



Πληροφοριακά Συστήματα - Εργαστήριο 1ο

Χρυσόστομος Συμβουλίδης, simvoul@unipi.gr
Jean-Didier Totow, totow@unipi.gr

Γενικά θέματα

- Στοιχεία επικοινωνίας:
 - Χρυσόστομος Συμβουλίδης, simvoul@unipi.gr
Οδ. Ανδρούτσου 150, Γραφείο 207
 - Jean Didier Totow, totow@unipi.gr
Οδ. Ανδρούτσου 150, Γραφείο 502
- Παρακολούθηση εργαστηρίου: **Προαιρετική**
- Βαθμολογία εργαστηρίου: Τελική εργασία (4 μονάδες)



Ύλη του εργαστηρίου

- Γνωριμία με τη Python 3
- Docker
- Service-oriented Architecture (SOA) και Web services
- Flask microframework
- MongoDB και PyMongo (library)
- Git / GitHub



Πίνακας περιεχομένων

- Εισαγωγή με την Υπηρεσιοστρεφή Αρχιτεκτονική (SOA) και τα Web Services
 - SOA
 - REST & Restful APIs
- Γνωριμία με τη Python 3
 - Γενικές έννοιες
 - Anaconda distribution
- Git
 - GitHub account
 - Δημιουργία νέου repository
 - Βασικές εντολές



Visual Studio Code

Προτεινόμενο code editor

Download: <https://code.visualstudio.com/>

Visual Studio Code Docs Updates Blog API Extensions FAQ

Version 1.43 is now available! Read about the new features and fixes from February.

Code editing. Redefined.

Free. Built on open source. Runs everywhere.

Download for Windows Stable Build

Other platforms and Insiders Edition

By using VS Code, you agree to its license and privacy statement.

Visual Studio Code Editing evolved

Start

- New file
- Open folder...
- Add workspace folder...

Recent

- Recent Explorer
- Recent Explorer
- Recent Explorer

Help

- Press the F1 key to open the Command Palette
- Press the F1 key to open the Command Palette
- Press the F1 key to open the Command Palette

Customize

Tools and languages

- Install support for JavaScript, Python, PHP, Azure, Docker and more

Settings and keyboard shortcuts

- Install the settings and keyboard shortcuts of VS, Sublime, Atom and others

Color theme

- Make the editor and your code look the way you love

Learn

Find and run all commands

- Press the F1 key to open the Command Palette (Ctrl+Shift+F1)

Interactive playground

- Try out essential editor features in a short walkthrough



Postman

Download: <https://www.postman.com/>

The Collaboration Platform for API Development

Download the free Postman app to get started.

[Download the App](#)

10 million Developers 500,000 Companies 250 million APIs

This website uses cookies to ensure you get the best experience on our website. [Learn more](#)

[Got it!](#)

What is Postman?

is there something I can help you with?

My Workspace

Untitled Request

GET localhost:5000/show?name=Chrysosmos Symvouklidis

Parameters

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> name	Chrysosmos Symvouklidis	
<input type="checkbox"/> key	Value	Description

Response

Hit Send to get a response

For you

[Learn how to monitor a collection](#) [Start](#)



Service Oriented Architecture (SOA)

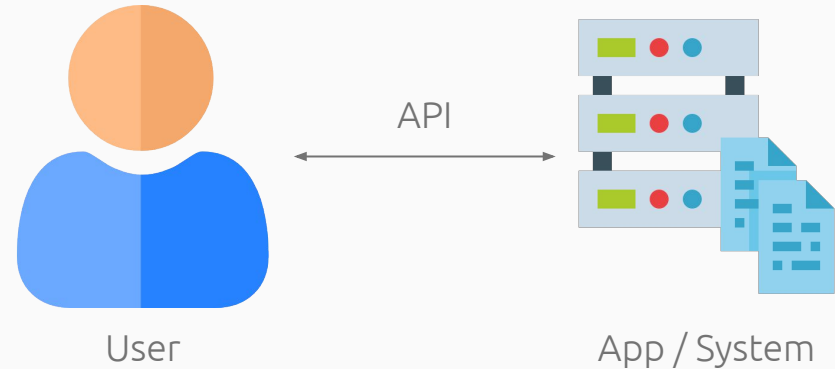
Υπηρεσιοστρεφής Αρχιτεκτονική

- Αρχιτεκτονική προσέγγιση κατά την οποία τα επιμέρους συστήματα (web services)
 - συνδέονται μεταξύ τους μέσω ενός δικτύου (internet, intranet, κλπ.)
 - Και επικοινωνούν μεταξύ τους με τη χρήση διεπαφών (interfaces)
- Γιατί SOA;
 - Καλύτερη αξιοποίηση των πόρων ενός συστήματος
 - Ελαχιστοποίηση της επαναχρησιμοποίησης κώδικα
 - Ευέλικτη και επεκτάσιμη
 - Κλπ.

APIs

Application Programming Interface (API)

- Είναι η διεπαφή (το μέσο μετάδοσης) με την οποία ένας χρήστης επικοινωνεί με ένα σύστημα
 - Στέλνει αιτήματα
 - Λαμβάνει απαντήσεις





REST & RESTful

REST

- **R**epresentational **S**tate **T**ransfer
- Αρχιτεκτονικό στυλ που χρησιμοποιείται για το σχεδιασμό εφαρμογών διαδικτύου
- Βασίζεται σε ένα *restless*, client-server πρωτόκολλο
 - Τις περισσότερες φορές χρησιμοποιείται το πρωτόκολλο HTTP

RESTful APIs

- Είναι τα API αυτά που συμμορφώνονται με τους περιορισμούς του REST

Το REST και το RESTful είναι *πρακτικά* το ίδιο πράγμα



Python 3

- Εγκατάσταση
- Γενικές έννοιες
 - Βιβλιοθήκες (Libraries) και πακέτα (Packages)
 - Συναρτήσεις (Functions)
 - Κλάσεις (Classes) και αντικείμενα (Objects)
- Anaconda
- Environments
- Requirements
 - pip & conda



Εγκατάσταση Python

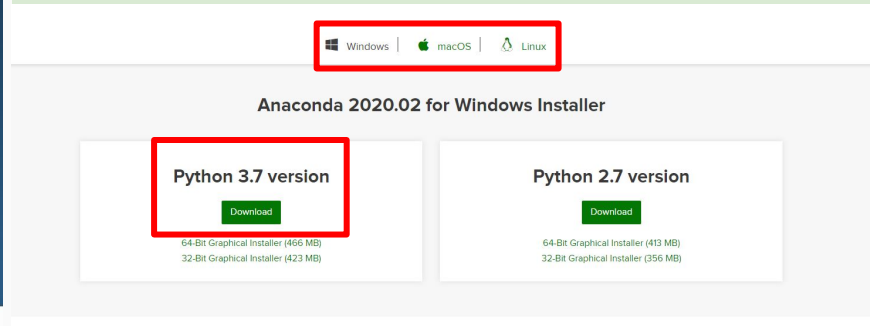
Python 3:

- <https://www.python.org/downloads/>



Anaconda (προτείνεται):

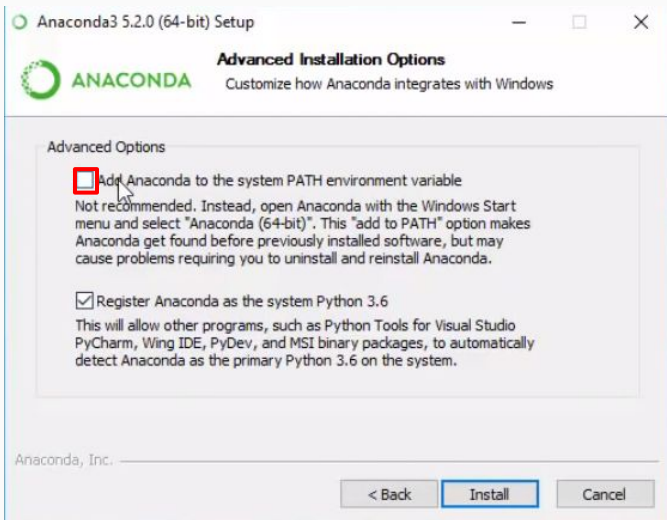
- <https://www.anaconda.com/distribution/>





Εγκατάσταση Anaconda Python (1/2)

- ❏ Add Anaconda to the system PATH environment variable



cmd / terminal: `python --version`

```
(s-ehr-cloud) csymvoul@DESKTOP-4HT3TL7:~/projects/infosyslabs/lab1$ python --version
Python 3.8.1
(s-ehr-cloud) csymvoul@DESKTOP-4HT3TL7:~/projects/infosyslabs/lab1$ conda --version
conda 4.7.12
```

```
(s-ehr-cloud) csymvoul@DESKTOP-4HT3TL7:~/projects/infosyslabs/lab1$ python
Python 3.8.1 (default, Jan 8 2020, 22:29:32)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("hello world")
hello world
>>> █
```



Εγκατάσταση Anaconda Python

Για χρήστες Windows:

Αν δεν επιλέξουμε την εισαγωγή της Python στο PATH (το πρώτο checkbox στη προηγούμενη διαφάνεια) πρέπει να την εισάγουμε μόνοι μας.

Βήματα:

1. Ανοίγουμε το CMD με δικαιώματα Διαχειριστή
2. Βρίσκουμε που έχει εγκατασταθεί η Python με την εντολή

```
where python
```

ανάλογα με τη διανομή που έχουμε εγκαταστήσει.

3. Κάνουμε copy το path
4. Και εκτελούμε την εντολή:

```
set PATH=python_path;%PATH%
```

Όπου *python_path* είναι το path της εγκατάστασης της Python που βρήκαμε στο βήμα 2



Γενικές έννοιες

- Αρχεία:
 - Τα αρχεία Python πρέπει να έχουν πάντα τη κατάληξη `.py`
file_name.py
- Python packages
 - Βιβλιοθήκες με λειτουργίες που μπορείτε να χρησιμοποιήσετε απλά εγκαθιστώντας τις:
 - `pip install packagename`
 - `conda install packagename`
 - Όταν τις κάνουμε εγκατάσταση σε κάποιο περιβάλλον, μπορούμε να τις εισάγουμε σε κάποιο πρόγραμμά μας έτσι:
 - `import package_name`



Python Virtual Environments (Anaconda)

- Εργαλείο που βοηθάει στη διατήρηση των dependencies που απαιτούνται από διαφορετικές εφαρμογές.
- Δημιουργία Conda περιβάλλοντος:
`conda create --name infosys`
- Ενεργοποίηση περιβάλλοντος:
`conda activate infosys`
- Απενεργοποίηση περιβάλλοντος:
`conda deactivate`
- Εγκατάσταση Flask:
`conda install flask`

```
(base) csymvoul@DESKTOP-4H3TL7:~/projects/Information-Systems-Lab$ conda create --name infosys
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.7.12
  latest version: 4.8.3

Please update conda by running

    $ conda update -n base -c defaults conda

## Package Plan ##

  environment location: /home/csymvoul/anaconda3/envs/infosys

Proceed ([y]/n)? y

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate infosys
#
# To deactivate an active environment, use
#
```



Python Virtual Environments

- Εγκατάσταση virtualenv:
`pip install virtualenv`
- Δημιουργία Virtual Environment:
`virtualenv venv_name`
- Ενεργοποίηση περιβάλλοντος:
`source path/to/venv_name activate`
- Απενεργοποίηση περιβάλλοντος:
`deactivate`



Install & Export Requirements

Αν χρησιμοποιούμε Anaconda:

1. Εξαγωγή των βιβλιοθηκών στο αρχείο `requirements.txt`
`conda list --export > requirements.txt`
2. Εγκατάσταση βιβλιοθηκών από το αρχείο `requirements.txt`
`conda install --file requirements.txt`

Αν χρησιμοποιούμε απλή Python 3:

1. Εξαγωγή των βιβλιοθηκών στο αρχείο `requirements.txt`
`pip freeze > requirements.txt`
2. Εγκατάσταση βιβλιοθηκών από το αρχείο `requirements.txt`
`pip install -r requirements.txt`



Τρόπος Α

```
introtopython > python_3.py > ...  
1 x = 5+5  
2 print(x)  
3 |  
4 y = 6-3  
5 print(y)
```

Αποτέλεσμα

```
(s-ehr-cloud) csymvou1@DESKTOP-4HT3TL7:~/projects/infosyslabs/lab1$ python introtopython/python_3.py  
10  
3  
(s-ehr-cloud) csymvou1@DESKTOP-4HT3TL7:~/projects/infosyslabs/lab1$
```

Τρόπος Β

```
introtopython > python_2.py > ...  
1 def add(x, y):  
2     return x+y  
3  
4 def sub(x, y):  
5     return x-y  
6  
7 x = add(5, 5)  
8 print(x)  
9  
10 y = sub(6, 3)  
11 print(y)  
12  
13
```

Αποτέλεσμα

```
(base) csymvou1@DESKTOP-4HT3TL7:~/projects/Information-Systems-Lab$ python lab1/introtopython/python_2.py  
10  
3  
(base) csymvou1@DESKTOP-4HT3TL7:~/projects/Information-Systems-Lab$
```



Εισαγωγή πακέτων
και βιβλιοθηκών

Σχόλια

Πάντα 4 spaces

Η κλάση Point

Αντικείμενο της
κλάσης Point

```
python_1.py > ...
1  import flask
2  from flask import Flask
3  from flask import Flask as F
4
5  # This is a single line comment in Python
6
7  """
8  This is how you
9  write a multiple line comment in Python
10 """
11
12 class Point:
13     def __init__(self, x=0, y=0):
14         self.x = x
15         self.y = x
16
17     def add_XY(self):
18         return self.x+self.y
19
20     def set_X(self, x):
21         self.x = x
22
23     def set_Y(self, y):
24         self.y = y
25
26     def get_X(self):
27         return self.x
28
29     def get_Y(self):
30         return self.y
31
32 point = Point(x=1, y=10)
33 print(point.get_X())
```

Constructor

Getters / Setters



Σχεσιακοί τελεστές	
Τύπος	Σύμβολο
Ισότητα	==
Μεγαλύτερο	>
Μικρότερο	<
Μεγαλύτερο ή ίσο	>=
Μικρότερο ή ίσο	<=
Διάφορο από	!=

```
lab1 > if_for > if_1.py > ...  
1   a = 10  
2   b = 15  
3  
4   if a > b:  
5       print(a , ' is bigger than ' , b)  
6   elif b > a:  
7       print(b , ' is bigger than ' , a)  
8   else:  
9       print("they are equal")  
10
```



Λογικοί τελεστές

Τύπος	Σύμβολο
ΚΑΙ	and
Ή	or
Άρνηση	not

```
lab1 > if_for > logical_op.py > ...
1   a = "this is a string"
2   b = "this"
3
4   if b not in a:
5       print("b is not in a")
6   else:
7       print("b is in a")
8
9   a = "this is a string"
10  b = "hello"
11
12  if b not in a:
13      print("b is not in a")
14  else:
15      print("b is in a")
```

```
(s-ehr-cloud) csymvou1@DESKTOP-4HT3TL7:~/projects/Information-Systems-Lab$
op.py
b is in a
b is not in a
```



Range

```
lab1 > if_for > for_loop.py > ...  
1 printme10times = "hello world"  
2  
3 for i in range(10):  
4     print(printme10times)
```

```
(s-ehr-cloud) csymvou1@DESKTOP-4HT3TL7:~/projects/Information-Systems-Lab$  
.py  
hello world  
hello world  
hello world  
hello world  
hello world  
hello world  
hello world  
hello world  
hello world  
hello world
```

Parsing a list

```
lab1 > if_for > for_list.py > ...  
1 mylist = ["apple", "lemon", "pie", 4]  
2  
3 for item in mylist:  
4     print("this item is a / an: ", item)  
5
```

```
(s-ehr-cloud) csymvou1@DESKTOP-4HT3TL7:~/projects/Information-Systems-Lab$  
.py  
this item is a / an: apple  
this item is a / an: lemon  
this item is a / an: pie  
this item is a / an: 4
```



Εύρεση στοιχείου

```
lab1 > lists > notinlist.py > ...
1  mylist = ["apple", "lemon", "pie", "Cinamon", "cinnamon"]
2
3  for item in mylist:
4      if "cinnamon" not in item:
5          print("Not here")
6      else:
7          print("Found it!")
```

```
(s-ehr-cloud) csymvou1@DESKTOP-4HT3TL7:~/projects/Information-Systems-Lab$
.PY
Not here
Not here
Not here
Not here
Found it!
```

Αντικατάσταση στοιχείου λίστας

```
lab1 > lists > replacingitem.py > ...
1  mylist = [1, 2, 3, 4, 5]
2
3  for i, item in enumerate(mylist):
4      if item == 1:
5          mylist[i] = 100
6
7  for i, item in enumerate(mylist):
8      print(i, " ", item)
```

```
(s-ehr-cloud) csymvou1@DESKTOP-4HT3TL7:~/projects/Information-Systems-Lab$
item.py
0  100
1  2
2  3
3  4
4  5
```



Αντικατάσταση στοιχείων λίστας

```
lab1 > lists > ➕ appendvextend.py > ...
```

```
1  mylist = [1, 2, 3, 4, 5]
2
3  mylist.extend("end")
4  mylist.append(["end"])
5  for item in mylist:
6      print(item)
7
8  print(type(mylist[0]))
9  print(type(mylist[-1]))
```

```
(s-ehr-cloud) csymvoul@DESKTOP-4HT3TL7:~/projects/Information-Systems-Lab$
xtend.py
1
2
3
4
5
e
n
d
['end']
<class 'int'>
<class 'list'>
```

Με το *extend* προσθέτουμε μόνο στοιχεία στη λίστα μας. Με το *append* μπορούμε να προσθέσουμε και άλλους τύπους όπως λίστες

Με το `mylist[-1]` μπορούμε να επιλέξουμε το τελευταίο στοιχείο της λίστας



Άσκηση 1

Να υλοποιηθούν οι παρακάτω συναρτήσεις:

Έχουμε τη λίστα:

```
my_list = [100, 200, 300, 199, 99, 9]
```

1. Να υλοποιηθεί η συνάρτηση `extendList` η οποία θα προσθέσει στο τέλος της `mylist` τη λίστα:

```
a_list = [101, 202, 303]
```

2. Να υλοποιηθεί η συνάρτηση `sum_list()` η οποία θα υπολογίζει και θα εμφανίζει το άθροισμα των στοιχείων της λίστας

3. Να υλοποιηθεί η συνάρτηση `max_min_item()` η οποία θα υπολογίζει και θα εμφανίζει το μέγιστο και το ελάχιστο στοιχείο της παραπάνω λίστας

```
(infosys) csymvou1@DESKTOP-4HT3TL7:~/projects/Information-Systems-Lab$ /t
py
100
200
300
199
99
9
101
202
303
The minimum item is 9
The maximum item is 303
The sum of this list is: 1513
```



Δημιουργία λεξικού

```
my_dict = { "name": "Student",  
            "course": "Information Systems",  
            "year": 2020  
}  
  
print(my_dict)
```

Εμφάνιση

```
{'name': 'Student', 'course': 'Information Systems', 'year': 2020}
```

Νέο key

```
lab1 > dicts > dicts_1.py > ...  
1  my_dict = { "name": "Student",  
2             "course": "Information Systems",  
3             "year": 2020  
4             }  
5  
6  print(my_dict)  
7  
8  my_dict["course_id"] = "DS-512"  
9  print(my_dict)
```

Εμφάνιση

```
{'name': 'Student', 'course': 'Information Systems', 'year': 2020}  
{'name': 'Student', 'course': 'Information Systems', 'year': 2020, 'course_id': 'DS-512'}
```



Αλλαγή τιμής σε κλειδί

```
my_dict = { "name": "Student",  
            "course": "Information Systems",  
            "year": 2020  
}  
  
my_dict["name"] = "Another Student"  
  
print(my_dict)
```

```
{'name': 'Another Student', 'course': 'Information Systems', 'year': 2020}
```

Επανάληψη σε λεξικό

```
my_dict = { "name": "Student",  
            "course": "Information Systems",  
            "year": 2020  
}  
  
for key in my_dict:  
    print("The key is:", key)  
    print("The value is:", my_dict[key], "\n")
```

```
The key is: name  
The value is: Student  
  
The key is: course  
The value is: Information Systems  
  
The key is: year  
The value is: 2020
```



Λίστα με λεξικά

```
lab1 > dicts > lists_dicts.py > ...  
1 my_list = [ {"title": "Information Systems", "id": "DS-512"},  
2             {"title": "C Programming", "id": "DS-501"},  
3             {"title": "Operating Systems", "id": "DS-209"}  
4 ]  
5  
6 for item in my_list:  
7     if item["id"] == "DS-512":  
8         print("There is a course with that ID.")  
9         print(item["title"])
```

```
There is a course with that ID.  
Information Systems
```

Λεξικά με λίστες

```
1 my_dict = {"grades": [5, 6, 4, 7, 10]}  
2 count = 0  
3 i = 0  
4 for key in my_dict:  
5     for grade in my_dict[key]:  
6         count += grade  
7         i += 1  
8 print("The average is equal to: ", count/i)
```

```
The average is equal to: 6.4
```



Άσκηση 2

Να υλοποιηθεί το λεξικό:

```
my_dict = {"a": 10, "b":20, "c":30 }
```

1. Να υλοποιηθεί συνάρτηση `count_dict()` η οποία θα αθροίζει τα στοιχεία του παραπάνω λεξικού
2. Να υλοποιηθεί η λίστα:

```
my_list = [my_dict]
```

και να υπολογιστεί ο μέσος όρος της μέσω της συνάρτησης `count_list()`



Git

- Σύστημα διαχείρισης εκδόσεων αρχείων (Version Control System)
- Παρακολουθεί τις αλλαγές σε ένα σύνολο αρχείων κρατώντας τα στιγμιότυπα τους
- Το σύνολο των στιγμιotypών ονομάζεται *repository*
- Υπάρχουν διάφοροι servers όπως οι:
 - GitHub
 - GitLab
 - Bitbucket
 - ...



<https://git-scm.com/downloads>

 **git** --distributed-is-the-new-centralized

Search entire site...

About

Documentation

Downloads

GUI Clients
Logos

Community

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on [Amazon.com](#).

Downloads



Older releases are available and the Git source repository is on GitHub.

GUI Clients

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

[View GUI Clients →](#)



Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

[View Logos →](#)

Git via Git

If you already have Git installed, you can get the latest development version via Git itself:

```
git clone https://github.com/git/git
```

You can also always browse the current contents of the git repository using the [web interface](#).

About this site
Patches, suggestions, and comments are welcome.

Git is a member of Software Freedom Conservancy

<https://github.com>

Why GitHub? Enterprise Explore Marketplace Pricing

Search GitHub

Sign in

Sign up

Built for developers

GitHub is a development platform inspired by the way you work. From **open source** to **business**, you can host and review code, manage projects, and build software alongside 40 million developers.

Username

Email

Password

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more](#).

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [Terms of Service](#) and [Privacy Statement](#). We'll occasionally send you account related emails.

Get started with
GitHub Enterprise



Enterprise



Talk to us





Δημιουργία project στο GitHub

The screenshot shows the GitHub homepage. In the top navigation bar, the 'New repository' button is highlighted with a red box. Below the navigation bar, the 'Repositories' section is visible, featuring a search bar and a 'New' button. The 'Working with a team?' section is also visible, providing information about GitHub's collaboration features.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner



csymvoul

Test-Repository



Great repository names are short and memorable. Need inspiration? How about [miniature-fortnight](#)?

Description (optional)

☒ Public

Anyone can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ Initialize this repository with a README

This will let you immediately clone the repository to your computer.

Add .gitignore: **None**

Add a license: **None**

Create repository

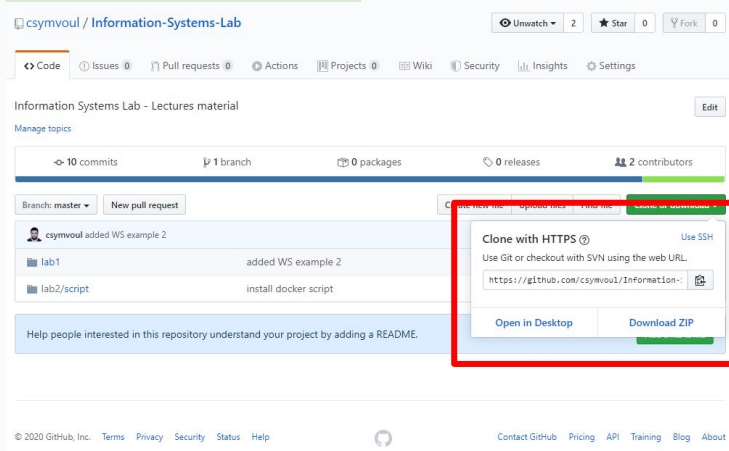
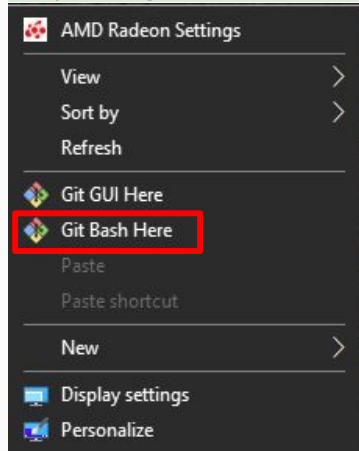


Clone project από το GitHub

1. Δεξί κλικ στο μέρος που θέλουμε να κάνουμε clone το repository

2. Clone or download
3. Αντιγραφή του link

4. git clone url



```
$ git clone https://github.com/csymboul/Information-Systems-Lab.git
Cloning into 'Information-Systems-Lab'...
remote: Enumerating objects: 55, done.
remote: Counting objects: 100% (55/55), done.
remote: Compressing objects: 100% (45/45), done.
remote: Total 55 (delta 5), reused 53 (delta 3), pack-reused 0
Unpacking objects: 100% (55/55), done.
```



Βασικές εντολές

1. Αντιγραφή project σε τοπικό φάκελο στον υπολογιστή μας:
`git clone url_to_project`
2. Προσθήκη αρχείων για το επόμενο commit
`git add file.py` Για συγκεκριμένα αρχεία
`git add .` Για όλα τα αρχεία που έχουν αλλαγές
3. Εισαγωγή των αλλαγών στο τοπικό repository (οχι στο remote - πχ σε αυτό που έχουμε στο GitHub) με κάποιο μήνυμα
`git commit -m "Commit message"`
4. Αποστολή / Ανέβασμα των αλλαγών στο remote repository
`git push`
5. Εμφάνιση λίστας με τις αλλαγές που έχουν γίνει στο τοπικό repository σε σχέση με το remote
`git status`
6. Τράβηγμα / Κατέβασμα των αλλαγών από το remote repository στο τοπικό
`git pull`



Προαιρετική Εργασία

Υλοποιήστε τις παρακάτω συναρτήσεις:

1. Υλοποιήστε τη συνάρτηση `removeDuplicates()` η οποία θα παίρνει ως είσοδο μία λίστα και θα αφαιρεί τα διπλότυπα.

Πχ: Αν πάρουμε τη λίστα

```
a_list = [10, 12, 14, 14, 16, 28, 28, 30]
```

η τελική λίστα θα πρέπει να είναι η:

```
a_list = [10, 12, 14, 16, 28, 30]
```

2. Υλοποιήστε τη συνάρτηση `sortList()` η οποία θα παίρνει ως είσοδο τη παραπάνω λίστα και θα την ταξινομεί κατά αύξουσα σειρά.

3. Να υλοποιηθούν και οι αντίστοιχες συναρτήσεις για το λεξικό:

```
a_dict= {"a":10, "b":12, "c":14, "d":14, "e":16, "f":28, "g":28, "h":30}
```