

# ECoS Handbook

## What is the ECoS Platform?

Coding is the new literacy. As technology continues to grow in importance in our daily lives programming has become important in our society and an increasing number of jobs require programming skills. Fundamental programming skills are also very useful for understanding how computers and next generation technology devices work which gives those in the know a distinct advantage. We already see a lack of programmers and this could become a problem in the future. This is why programming skills are an area that promoters of STEM education focus on.

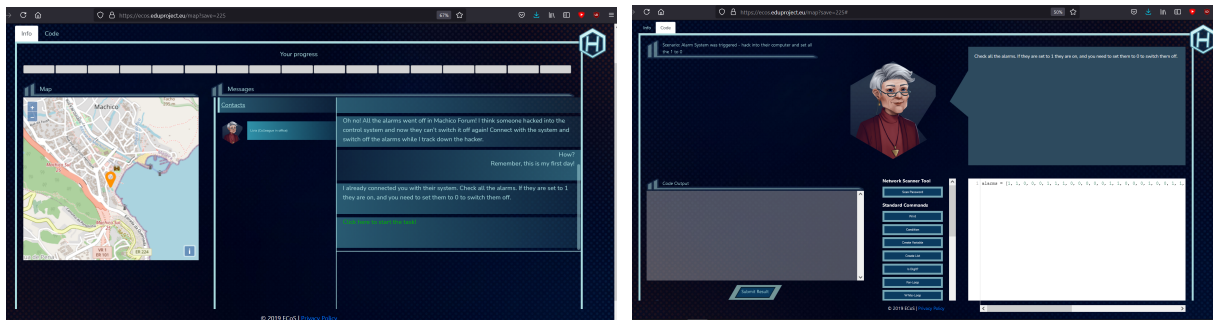
However, it is important to support the rest of the curriculum as often in the lower grades there is not a lot of time to add additional lessons. Language, history, and culture continue to be important skills for students.

The ECoS Platform is an answer to this issue, offering a fun game-based learning tool that teaches coding, encourages interest in STEM, while also teaching regional history and culture. It lets students explore a regional map and reveal locations on the map to where they are able to travel through coding. The coding tasks increase gradually in difficulty and introduce more and more concepts as the students advance in their knowledge of coding. The right line of code is important and allows the player (students) to further progress on the game. This game/learning platform is, therefore, able to teach coding skills alongside history and culture and as well as teaching how computers work and especially how to get them to do what you tell them to do all the while practising critical thinking and learning about regional history and culture.

So, the ECoS Platform (<https://ecos.eduproject.eu/>) is a cross-curricular learning tool for computational logic, coding and local culture.

While playing it students face a challenging learning environment while being supported by their teacher. The storytelling is fun, simple and engaging to motivate our learners to play and learn at the same time.

At the ECoS Platform the learner, who is an ISA (International Security Agency) agent, embarks on different assignments around their home region in order to avoid or solve cyber-attacks. Each assignment can only be solved through code. The story unfolds at the player's virtual workspace where in the INFO tab interacts with the game characters and is called for assignments associated with a spot in the map screen. At the CODE tab, if the player solves the assignment with the correct code, the situation is solved and the story advances. As the story goes on, more situations that must be solved arise, then, more regional targets become available for the players to learn about its history and cultural impact.



ISA agent workspace.

This game fills a very important gap in the efforts to increase coding skills of young students. By making it into a fun game it uses a kind of stealth learning which encourages students to learn while engaging in a fun activity. The tool can easily be used in class because it runs in a web browser and doesn't require any installation. Teachers have the freedom to let their students learn at their own pace and support them to go through the game. This also makes it an ideal tool for students to learn on their own and discover more about coding outside of class.

There are a lot of projects on national and international levels that try to improve STEM education or make STEM education more attractive. Projects that have a focus on technology will benefit from this platform and in most cases it will be very complementary to new learning approaches, projects that develop teaching materials and curricula, and projects that focus on different approaches to teaching STEM.

## What is programming?

So you want to learn programming? First, you need to know what it is. One of the most comprehensive definitions and yet easy to understand can be found in the most likely of places, the Wikipedia:

“Computer programming is the process of designing and building an executable computer program to accomplish a specific result or to perform a specific task. Programming involves tasks such as: analysis, generating algorithms, profiling algorithms' accuracy and resource consumption, and the implementation of algorithms in a chosen programming language (commonly referred to as coding). The source code of a program is written in one or more languages that are intelligible to programmers, rather than machine code, which is directly executed by the central processing unit. The purpose of programming is to find a sequence of instructions that will automate a task on a computer, often for solving a given problem. Proficient programming thus often requires expertise in several different subjects, including knowledge of the application domain, specialized algorithms, and formal logic.”

In summary, programming allows us to write a “recipe” for the computer to follow. This “recipe” can be many things, for instance a game, a word processor, a task that we need to automate, etc., the computer will then go through that “recipe” executing it exactly as it is written.

# What is Python?

Python is an interpreted high-level general-purpose programming language, its design philosophy emphasizes code readability with its notable use of indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

This philosophy of code readability is at its core and can be summarized in a very elegant way, in fact you can actually see it directly in Python, just install the official Python runtime for your computer, access it and write the following “import this” on the shell, you will be greeted with:

- 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!

Python was designed to be highly extensible as such rather than having all of its functionality built into its core it uses modules to augment its capabilities. This modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. An important goal of Python's developers is keeping it fun to use, something that surely has also helped it gain steam in education. This is reflected in the language's name, which is a tribute to the British comedy group Monty Python (and not the animal), this playfulness can also be seen in many tutorials and reference materials available, where you can find references to spam and eggs, once again references to the British comedy group. You can also see it on the Zen of Python.

All of the features and capabilities have allowed Python to be consistently ranked as one of the most popular programming languages.

# Why Python?

So why should we choose Python for our project?

Well, before anything else you should think of a programming language as a tool that allows you to create something, as such you should choose the correct tool for the job at hand. This also means that Python might not always be the answer, for instance if you are developing a AAA game, with heavy use of 3D, shaders, etc. Then it's probably not the most common choice, though not unheard of, for instance the massive online game that is EVE Online uses Python.

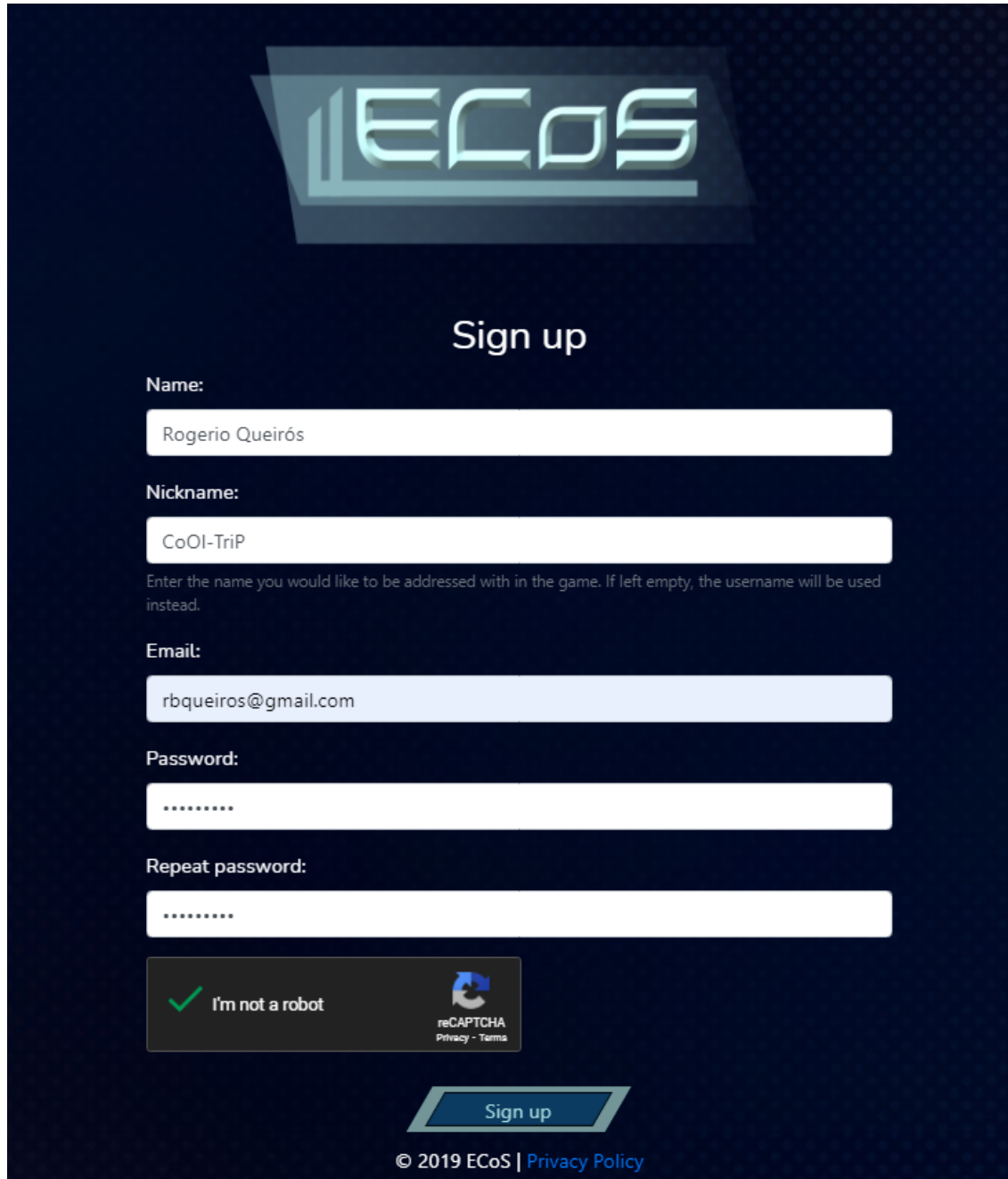
So let's go through some of the reasons that make Python one of the most popular languages today:

1. It's easy to read, write, learn and use;
2. Versatile, meaning that you have a great range of projects that can be completed using it;
3. Highly flexible and extensible, from AAA games to web pages, you can accomplish almost anything with it;
4. One of the fastest growing programming languages;
5. Programmers that are fluent in it are in high demand;
6. Very active and supportive community;
7. Has a library for every need, from math to chemistry, to astronomy Python has them all;
8. It supports multiple programming paradigms, from object oriented programming to imperative programming, all can be used, even in the same project, thus making a great teaching tool;
9. The language of choice for Artificial Intelligence and Data Science (data extraction, manipulation, etc.);
10. Supports multiple platforms, from Windows, to Android, to Linux, even embedded systems support it. You can use micro-controllers like the ESP32 and program them directly using python, no need to flash new firmware on to it whenever you change your code;
11. Open source;
12. Etc.

As you can see there are many reasons to choose Python and some of these might appeal to you. On a personal note, the code readability is one that I admire, being able to read the code that might have been written some months ago, or that a student has submitted to evaluation with little difficulty is a great achievement of this language.

# Tutorial

Register at the platform



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 both contain eight dots, indicating masked text. 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 widget 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:**  
.....

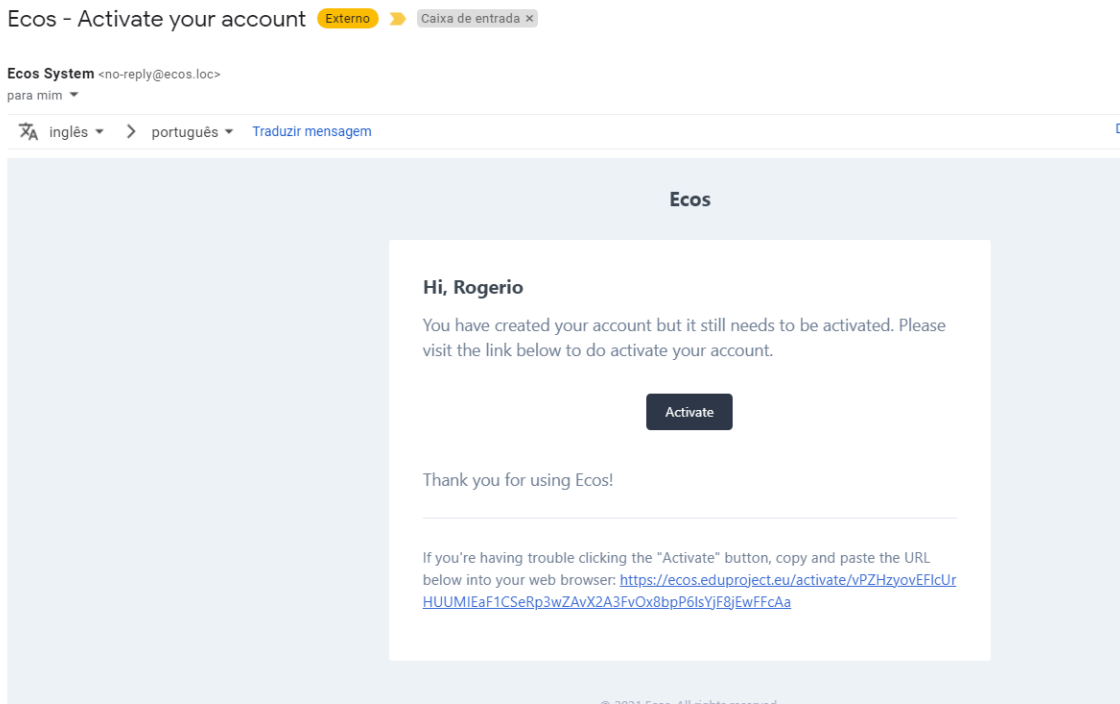
I'm not a robot  reCAPTCHA  
Privacy - Terms

**Sign up**

© 2019 ECOS | [Privacy Policy](#)

Each player to register at the platform must have a valid email, choose a nickname and select a password with at least 8 characters

# Validate Account



After the registration process, players must activate their accounts by logging in their email account and clicking on “Activate” in the email sent by Ecos Platform

## Log in

**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](#)

Users must input their email and password and click on the button “Login In”. Alternatively if they have the “Session Token” they can also log in that way.





## 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.

After login users must copy the “Cyber-Security Scenário” code to “Start Scenario”

# Introduction

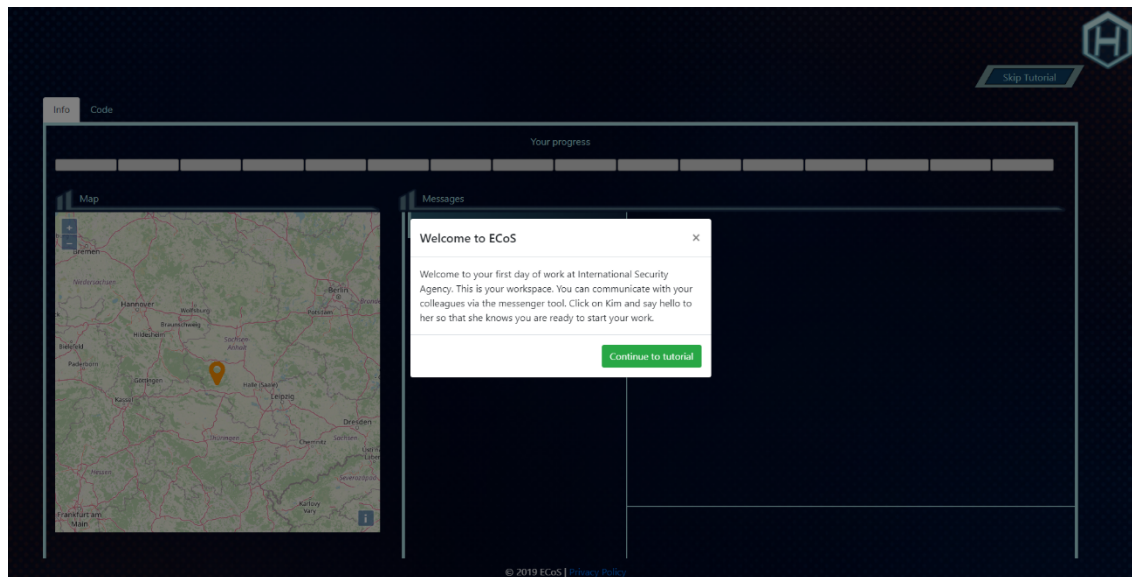
The following tutorial is going to introduce you to the basics of using ECoS as well as Python. If you feel like you do not need a tutorial, feel free to skip it using the button below.

Start

Skip Tutorial

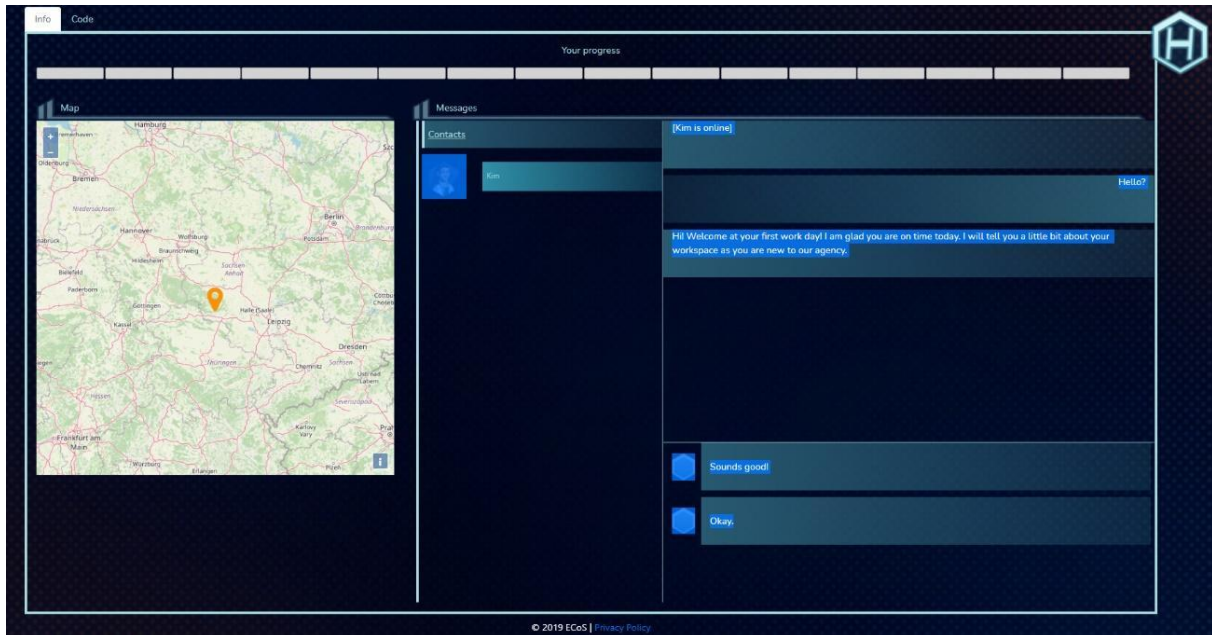
After login in for the first time, users will have the possibility to run the game tutorial or start playing without help.

## Platform Scenario



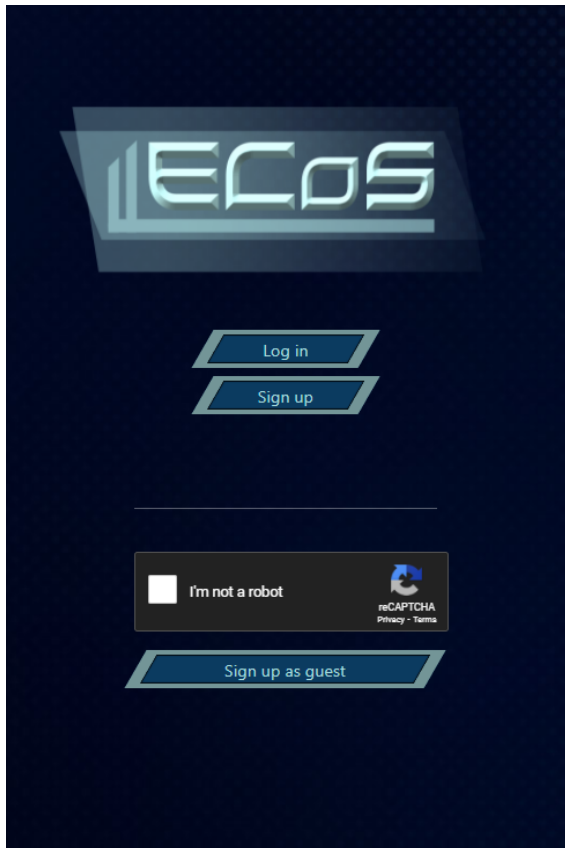
The first time accessing the game users will be greeted to a welcome message

# Agent KIM



Kim is the agent that will give you instructions through the game, from the tutorial all the way to tips to help the player complete its tasks.

## Sign up as a Guest



If you desire to explore the game without registering, you can also login as a guest by clicking at "Sign up as a guest"



## 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](#)

Copy your "Session Token" by clicking "Copy" and return "Home"



## Scenario Code

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.

Paste your "Session Code" at "Scenario Code" and click "Start Scenario"

## ECOS Scenarios

In this section we will take you through the scenarios so that you know exactly what the tasks and solutions are. The section is meant as a guide for teachers who want to use ECOS in their classroom. The content of each scenario will be shown along with the solution for each programming task. In addition each section will be commented.

## The Setting

In the ECOS game the player will be working for ISA, a security agency that monitors the city's network security. ISA is of course a fictional agency which has nothing to do with how network security really works. The scenarios are seen as a lighthearted and fun way to approach this topic and the purpose is to make the learner feel important despite being a beginner in coding.

## Characters

The characters that the players interact with are:

Michael – Michael is the boss of the player. He is very busy with boss tasks so he won't get involved in things that often.

Piedro – Every good agency needs field agents who do the dirty work. Piedro is the secret agent who goes out and gets the job done.

Livia – As an experienced security agent Livia will keep an eye on current events and give tasks to the player. She is a helpful colleague.

## Scenarios

The text content in the scenarios is always shown as chat messages in the game. Any content that is in square brackets ([ ]) is dynamic text that will be replaced with the appropriate content. This could be the player's name or the name of a location. Any content in curly braces ({ }) denotes a choice that the player has.

## Introduction

Quick Overview:

This is only an introduction for the player that launches the first scenario right away.

Chat Text:

Livia: Hi [Name]. I heard that today is your first day here. Kim told me that she already introduced you to our workspace. That is great, it means that you can take over a few of my tasks.

{Wow, you really seem to be straight forward}

{Sure, what do you want me to do?}

Livia: As you know we are responsible for the cyber-security of the locations that are situated here. Have a look at the map. These are places we take care of. Usually, there is not much to do, so grab a snack and...

Livia: Oh, hold on... we just got an alarm!

Purpose:

The reason for keeping the introduction so short is to get the player involved in coding right away. The introduction gives the players a quick idea where they are and what is expected of them.

## Scenario 1: Public Building (city hall)

Quick Overview:

This scenario introduces an easy task in which the learner has to compare a value in an array and change it to 0 if it is 1.

Chat Text:

Livia: Oh no! All the alarms went off in [city hall]! I think someone hacked into the control system and now they can't switch it off again! Connect with the system and switch off the alarms while I track down the hacker.

{How?}

{Remember, this is my first day!}

Livia: I already connected you with their system. Check all the alarms. If they are set to 1 they are on, and you need to set them to 0 to switch them off.

Purpose:

The scenario introduces an easy task to get the learner into the scenario quickly.

### Task

Check all the alarms. If they are set to 1 they are on, and you need to set them to 0 to switch them off.

```
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]
```

### Solution



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

## Scenario 2: Educational Institution

Quick Overview:

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.

Chat Text:

Livia: Well done! I am still tracking down the hacker. I already know that he connected from Asia. While I'm busy, could you take care of the local university? A friend of mine, Marcus, is the system administrator there. Let's send him an encrypted message.

{Why encrypted?}  
{What for?}

Livia: The [university] was hacked before and I want to make sure that they are alert. However, if they are already hacked I don't want the hackers to be able to read the message. So encrypt the message, Marcus will be able to decipher it.

{How do I encrypt it?}  
{What is the message?}

Livia: Just shift the letters in the alphabet by 2. An A becomes a C, a B becomes a D, a C becomes an E, and so on. When you reach the end you start at the beginning again. The Y becomes an A, The Z becomes a B. Send the message: "Hacker alert! Check your systems!"

Purpose:

This task is tricky enough to make the learner think a little. Ideally the learner breaks down the task into smaller parts and solves them one after another.

### Task

Encrypt the message that is assigned in the variable "toEncrypt" and print it using the Caesar cypher. An A becomes a C, a B becomes a D, a C becomes an E, and so on. When you reach the end you start at the beginning again. The Y becomes an A, The Z becomes a B. When the character is a space (" "), do not change anything about it.

```
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 = ""
```

## Solution

```
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: Good job! I think our work for the day is done, see you again tomorrow!

## Scenario 3: Library

Quick Overview:

This scenario is a little more complex and consists of several tasks that are chained together. The big challenge here is including a different module by using the import command.

Chat Text:

Unknown Contact: Hello? Is this the ISA?

{Yes, who are you?}

{How can we help you?}

Unknown Contact: This is Lazer Infiltration Squad. Pay 1.000.000 € or you will get into big trouble. This time you won't be able to fend off our attacks.

{Yea yea, sure}  
{Let me ask my colleagues}

Unknown Contact: This is not a joke.

Livia: Hi! Everything good? How are you progressing with your work?

{I just got a message from an unknown contact}

Livia: What? In our internal message system? That should not be possible. Give me a moment....

Livia: Yes, I can see it. There is a new contact in our database! How did he manage to create that contact entry?

{How should I respond?}

Livia: Okay. First step in such serious situations: Try to find out his IP-Address. Do you know what an IP-Address is?

{Not really}  
{What is that?}

Livia: Everyone who is surfing the internet has an IP-Address. IP-Addresses can have different schemes. A traditional IPv4 address consists of 4 numbers with a maximum of 3 digits, all separated by a dot.

Livia: For example: 135.235.86.23. Let's make sure you got this. Please, define a variable containing a valid IP address and print it for me.

## Task

Define a variable that contains an IPv4 address as a string and print it. Remember that it consists of 4 numbers with a maximum of 3 digits, all separated by dots.

## Solution

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

Chat Text:

When you have the right tools and someone's IP-Address you can track people and find out where they are connected from. For example, the police and we are able to find out a more or less exact position of the user.

{How can I do that?}

Livia: We have all kinds of different secret tools. One of them is the IP-Tracking-Device. We have the username, he is using "nightwolf". Weird username. That guy must be a real freak.

Livia: Use the "read\_ip(string ip)" function of the ip-tracking-device and enter the exact username as its parameter. The tracking device will check our logs and return his IP.

### **Task**

Define a variable which contains the hacker's username "nightwolf". Call the read\_ip() function of the IP-Tracking-Device. Check your options to your left. You can easily select the read\_ip() function there. All you need to do is enter the name of your variable as a parameter.

Print the IP the tracking device returns.

### **Solution**

```
variable_name = "nightwolf"
```

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

(Important here: Defining Variable and printing the result of the tracking device, which should be correct. The result it is returning on "nightwolf" is "183.42.58.12")

Chat Text:

Livia: Were you able to get the IP?

{Yes. The IP is: 183.42.58.12}

Livia: Okay. Good! He still seems to be online. He seems to be at [LIBRARY]. Lets try to hack him fast before he logs out of their network.

{Okay. How will you do that?}

Livia: YOU will do it. Use the Port-Checker-Module. Loop through all kinds of different ports to make sure that you find one we can use to get into his computer. I will use it and place a trojan there.

## Task

Initialize the Port-Checker-Module. Call its `set_ip(ip)` function and pass the IP you determined. Remember: The IP must be provided as a string. Remember: The IP is: 183.42.58.12

Use a for loop to iterate through the numbers 1 to 99999. Pass each number to the Port-Checker-Module's `is_port_vulnerable(port)`. Print the number if it is a vulnerable port.

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
```

## Solution

```
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)
```

Chat Text:

Livia: Did you find some vulnerable ports I can use?

{Yes. 553, 145 and 86744}

Livia: Perfect. I placed some code on his computer. Please, do me a favor. You should be able to use the functionality of the trojan now. It has a function called `get_recent_files()`. Please let it list all desktop files for you and print the names of them.

## Task

Now that the trojan is installed on the hacker's computer you can use its functions to access the computer. Use the trojan's `get_recent_files()` command, which returns a list and assigns the list to a variable. Loop through the list and print all file names in the list.

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)
```

Solution

```
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
```

Purpose:

The reason why this section is broken up into several tasks is that the learner should be introduced to the concept of calling a different class gradually. The idea is also that the learner repeats this task to become more confident in using it.

## Scenario 4: Sports Facility

Quick Overview:

This task is straightforward again. All the learner needs to do is filter out unwanted code from a string.

Chat Text:

Livia: Did it work?

{Yes, I found some very weird files. There is one File called secretTarget.txt}

Livia: Oh. That can't be good. Let me check the file.... oh....

{What?}

{Bad news?}

Livia: Their next target is.... [Sports Facility]

Livia: We should set up a firewall for them to make sure that nobody can hack them.

### Task

It seems like the hacker wants to attack [Sports Facility]! Time to set up a firewall protection for them. Write an algorithm that removes everything but zeros and ones from the input string. Every time you test it the string will have a different value, so make sure that your program can handle that. Declare a second variable which contains the first variable's value but without everything but 0 and 1. Print that string.

Solution

```
request =
```

```
“100101010001010worm?hack0101010101101malware0010110100110101virus101011010010  
100101010001111010101101010011”
```

```
result = “”
```

```
for c in request:
```

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

print(result)
```

Purpose:

This is an encouraging task again that the learner can solve easily and quickly. It is placed after the more complex scenario 3 in order to encourage the learner again.

## Scenario 5: Church

Quick Overview:

This scenario revisits the task in scenario 1 and makes it slightly more complex.

Chat Text:

Livia: Thanks. I hope they are secure enough to withstand their attacks.

Livia: I just got a call from the church. Their bells have started ringing and the electronic system that controls them is jammed. This must have been the hacker again, I wonder what he is trying to do?

{Maybe he just wants to annoy people?}

{Maybe he is testing us?}

Livia: Maybe. But we must act and put an end to this. Connect with the control system and switch off the bells.

### Task

Check the `connectedDevices` list. If the device is a church bell, switch it off in the `deviceStates` list. 0 in the `deviceStates` array means off, 1 in the `deviceStates` array means on.

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

### Solution

For `i` in `connectedDevices`:

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

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

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



Purpose:

The idea here is to challenge the learner by taking something that should be familiar now and raising the difficulty level a little.

## Scenario 6: Economy / Bank

Quick Overview:

This task entails comparing two arrays and calculating the difference in value.

Chat Text:

Michael (Boss): [Name]! What's going on! The bank was hacked! Why didn't you do anything to prevent that?

{Sorry, I was busy switching off the church bells...}

Michael (Boss): Why did you focus on the church? Nobody cares if the bells are ringing, unless it is at Sunday mornings at 10 and you are still in bed!

Livia: [Name], it seems like the hacker did set up a trap to distract us. Apparently [Sports facility] was never planned to be the target... and the timing of the church bells was certainly no coincidence.

{Damn...}

{What should we do now?}

Livia: I think the hack was successful. The hacker seems to have stolen a lot of money. I think he is very advanced. Please, help the bank to find out how much money he stole. We should at least know about that - and then we should try to catch him!

### Task

Calculate the sum of the missing money in the different account groups of the bank. Compare the lists `afterHack` and `beforeHack` and their numbers at the individual positions. If the numbers at the same position of both lists are different, add up the difference to the sum of missing money. Print the sum of missing money.

### Solution

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

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

```
sum = 0

for i in range(0, afterHack.length):
    if(beforeHack[i] != afterHack):
        sum += beforeHack[i] - afterHack[i]

print(sum)
```

Purpose:

This task introduces calculations with variables that are a little complex but still easy. The learner should not be surprised by any of this.

## Scenario 7: Transportation

Quick Overview:

Once again the task entails going through an array and doing a comparison.

Chat Text:

Livia: That's quite a lot of money. We have to catch this person, or at least get more information.

Livia: I analyzed the hacker's computer a bit more and found out his name. Also, I found out that apparently between the attack on [public building] and the attack on [bank] he travelled to this city.

Livia: His Name is Mario.

{Oh, that's good!}

{Cool. Like Super Mario}

Livia: Yeaah.... However, my idea. Since he arrived here recently, we could go to [Transportation] and check the last passenger lists for his name. One of the Marios will probably be him. That will give us more information about him.

{Wow! You are so smart.}

{Let's do this!}

### Task

Loop through the passengers list and check if the name is "Mario". If yes, check the same position of the passport list and print it.

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

### **Solution**

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

Purpose:  
The learner

## **Scenario 8: Parks / nature**

Quick Overview:

In this scenario the learner will have to work with positions in an array in order to decrypt an encrypted message.

Chat Text:

Livia: Good. There we have our Mario...

Livia: I think our target just logged into the public Wi-Fi in the park.

{I didn't even know we had public Wi-Fi in the park!}  
{Good thing he didn't use a VPN.}

Livia: Yes, I don't usually suggest using the park Wi-Fi, it isn't very secure, as you can see. Can you monitor the traffic and listen in on his messages?

{I'll see what I can do}

### **Task**

Nightwolf uses the public wifi of [Park Location] to send encrypted messages to somebody. It looks like he used the Caesar cypher which we had also used before when we sent our warnings to our friend. This time, however, it's reversed - the message can't be read until we shift all letters in the message by a certain number. I suggest that you try out different numbers and see which works for decrypting the message. Please print the message after decrypting it

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."

## Solution

```
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: Were you able to decrypt the message?

```
{ "Hey friends! I will be hacking LOCATION in just a few hours! Stay tuned for some real action. Oh, and by the way, I bought a ticket for Macbeth at THEATER for tonight. I hope they put on a good production." }
```

Purpose:

This task is a very complicated puzzle for the learners. It builds on something familiar so that the learner has some kind of orientation. However, the decrypting is quite a challenge and the learner might need help here. The purpose of this exercise is to push the learners to the limit and challenge them

## Scenario 9: Theater

Quick Overview:

This is an easy logic problem which requires logical thinking.

Chat Text:

Livia: Oh no, this is very bad news! But at least we found out his plans for tonight. Ah, I have a great idea. We'll send an agent to sit on the seat beside him to apply a tracking device. We just need to book our agent a seat next to him.

{How do we do that?}  
{I'm on it!}

Livia: Connect to the theater's booking system. Nightwolf is in seat 16. Change the reservation of seat 15 to "Lisa Fernandez". That's our agent.

### Task

Change the booking of seat 15 to our agent's name in [Theatre]. I have already logged you in to their booking system. If there is already a person assigned to seat 15, please assign that person to another seat which is free - we are hackers, but we are not mean people.

```
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"]
```

### Solution

```
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
```

Purpose:

The purpose of this exercise is to challenge logical thinking skills.

## Scenario 10: Market

Quick Overview:

This exercise introduces functions and focuses on data comparison.

Chat Text:

Livia: [Name!] Our tracking device shows us that Nightwolf is currently at the market. Pedro went there to observe him, and he can see that he is talking to some other person. However, he cannot see what that person looks like, because he is too far away!

Livia: Fortunately we have high tech security cams which can be used to identify that person. Please, log into security cam X23M1 and use its identification features to get more data.

{How do I do that?}

{Do you think I am already so comfortable with your high tech tools?}

Livia: I will show you how to do it.

### **Task**

Log into security camera “X23M1” using the `security_cameras.connect_to(“camera id”)` command. Use the `security_cameras.get_target_feature(“feature_type”)` command to get specific features of the person the hacker is meeting. Possible features are “hair\_color”, “eye\_color”, “hair\_type”, “skin\_color” and “features”. Compare these features with our criminal database. If one of the positions in the database has all features correctly, print its name.

```
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"]
```

Purpose:

This exercise lets the learner use functions-

## **Scenario 11: Power Supply / Water Supply**

Quick Overview:

In this section the learner will create a class and call it in the main code.

Chat Text:

Livia: We should create an alarm tool for [Power Supply] since we don't know the exact time when he strikes. Since we do have our tracking device on Nightwolf, it should be quite easy for us to do that.

{Okay, I will take care of this}

Livia: Sounds good. Check the functions that are available for the tracking device. They should be enough help to create a tool. Make sure to trigger the alarm when he is less than 500 meters away from [Power Supply] and we will dispatch a special police unit.

### Task

Create an alarm tool based on the tracking device. The alarm tool needs to be a class with the function “check\_status” which makes sure that Nightwolf is not within 500 meters distance to LOCATION. If he is within 500 meters range, return 1. If he is not within 500 meters range, return 0. I already prepared an empty class for you. Enter the name of the location, “LOCATION”, as parameter for the check\_distance\_to() function of the tracking device. Print it when “check\_status” is called. Call the function outside of the class to try out if it works.

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

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

Purpose:

This exercise lets the user create a class and thereby introduces an advanced part of coding.

## Scenario 12: Festival / Events

Quick Overview:

In this exercise the learner will use the brute force hacking method and write a simple loop.

Chat Text:

Livia: Nightwolf has struck again! He’s hacked the speakers at the festival and is demanding a million Euros or he’ll permanently ruin the [monument]! Everyone can hear his voice, and they are quite frightened. We need to get control of those speakers again, so he can’t panic any more people.

```
{I’ll get right on turning those off.}
{Curses! I was hoping to get a nap. Oh well.}
```

### Task

Write an algorithm that finds out the right code to access the speaker again. The code is a 4-digit number. Print all possible combinations.

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)
```

### Solution

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

Purpose:

This should feel easy to the learner. The task is an introduction to brute force hacking.

## Scenario 13: Monument

Quick Overview:

This exercise demands an extended if/elif structure.

Chat Text:

Livia: Okay, while you were turning off the speakers I notified the police, who have already arrived at the monument.

Livia: The problem is that they seem to be unable to access the bomb's controls. However, I successfully connected with the bomb, because it is connected to the internet. Unfortunately I am unable to understand its mechanism. Maybe you have an idea how it works?

```
{I'll give it all I got, boss.}
{It'll be a blast to disarm}
```

### Task



Someone set up a bomb for us! Defuse the bomb. Make sure that you print the right colors in the right order. The bomb has a color sequence which needs to be answered with the right answer. There are 4 variables defining the answer for each color. Print "red" if the sequence is yellow. Print "yellow" if the sequence is blue. Print "green" if the sequence is red. Print "blue" if the sequence is green.

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

### **Solution**

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

Purpose:

From a code point of view this exercise should be very straightforward to the learner. It is mostly an exercise in logical thinking.

## **Scenario 14: Museum**

Quick Overview:

In this last task the learner needs to deal with an expanded version of the brute force hacking method.

Chat Text:

Livia: Unbelievable! The police couldn't nab Nightwolf before he noticed the tracking device. He fled to the museum and managed to lock himself inside. All the doors are electronically sealed. You're going to have to find the code to the locks, so the police can get in there and bring him into custody. I know you can do this! The locks use a 4 digit code. Hurry!

{This guy is really making me work hard.}

{Can we go for ice cream afterwards?}

### **Task**

Write an algorithm that finds out the right code for the museum's doors. The hacker seems to have improved his password method. He uses the following format now: Number(1-9) Number(1-9) Letter (a-z). Write an algorithm that checks all possible versions of the password to find the right one and open the door

*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)
```

## **Solution**

```
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)
```

Chat Text:

Livia: Whew, that was close! Thanks to all your hard work, the police got inside the museum and arrested that troublemaker Nightwolf and his hacker buddies. No more internet for them where they're going! You did really well. Thank goodness you're on our side.

{All in a day's work.}

{So is that a 'yes' to ice cream?}

Purpose:

This task brings the scenario to a close. The exercise builds on the brute force hacking method used earlier and expands it to a more complicated scenario. The purpose here is to challenge the learner again to a logical thinking puzzle and at the same time provide a familiar element.

**THE END**

	Task	Context
Public buildings (city hall, etc.)	Alarm System was triggered - hack into their computer and set all the 1 to 0	Hacker is testing the city's defense mechanisms
Educational Institution	Send encrypted warning (caesar cypher)	We fear that educational institution was hacked as well and we want to warn the system administrator
Library	Hack hacker's computer for the first time Search through book catalogue and find the book hacker checked out	The hacker contacts the secret agency and wants money! However, in the internal logs his IP can be tracked. We hack the hacker using our IP-Checker.
Sports Facility	Establish a firewall	While scanning the hackers computer the agency finds the hacker's private files. One of them is "next_target.txt". In that document the sports facility is listed as target. The agency decides that they should set up a firewall for the sports facility.
Churches	Switch off ringing bells.	Hacker uses the church hacking as a distraction! While agents are busy dealing with the bells the hacker attacks the bank.
Economy / Bank	Check how much money the hacker stole	Hacker robs the bank instead of the first two targets
Transportation	Check the passendjfers list for the name of hacker	Since we found out that the hackers name is Mario, we check all the last flight passengers lists for the name Mario.
Parks / nature	Hacker is using public wifi and sends encrypted messages to his organisation. Decrypt the messages	We find out that hacker plans to sabotage critical infrastructure. By-product: We notice that hacker bought theater tickets for the evening.
Theater	Find Seat Number	Field agent puts tracking device on hacker
Market	Security Cameras - get different traits of a person and check against criminal database	Hacker meets accomplices, we identify them
Power Supply / Water Supply	Proximity alert – create alarm tool	We are alerted and need to protect the critical infrastructure. Hack approaches but our alarm system works and police is dispatched
Festivities / Events	Regain access of a list of hacked speakers at the event	The hacker hacks the speakers of a festivity to speak to the public, demanding money and threatening that the local monument will be blown up by a bomb if he does not get what he wants.
Monument	Return signals based on instructions to defuse a bomb somebody set up for us	Hacking group resorts to bombing the monument. We notice and diffuse the bomb.
Museum	Hacker locks all the doors - 4 digit code to find out	Hacker is mad and notices the tracking device. Before police can catch him he flees into the museum and seals all the doors. We need to open them again to help the police catch him.

```

class IpTrackingDevice:
    def read_ip(self, parameter):
        if parameter == "nightwolf":
            return "183.42.58.12"
        else:

```

```
return "Couldn't find ip"
```

## How to use ECOS in formal, non-formal and informal learning?

The ECOS platform can not only be used in a formal classroom or school context, but can also be used in non-formal and informal education.

### What is the difference between formal, non-formal and informal education?

Formal education is education that is regulated and recognised, has tests to guarantee competence and knowledge and has a vertical structure. It is compulsory education, VET and university education, for example.

Non-formal education is structured, intentional and also has learning objectives and assessments, but in this case it is voluntary and each participant decides to what extent he/she is involved and is also in charge of his/her own assessment.

Informal education is education that happens involuntarily, without the person being aware that he or she is learning. When we play we are learning rules and strategies, and often we are not even aware of it.

**In informal education**, we can for example offer young people in a youth centre the possibility to play ECOS. In this way we let them learn on their own. However, it is advisable to be available in case they have questions or need some support. The important thing is to wait for them to ask for help rather than direct their learning.

**In non-formal education** the ECOS game can be used on its own or also in the context of wider learning (e.g. within a youth exchange in the framework of the Erasmus+ programme).

-The facilitator will think about how he/she wants to use the game, whether to use the whole game or just a part of it. If the participants do not know anything about programming, it is advisable to start with the tutorial.

-You can propose individual learning or try to solve the tasks in pairs or groups.

-You can add a competitive element between groups, such as points or prizes for completing tasks, but this is not necessary.

-At the end of the process it is useful to debrief with the participants to share how the game experience was. Some questions for the debriefing could be:

-What difficulties did you encounter and how did you deal with them?

-Which parts were most motivating and which were most frustrating? How did you deal with that frustration?

-When you collaborated as a group, what role did you have within the group, was there a leader, how were decisions made?

-Finally, you should end with an evaluation, either of the ECOS session or of the whole training programme, and also with a reflection on what they have learned from this experience.

## How to use ECoS in formal learning?

Without a doubt, all formal education organisations like schools, colleges, universities in different countries have various levels of flexibility in their curriculum. It would all depend on the organisation and the teacher as well. However, there could be two options among other how to include ECoS in your system:

- Look from the unit of learning outcomes point and focus on skills that are developed while playing ECoS game. You can introduce this game in computer class, ICT, programming lessons, while learning logic or even cultural discovery aspects while solving the mystery. You can plan to have a couple of lessons with a homework and presentation about the journey, challenges and solutions they took, because in this game there could be different solutions to the problem. Students can present their way of thinking.
- Invite students to get extra credits while playing this game in their free time as extracurricular activities (special clubs).

Developed skills:

- Digital
- Computer
- Programming / coding
- Logic
- Research
- Problem solving
- Decision making
- Multicultural discovery
- Foreign language

The best thing is to try it out yourself and see the solutions that are most suitable for you and your learners.

## FAQ

Where can I play the ECoS Platform?

In any web browser at the following URL: <https://ecos.eduproject.eu/>

What if my classroom doesn't have enough devices (PCs and tablets)?

Even one device can make it possible to use the ECoS Platform in your classroom. For example, you can project the screen and play along with the class.

By applying the strategy of “learning by sharing” you can set up a group for the class if you have access to limited devices.

Remember to ask around and search on our website (<http://earlycodinginschools.eu>) when you're looking for ideas. Also, we would love to hear from you if you want to share something amazing that you came up with.

I don't feel comfortable with my programming skills but I would like to use the ECoS Platform in my classroom.

The ECoS Platform tutorial is a way to practice your coding skills starting from scratch. It introduces you to the basics of using ECoS Platform as well as Python. Follow Kim's instructions and you'll succeed.

Can I go straight to the ECoS game without doing the tutorial first?

Yes! Feel free to do it and challenge your coding skills as an ISA agent.

While coding, are spaces different from tabulations?

When using indentation, take care to always use one or the other, never mix them in your project. The standard convention states that 4 spaces should be used for indentation and most of the python editors even go so far as to automatically replace tabs with 4 spaces when doing indentation.

Failure! My code is not correct although the console window says “Program ran successfully”.

This message means that your code is correct, however it does not solve the assignment. Read the problem again and check your code if it makes sense!!

Are learners evaluated while playing the ECoS platform?

No. As learners solve their assignments they move on in the storytelling, learning more about coding and local history and culture.

An informal assessment can be done by checking their progress bar in the game workspace, it shows how many assignments have been solved by the learner.

I found a bug in the ECoS Platform. Where can I report?

Please, write us an email to [support@ingeniousknowledge.com](mailto:support@ingeniousknowledge.com)

Or contact us via our website at: <http://earlycodinginschools.eu>

Your contribution will be highly appreciated.



# Glossary

“>>>”

- The default Python prompt of the interactive shell. Often seen for code examples which can be executed interactively in the interpreter.

abs

- Return the absolute value of a number.

argument

- Extra information which the computer uses to perform commands

assignment

- Giving a value to a variable.

block

- Section of code which is grouped together

break

- Used to exit a for loop or a while loop.

class

- A template for creating user-defined objects.

compiler

- Translates a program written in a high-level language into a low-level language.

debugging

- The process of finding and removing programming errors.

def

- Defines a function or method

dictionary

- A mutable associative array (or dictionary) of key and value pairs. Can contain mixed types (keys and values). Keys must be a hashable type.

## evaluation order

- Python evaluates expressions from left to right. Notice that while evaluating an assignment, the right-hand side is evaluated before the left-hand side.

## for

- Iterates over an iterable object, capturing each element to a local variable for use by the attached block

## function

- A parameterized sequence of statements.

## function call

- An invocation of the function with arguments.

## high level language

- Designed to be easy for humans to read and write.

## IDLE

- Integrated development environment

## if statement

- Conditionally executes a block of code, along with else and elif (a contraction of else-if).

## immutable

- Cannot be changed after its created.

## import

- Used to import modules whose functions or variables can be used in the current program.

## indentation

- Python uses white-space indentation, rather than curly braces or keywords, to delimit blocks.

## interpret

- Execute a program by translating it one line at a time.

## iterable

- An object capable of returning its members one at a time.

## list

- Mutable list, can contain mixed types.

## methods

- A method is like a function, but it runs "on" an object.

## object

- Any data with state (attributes or value) and defined behavior (methods).

## object-oriented

- allows users to manipulate data structures called objects in order to build and execute programs.

## PEP 8

- A set of recommendations on how to write Python code.

## slice

- Sub parts of sequences

## strings

- Can include numbers, letters, and various symbols and be enclosed by either double or single quotes, although single quotes are more commonly used.

## variables

- Placeholder for texts and numbers. The equal sign (=) is used to assign values to variables.

## while

- Executes a block of code as long as its condition is true.