens interno? Isso não deveria ser possível. Dá-me um momento....

Livia: Sim, consigo vê-lo. Existe um novo contacto na nossa base de dados! Como é que ele conseguiu criar o contacto?

{Como devo responder?}

Livia: Ok. O primeiro passo em situações sérias como esta: Descobrir o seu endereço IP. Sabes o que é um endereço IP?

{Na verdade, não}

{O que é isso?}

Livia: Todos quanto navegam na Internet têm um endereço IP. Os endereços IP podem ter diversas estruturas. Um endereço IPv4 tradicional consiste em 4 números, com um máximo de 3 dígitos, todos separados por um ponto.

Livia: Por exemplo: 135.235.86.23. Vamos garantir que consegues fazer isto. Por favor, define uma variável que contenha um endereço IP válido e imprime-o.

Tarefa

Define uma variável, do tipo string, que contenha um endereço IPv4 e imprime-a. Tem em conta que o endereço IPv4 consiste em 4 números, com um máximo de 3 dígitos, todos separados por um ponto.

Solução

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

Texto do chat:

Quando tens as ferramentas certas e o endereço IP de alguém, podes segui-los e descobrir de onde é que se estão a ligar. Por exemplo, a polícia e nós somos capazes de descobrir a posição mais ou menos exata do utilizador.

{Como consigo fazer isso?}

Livia: Nós temos todo o tipo de diferentes ferramentas secretas. Uma delas é o dispositivo de seguimento IP (IP-Tracking-Device). Nós temos o nome de utilizador, o hacker está a usar o nome “nightwolf”. É um nome de utilizador estranho. Ele deve ser uma verdadeira aberração.

Livia: Utiliza a função “read_ip(string ip)” do dispositivo de seguimento IP e introduz o nome de utilizador exacto como o seu parâmetro. O dispositivo de seguimento irá verificar os nossos registos e devolver o IP dele.

Tarefa

O nome de utilizador do hacker é Nightwolf. Define uma variável que contenha o seu nome de utilizador. Chama a função read_ip() do dispositivo de seguimento IP (IP-Tracking-Device). Verifica as opções à tua esquerda. Aí poderás facilmente seleccionar a função read_ip() e o única coisa que necessitarás de fazer, é introduzir o nome da tua variável como um parâmetro. Finalmente, imprime o IP que o dispositivo de seguimento te fornecerá.

Solução

```
variable_name = “nightwolf”  
  
print(ip_tracking_device.read_ip(variable_name));
```

(Importante: Definir uma variável e imprimir o resultado do dispositivo de seguimento, que deverá estar correto. O resultado que deverá retornar para “nightwolf” é “183.42.58.12”

Texto do chat:

Livia: Conseguiste obter o IP?

{Sim. O IP é: 183.42.58.12}

Livia: Ok. Bom! Parece que ele ainda está ligado. Ele parece estar no [Biblioteca]. Vamos tentar atacá-lo rapidamente antes que ele saia da rede deles.

{Ok. Como vais fazer isso?}

Livia: TU vai fazê-lo. Utiliza o Módulo de verificação de portas. Passa por todas as diferentes portas para ter a certeza que consegues encontrar uma que podemos usar para entrar no computador dele. Eu vou utilizá-a e colocar um cavalo de tróia lá.

Tarefa

Inicia o módulo de verificação de portas (Port-Checker-Module). Chama a sua função `set_ip(ip)` e passa o IP que acabaste de obter. Lembra-te: o IP deve ser fornecido como uma string. E já agora, o IP é: 183.42.58.12

Utiliza um ciclo `for` para percorrer os números de 1 a 99999. Passa cada um dos números à função `is_port_vulnerable(porta)` do módulo de verificação de portas. Imprime o número se for uma porta vulnerável.

Hidden Code:

```
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
```

Solução

```
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 do chat:

Livia: Encontre algumas portas vulneráveis que possamos utilizar?

{Sim. 553, 145 e 86744}

Livia: Perfeito. Coloquei um pouco de código no computador dele. Faz-me o seguinte favor: Deves conseguir utilizar a funcionalidade do cavalo de tróia agora. Tem uma função chamada `list_files()`. Por favor, lista todos os ficheiros do ambiente de trabalho e imprime os respetivos nomes.

Tarefa

Agora que o cavalo de tróia está instalado no computador do hacker, é possível usar as suas funções para aceder ao computador. Usa o comando `get_recent_files()` do cavalo de tróia. Este comando devolverá uma lista e essa mesma lista será atribuída a uma variável. Percorre a lista através de um loop, e imprime todos os nomes que constam nessa lista.

Hidden Code

```
#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)
```

Solução

```
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:

A razão pela qual esta seção está dividida em várias tarefas é porque o aluno deverá ser introduzido ao conceito de chamar uma classe diferente de uma forma gradual. A ideia é que o aluno repete esta tarefa para que se torne mais confiante em usá-la.

Cenário 4: Centro Desportivo

Visão geral:

Esta tarefa é novamente simples. Tudo o que o aluno tem a fazer é filtrar todo o código indesejado da string.

Texto do Chat:

Livia: Funcionou?

{Sim, encontrei alguns ficheiros estranhos. Existe um ficheiro chamado secretTarget.txt}

Livia: Oh. Isso não pode ser bom. Deixa-me verificar o ficheiro.... oh....

{O quê?}

{Más notícias?}

Livia: O seu próximo alvo é....o [Centro Desportivo].

Livia: Deveríamos configurar-lhes uma firewall para garantir que ninguém consiga atacá-los.

Tarefa

Parece que o hacker pretende atacar o [Centro Desportivo]! Está na hora de configurar firewall de proteção. Escreve um algoritmo que remova tudo, menos os valores 0 e 1 da string de entrada. Sempre que testares o teu algoritmo, a string terá um valor diferente. Por isso, certifica-te que o programa consegue correr bem. Define uma segunda variável que contenha o valor da primeira variável com apenas os valores 0 e 1. Imprime essa string.

Solução

```
request =  
"100101010001010worm?hack0101010101101malware0010110100110101virus101011010010  
100101010001111010101101010011"  
  
result = ""  
  
for c in request:  
    if c == "0" or c == "1":  
        result += c  
  
print(result)
```

Objetivo:

Esta é uma tarefa que visa encorajar o aluno uma vez que pode ser resolvida facilmente e rapidamente. Está localizada imediatamente após o cenário 3, que é mais complexo, exatamente com o propósito de encorajamento.

Cenário 5: Igreja

Visão geral:

Este cenário revisita a tarefa do cenário 1 e aumenta ligeiramente a dificuldade.

Texto do chat:

Livia: Obrigado. Espero que eles tenham segurança suficiente para aguentar os ataques.

Livia: Acabei de receber uma chamada da igreja. Os sinos começaram a tocar e o sistema eletrónico que os controla está bloqueado.

{Talvez ele só queira irritar as pessoas?}

{Talvez ele esteja a testar-nos?}

Livia: Talvez. Mas precisamos de agir e pôr um fim nisto. Liga-te ao sistema de controlo e desliga os sinos.

Tarefa

Verifica a lista `connectedDevices`. Se o dispositivo for sinos de igreja, desliga-os na lista `deviceStates`. O valor 0 na array `deviceStates`, significa desligado. E o valor 1 na array `deviceStates`, significa ligados.

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

Solução

```
For i in connectedDevices:
    If connectedDevices[i] == "churchbell":
        If deviceStates[i] == 1:
            deviceStates[i] = 0
```

Objetivo:

A ideia aqui é desafiar o aluno agarrando em algo que deverá ser já familiar e aumentar ligeiramente a dificuldade.

Cenário 6: Economia / Banco

Visão geral:

Esta tarefa exige efetuar uma comparação entre dois arrays e calcular a diferença entre os valores.

Texto do chat:

Michael (Patrão): [Nome]! O que se está a passar! O banco foi atacado! Porque é que não fizeste nada para impedir isso?

{Desculpe, estava ocupado a desligar os sinos da igreja...}

Michael (patrão): Porque te focaste na igreja? Ninguém se importa se os sinos estão a tocar, a não ser que seja um domingo de manhã às 10h e ainda estejas na cama!

Livia: [Nome], parece que o hacker montou uma armadilha para nos distrair. Aparentemente o [Centro Desportivo] nunca foi um alvo... e o momento em que os sinos da igreja começaram a tocar não foi uma coincidência.

{Raios...}

{O que devemos fazer agora?}

Livia: Acho que o ataque foi bem sucedido. O hacker conseguiu roubar bastante dinheiro. Acho que ele é bastante avançado. Por favor, ajuda o banco a descobrir quanto dinheiro foi roubado. Devemos pelo menos descobrir isso e depois tentar apanhá-lo!

Tarefa

Calcula o somatório do dinheiro em falta nos diferentes grupos de contas do banco de Portugal. Compara a lista afterHack com a lista beforeHack e os seus números nas posições individuais. Se os números na mesma posição de ambas as listas forem diferentes, adiciona esta diferença ao somatório do dinheiro em falta. Imprime o somatório do dinheiro em falta.

Solução

```
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 tarefa introduz cálculos com variáveis que são um pouco mais complexos mas ainda acessíveis. O aluno não deverá ser surpreendido com esta tarefa.

Cenário 7: Transporte

Visão geral:

Uma vez a tarefa envolve percorrer um array e efetuar uma comparação.

Texto do chat:

Livia: É bastante dinheiro. Temos que apanhar esta pessoa, ou pelo menos obter mais informação.

Livia: Analisei o computador do hacker um pouco mais e descobri o nome dele. Também descobri que, aparentemente, entre o ataque ao [Edifício público] e o ataque ao [Banco] ele viajou para esta cidade.

Livia: O nome dele é Mario

{Oh, isso é bom!}

{Fixe. Como o Super Mario}

Livia: Simmm.... No entanto, a minha ideia. Uma vez que ele chegou cá recentemente, poderíamos ir ao [Transporte] e verificar as últimas listas de passageiros em busca do nome dele. Um dos Marios será provavelmente ele. Isso vai dar-nos mais informações.

{Ena! És tão esperto.}

{Vamos a isto!}

Tarefa

Percorre, através de um loop, a lista de passageiros. Verifica se o nome é "Mario". Se for afirmativo, verifica a mesma posição na lista de passaportes e imprime esses dados.

```
passengersList = ["Mario", "Luigi", "Carlo", "Isabella", "Liseth", "Maja", "Gabriel", "Santos",  
"Cristina"]  
passportList = ["A92fKsk2", "Kfask201", "kf0kak2D", "kbcA12d9", "Ake012F34", "Yke92A1f",  
"Rks18dA1"]
```

Solução

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

Objetivo:

O aluno solidificar o seu conhecimentos de loops e a sua utilidade para a pesquisa de informação.

Cenário 8: Parques / natureza

Visão geral:

Neste cenário o aluno terá que trabalhar com posições num array para conseguir descriptar uma mensagem encriptada.

Texto do chat:

Livia: Boa. Aqui temos o nosso Mario ...

Livia: Acho que o nosso alvo acabou de se ligar à rede Wi-Fi pública na Levada do Norte.

{Eu nem sabia que tínhamos Wi-Fi pública no parque!}}

{Ainda bem que ele não usou uma VPN.}

Livia: Sim, geralmente não sugiro usar uma rede Wi-Fi pública, pois não é muito segura, como podes ver. Podes vigiar o tráfico e escutar as mensagens dele?

{Vou ver o que posso fazer}

Tarefa

O Nightwolf usa a rede Wi-Fi pública na Levada do Norte para enviar mensagens encriptadas a alguém. Parece que usou a Cifra de César que já usámos anteriormente para enviar os nossos alertas. Desta vez, contudo, é o inverso - a mensagem não pode ser lida até se trocarem todas as letras na mensagem, por um determinado valor. Sugiro-te que tentes diferentes números e vejas quais funcionam para desencriptar a mensagem. Por favor, imprime a mensagem depois de desencriptá-la.

```
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."
```

Solução

```
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)
```

Chat Text:

Livia: Conseguiste desencriptar a mensagem?

{“Olá amigos! Eu vou atacar a LOCALIZAÇÃO dentro de algumas horas! Fiquem atentos para verem ação a sério. Ah, e já agora, comprei um bilhete para a peça Macbeth no TEATRO para esta noite. Espero que seja uma boa produção.”}

Objetivo:

Esta tarefa é um puzzle bastante complicado para os alunos. Baseia-se em algo familiar para que o aluno tenha algum tipo de orientação. No entanto, o processo de descriptação é um grande desafio e o aluno poderá necessitar de ajuda. O objetivo deste exercício é levar os alunos ao limite e desafiá-los

Cenário 9: Teatro

Visão geral:

Este é um problema lógico, bastante simples, que requer raciocínio lógico.

Texto do chat:

Livia: Oh não, isto são muito más notícias! Mas pelo menos conseguimos descobrir os planos dele para esta noite. Ah, tive uma boa ideia. Vamos enviar um agente para que se sente no assento ao lado dele e colocar um dispositivo de seguimento. Apenas temos que marcar um assento para o nosso agente ao lado dele.

{Como fazemos isso?}

{Estou nisso!}

Livia: Liga-te ao sistema de reservas do [Teatro]. O Nightwolf está no assento 16. Muda a reserva do assento 15 para “Lisa Fernandez”. Essa é a nossa agente.

Tarefa

No [Teatro], altera a reserva do assento 15 com o nome da nossa agente. Eu já fiz o teu registo no sistema de reservas do teatro. Se houver alguém com reserva para o lugar 15, por favor, altera a reserva dessa pessoa para outro lugar, de forma gratuita - somos hackers, mas não somos más pessoas.

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”]

Solução

```
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:

O objetivo deste exercício é desafiar as competências de raciocínio lógico.

Cenário 10: Mercado

Visão geral:

Este exercício introduz funções e foca-se na comparação de dados.

Texto do chat:

Livia: [Nome!] O nosso dispositivo de seguimento mostra-nos que o Nightwolf está neste momento na Livraria Esperança. O Piedro foi até lá para observá-lo e consegue ver que ele está a falar com alguém. Infelizmente, devido à distância, ele não consegue ver quem é.

Livia: Felizmente temos câmaras de segurança de alta tecnologia que podem ser utilizadas para identificar essa pessoa. Por favor, entra na câmara de segurança X23M1 e utiliza as funcionalidades de identificação para obter mais dados.

{Como faço isso?}

{Acha que já estou assim tão confortável com todas as suas ferramentas de alta tecnologia?}

Livia: Vou mostrar-te como fazê-lo.

Tarefa

Acede à câmara de segurança "X23M1", através do comando `security_cameras.connect_to("camera id")`. Usa o comando `security_cameras.get_target_feature("feature_type")` para obter atributos específicos da pessoa que se encontra a falar com o nosso hacker. Os possíveis atributos são "hair_color" (cor de cabelo), "eye_color" (cor de olhos), "hair_type" (tipo de cabelo), "skin_color" (cor de pele) e "features" (atributos). Compara estes atributos com a nossa base de dados criminal. Se uma das posições da base de dados criminal fizer correspondência com todos os atributos identificados, imprime o seu nome.

```
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", "scar", "none"]
```

Objetivo:

O exercício introduz o uso de funções ao aluno

Cenário 11: Central energética / Fornecimento de água

Visão geral:

Nesta seção o aluno irá criar uma classe e chamá-la no código principal.

Texto do chat:

Livia: Deveríamos criar uma ferramenta de alarme para a [Central energética], pois não sabemos a hora exata em que ele vai atacar. Uma vez que temos o nosso dispositivo de seguimento no computador de Nightwolf, deve ser bastante fácil para nós fazermos isso.

{Ok, eu vou tratar disto}

Livia: Parece-me bem. Verifica as funções que estão disponíveis para o dispositivo de seguimento. Elas devem ser suficientes para ajudar a criar uma ferramenta. Certifica-te que o alarme é ativado quando ele estiver a menos de 500 metros da [Central energética] e nós enviaremos uma unidade especial da polícia.

Tarefa

Cria uma ferramenta de alarme no dispositivo de seguimento. A ferramenta de alarme tem que ser uma classe com a função "check_status" que nos assegure que Nightwolf não se encontra a uma distância de 500 metros da [Localização] Se ele estiver a menos de 500 metros, a ferramenta devolverá o valor 1. Senão, devolverá o valor 0. Eu já preparei uma classe vazia para ti. Introduce o nome da localização, [Localização], como parâmetro da função check_distance_to() do dispositivo de seguimento. Imprime-a quando "check_status" for chamado. Chama a função do exterior da classe para verificar se funciona.

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

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

Objetivo:

Este exercício deixa o aluno criar uma classe e introduz um conceito avançado de programação.

Cenário 12: Festival / Eventos

Visão geral:

Neste exercício o aluno irá utilizar um método de ataque por força bruta e escrever um loop simples.

Texto do chat:

Livia: O Nightwolf atacou novamente! Ele atacou os altifalantes no festival e está a exigir um milhão de euros ou irá destruir permanentemente a [Monumento]! Todos conseguem ouvir a sua voz e estão bastante assustados. Temos que recuperar o controlo dos altifalantes para que ele não consiga causar mais pânico.

{Eu vou desligar isso imediatamente.}

{Raios! Eu queria dormir uma soneca. Ora bem.}

Tarefa

Escreve um algoritmo que encontre o código correto para aceder novamente aos altifalantes. O código é um número de 4 dígitos. Imprime todas as combinações possíveis.

Hidden Code

```
# 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)
```

Solução

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

```
print(i);
```

Objetivo:

Este exercício deverá ser fácil para o aluno. A tarefa é a introdução ao ataque por força bruta.

Cenário 13: Monumento

Visão geral:

Este exercício exige a utilização da estrutura if na sua forma estendida eli.

Texto do chat:

Livia: Ok, enquanto estavas a desligar os altifalantes, eu notifiquei a polícia que, entretanto, já chegou até junto da [Monumento].

Livia: O problema é que eles parecem ser incapazes de aceder aos controlos da bomba. Contudo, eu liguei-me com sucesso à bomba, porque ela está ligada à Internet. Infelizmente não consigo perceber o seu mecanismo. Talvez tu tenhas uma ideia do seu funcionamento?

{Vou dar tudo o que tenho, chefe.}

{Vai ser um espetáculo desarmar.}

Tarefa

Alguém preparou uma bomba para nós! Tens que desativá-la. Certifica-te que imprimes as cores corretas, na ordem também correta. A bomba tem uma sequência de cores que precisa de ser respondida com a resposta certa. Existem 4 variáveis que definem a resposta para cada cor. Imprime "vermelho" se a sequência é amarelo. Imprime "amarelo" se a sequência é azul. Imprime "verde" se a sequência é vermelho. Imprime "azul" se a sequência é verde.

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

Solução

```
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:

De um ponto de vista de programação este exercício deverá ser bastante simples para o aluno. É na sua maioria um exercício de raciocínio lógico.

Cenário 14: Museu

Visão geral:

Nesta última tarefa o aluno necessitará de ligar com uma versão expandida do método de ataque por força bruta.

Texto do chat:

Livia: Inacreditável! A polícia não conseguiu prender o Nightwolf antes de ele descobrir o dispositivo de seguimento. Ele fugiu para o Museu da Baleia e barricou-se lá. Todas as portas são controladas eletronicamente. Terás que encontrar o código das fechaduras para que a polícia consiga entrar e trazê-lo sob custódia. Eu sei que consegues fazer isto! As fechaduras usam um código de 4 dígitos. Rápido!

{Este gajo está obrigando-me a trabalhar a sério.}

{Podemos ir comer um gelado a seguir?}

Tarefa

Escreve um algoritmo que encontre o código das fechaduras do museu. O hacker parece ter melhorado o seu método de palavras-passe. Ele agora o seguinte formato: Número(1-9) Número(1-9) Letra (a-z). Escreve um algoritmo que verifique todas as configurações possíveis da palavra-passe para encontrar a correta e assim, abrir as fechaduras.

Hidden code:

```
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)
```

Solução

```
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 do chat:

Livia: Ufa, essa foi por pouco! Obrigado por todo o teu trabalho, a polícia entrou no Museu da Baleia e prendeu o nightwolf e os seus amigos hackers. Não vai haver Internet para o sítio onde eles vão! Portaste-te bastante bem. Ainda bem que estás do nosso lado.

```
{Tudo num dia de trabalho.}  
{Então, isso é um 'sim' ao gelado?}
```

Objetivo:

Esta tarefa conclui o cenário. O exercício baseia-se no método de ataque por força bruta utilizado anteriormente e expande-o para um cenário mais complicado. O objetivo é novamente desafiar o aluno com um puzzle de raciocínio lógico e ao mesmo providenciar algo familiar.

O FIM

	Tarefa	Contexto
Edifícios Públicos (câmara municipal, etc.)	Um sistema de alarme foi acionado - entra no sistema e muda todos os 1 para 0	Um atacante está a testar os mecanismos de defesa da cidade
Instituição educacional	Enviar aviso encriptado (cifra de César)	Temos receio que a instituição também foi atacada e queremos avisar o administrador do sistema
Biblioteca	Ataca o computador do atacante pela primeira vez. Pesquisa o catálogo de livros e descobre que livro o atacante obteve	O atacante contata a agência e quer dinheiro! No entanto, nos registos internos o seu IP pode ser seguido. Nós atacamos o atacante utilizando o nosso IP-Checker.
Centro desportivo	Configurar uma firewall	Ao pesquisar o computador do atacante a agência descobre os ficheiros privados do atacante. Um deles é "next_target.txt". Nesse documento o centro desportivo é listado como um alvo. A agência decide que deve configurar uma firewall para o centro desportivo.
Igrejas	Desligar os sinos	O atacante utiliza um ataque à igreja como uma distração ! Enquanto os agentes estão ocupados a lidar com os sinos, o atacante ataca um banco.
Economia / Banco	Verifica quanto dinheiro o atacante roubou.	O atacante rouba um banco em vez dos dois primeiros alvos.

Transporte	Verifica a lista de passageiros em busca do nome do atacante	Uma vez que descobrimos que o nome do atacante é Mario, verificamos as últimas listas de passageiros em busca deste nome.
Parques / natureza	O atacante está a usar Wifi pública e envia mensagens encriptadas para a sua organização. Descripta as mensagens	Nós descobrimos que o atacante planeia sabotar infraestrutura crítica. We find out that hacker plans to sabotage critical infrastructure. Subproduto: reparamos que o atacante comprou bilhetes para o teatro para essa noite.
Teatro	Descobrir o número do lugar	O agente de campo coloca um dispositivo de seguimento no atacante.
Mercado	Câmaras de segurança - obter diferentes características da pessoa e pesquisar na base de dados criminal	O atacante encontra-se com cúmplices, nós identificamo-los
Central energética / Fornecimento de água	Alerta de proximidade - criar uma ferramenta de alarme	Somos alertados e precisamos de proteger infraestrutura crítica. O ataque chega mas o nosso sistema de alarme funciona e a policia é enviada para o local
Festividades / Eventos	Recuperar acesso a altifalantes comprometidos no evento	O atacante compromete altifalantes numa festa e usa-os para falar com o público, exigindo dinheiro e ameaçando que o monumento local irá explodir se ele não obtiver o que procura.
Monumento	Devolve sinais baseados em instruções para desarmar a bomba que alguém preparou para nós	Um grupo de atacantes ameaça explodir um monumento. Hacking group resorts to bombing the monument. Nós reparamos e desarmamos a bomba.
Museu	O atacante fecha todas as portas - descobrir código de 4 dígitos	O atacante está louco e descobre o dispositivo de seguimento. Antes que a polícia o consiga prender ele foge para um museu e sela todas as portas. Precisamos de abri-las novamente para ajudar a polícia a apanhá-lo.

Como utilizar o ECoS na aprendizagem formal, não formal e informal?

A plataforma ECoS pode ser utilizada não só numa sala de aula formal ou contexto escolar, mas também na educação não formal e informal.

Qual é a diferença entre a educação formal, não-formal e informal?

Educação formal é a educação que é regulada e reconhecida, tem testes para garantir a competência, conhecimento e possui uma estrutura vertical. É a educação obrigatória, a educação vocacional e universitária, por exemplo.

A educação não formal é estruturada, intencional mas também tem objetivos de aprendizagem e avaliação, mas neste caso é voluntária e cada participante decide até que ponto ele(a) está envolvido, inclusivamente a nível da sua própria avaliação.

A educação informal é a educação que ocorre involuntariamente, sem que a pessoa esteja consciente de que ele(a), esteja a aprender. Quando jogamos estamos a aprender regras e estratégias e muitas vezes não nos apercebemos deste facto.

Na educação informal, podemos, por exemplo, oferecer aos jovens num centro de juventude a possibilidade de jogar ao ECoS. Desta forma permitimos que possam aprender de forma independente. No entanto, é aconselhável estar disponível caso estes tenham questões ou necessitem de apoio. O importante é esperar para que sejam eles a pedir por ajuda ao invés de direcionar a sua aprendizagem.

Na educação não formal o jogo ECoS pode ser usado por si só ou no contexto mais alargado de aprendizagem (por exemplo, no contexto de um programa de intercâmbio no âmbito do programa Erasmus+).

-O facilitador irá ponderar sobre a forma como deseja utilizar o jogo, seja o jogo na sua totalidade ou apenas uma parte do mesmo. Se os participantes não tiverem experiência de programas é aconselhável começar com o tutorial.

-Pode propor uma aprendizagem individual ou tentar resolver as tarefas em pares ou grupos.

-Pode adicionar um elemento de competição entre grupos, como pontos ou prémios para as tarefas completadas com sucesso, mas isto não é obrigatório.

- No final do processo é útil fazer um balanço da experiência com os participantes partilhando como foi a experiência do jogo. Algumas questões pertinentes podem ser:

- Que dificuldades encontrou e como lidou com elas?

- Que partes foram mais motivantes e quais as mais frustrantes? Como lidou com essa frustração?

- Quando colaborou num grupo, que papel tinha no grupo, existia um líder, como foram as decisões tomadas?

-Por fim, deverá terminar com uma avaliação, seja da sessão do ECoS ou de todo o programa de treino e também com uma reflexão sobre o que aprenderam com esta experiência.

Como utilizar o ECoS na aprendizagem formal?

Sem qualquer dúvida, todas as organizações de educação formal, como escolas, colégios, universidades em diferentes países têm variados níveis de flexibilidade no seu currículo. Tudo dependerá da organização e do professor. Existem, no entanto, duas opções de inclusão do ECoS no seu sistema:

- Olhando para as metas de aprendizagem e focando-se nas competências que são desenvolvidas enquanto se joga o ECoS. Pode introduzir este jogo, numa aula de Informática, TIC, aulas de programação, enquanto aprende lógica e até aspectos culturais enquanto tenta resolver o mistério. Pode planear algumas lições como trabalho de caso e uma apresentação sobre a jornada, desafios e soluções a que chegaram, uma vez que neste jogo podem haver diferentes soluções para um problema. Os alunos poderão apresentar a sua forma de raciocínio.
- Convidar os alunos a melhorar a sua classificação jogando este jogo nos seus tempos livres, como uma atividade extracurricular (clubes).

Competências desenvolvidas:

- Digitais
- Computadores
- Programação
- Lógica
- Investigação
- Resolução de problemas
- Tomada de decisões
- Descoberta multicultural
- Idiomas estrangeiros

O melhor é experimentar você mesmo e ver que soluções são mais adequadas para si e os seus alunos.

FAQ

Onde posso jogar ao ECoS?

Em qualquer navegador, através da seguinte hiperligação: <https://ecos.eduproject.eu/>

E se a minha sala de aula não tiver dispositivos suficientes (PCs e tablets)?

Até mesmo um só dispositivo pode permitir-lhe utilizar a plataforma ECoS na sua sala. Pode por exemplo, projetar o ecrã e jogar em conjunto com a sua turma.

Ao aplicar a estratégia de “aprendizagem através da partilha” pode criar um grupo para a turma se tiver acesso a um número limitado de dispositivos.

Lembre-se de perguntar e visitar o nosso website (<http://earlycodinginschools.eu>) quanto estiver à procura de ideias. Também gostaríamos de ouvir as suas ideias se quiser partilhá-las connosco.

Não me sinto confortável com as minhas competências de programação mas gostaria de utilizar a plataforma ECoS na minha sala de aula.

O tutorial da plataforma ECoS é uma forma de praticar as suas competências de programação começando do zero. Este introduz-lhe o básico da utilização da plataforma ECoS bem como o Python. Siga as instruções da Kim e será bem sucedido.

Posso entrar logo no jogo sem fazer o tutorial?

Sim! Está à vontade para saltar o tutorial e desafiar as suas competências de programação como uma agente da ISA.

Ao programar, os espaços são diferentes das tabulações?

Ao usar a indentação, tenha em atenção que deverá usar um ou o outro, nunca misture os dois. A convenção indica que devem ser usados 4 espaços para a indentação e maioria dos editores de python substitui automaticamente as tabulações por 4 espaços ao fazer a indentação.

Erro! O meu código não está correto apesar da consola indicar que “Program ran successfully”.

Esta mensagem significa que seu código está correto, no entanto este não resolve a tarefa. Leia o problema novamente e verifique se o seu código faz sentido!!

Os alunos são avaliados ao jogar na plataforma ECoS?

Não. À medida que os alunos resolvem as suas tarefas eles avançam na história, aprendendo mais sobre programação e sobre a história e cultura local.

Uma avaliação informal pode ser feita, verificando a barra de progresso no ambiente de trabalho do jogo, esta mostra quantas tarefas já foram resolvidas pelo aluno.

Encontrei um bug na plataforma ECoS. Para onde posso enviar esta informação?

Por favor, envie-nos um email para support@ingeniousknowledge.com

Ou contacte-nos através do nosso website em: <http://earlycodinginschools.eu>

A sua contribuição será apreciada

Glossário

“>>>”

- A prompt por omissão do shell interativa do Python. Habitualmente vista em exemplos de código que podem ser executados interativamente no interpretador.

abs

- Retorna o valor absoluto de um número.

argumento

- Informação extra que o computador utiliza para executar comandos.

atribuição

- Coloca um valor numa variável.

bloco

- Secção de código que está agrupada.

break

- Utilizado para sair de um ciclo for ou while.

classe

- Um modelo para criar objetos definidos pelo utilizador.

compilador

- Traduz um programa escrito numa linguagem de alto nível para uma linguagem de baixo nível.

depuração (debugging)

- O processo de encontrar e remover erros de programação.

def

- Define uma função ou método.

dicionário

- Um array mutável(ou dicionário) de pares chave e valor. Pode conter tipos mistos (chaves e valores). As chaves têm que ser de um tipo hash.

ordem de interpretação

- O Python interpreta expressões da esquerda para direita. Tenha em atenção que ao avaliar uma atribuição, o lado direito é interpretado antes do lado esquerdo.

for

- Percorre um objeto, capturando cada elemento numa variável local para uso no bloco de código respectivo.

função

- Uma sequência parametrizada de declarações.

chamada de função

- Uma invocação da função com argumentos.

linguagem de alto nível

- Desenhada para ser de fácil leitura e escrita por humanos.

IDLE

- Ambiente integrado de desenvolvimento (Integrated development environment)

declaração if

- Executa, condicionalmente, um bloco de código, juntamente com else e elif (uma contração de else-if).

imutável

- Não pode ser alterado após a sua criação.

import

- Utilizado para importar módulos cujas funções ou variáveis podem ser utilizados no programa atual.

indentação

- O Python utiliza indentação de espaço em branco, ao invés de chavetas ou palavras chave, para delimitar blocos.

interpretar

- Executa um programa traduzindo-o uma linha de cada vez.

iterável

- Um objeto capaz de devolver um dos seus membros um de cada vez.

lista

- Uma lista mutável, pode conter tipos mistos.

métodos

- Um método é como uma função, mas executa-se “num” objeto.

objeto

- Qualquer dado com um estado (atributos ou valor) e comportamentos definidos (métodos).

orientado a objetos

- Permite aos utilizadores manipular estruturas de dados chamadas objetos por forma a construir e executar programas.

PEP 8

- Um conjunto de recomendações sobre como escrever código Python.

slice

- Retorna (divide) sequências em subpartes

strings

- Podem incluir números, letras e vários símbolos e têm que estar dentro de aspas duplas ou simples, se bem que as aspas simples são as mais utilizadas.

variáveis

- Espaço reservado para textos e números. O sinal de igual (=) é utilizado para atribuir valores a variáveis..

while

- Executa um bloco de código enquanto a sua condição for verdade.