



ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ  
ΤΜΗΜΑ ΦΥΣΙΚΗΣ

ΠΡΟΒΛΗΜΑΤΑ ΚΒΑΝΤΙΚΗΣ ΦΥΣΙΚΗΣ



Αριστοτέλειο Πανεπιστήμιο  
Θεσσαλονίκης  
**ΤΜΗΜΑ ΦΥΣΙΚΗΣ**

# Εισαγωγή στα Εργαλεία του μαθήματος



Διακονίδης Θόδωρος  
Ε.ΔΙ.Π

Εργαστήριο Θεωρ. Φυσικής  
Τμήμα Φυσικής ΑΠΘ



## Ποια εργαλεία θα χρησιμοποιήσουμε

- Python Programming Language

(Με χρήση του Jupyter notebook ως editor)

- Google Colab

(Εναλλακτική λύση εκτέλεσης των προγραμμάτων python)

- Git & Github

(Git: Σύστημα ελέγχου version control, καταγραφής αλλαγών του προγράμματός, ατόμου ή ομάδας.

Github: Cloud based υπηρεσία αποθ.-παρ. Κώδικα)



## Η γλώσσα προγραμματισμού Python

- Τόσο η Python όσο και το jupyter notebook καθώς και πολλά άλλα προγράμματα και αρκετές από τις βιβλιοθήκες που θα αναφέρουμε βρίσκονται στο πρόγραμμα Anaconda.
- Για την εγκατάσταση του πακέτου:  
<https://www.anaconda.com/download>
- Για έξτρα πακέτα που θα χρειαστούν στην εγκατάσταση, θα εγκατασταθούν μέσω του Anaconda Prompt:

[Πως ανοίγουμε το Anaconda Prompt από τα Windows \(video\)](#)





## Η γλώσσα προγραμματισμού Python

- Από τις βασικές γλώσσες προγραμματισμού-υπολογισμών στη Φυσική, μαζί με Mathematica, Matlab, (αρχαιότερες Fortran, C, C++)
- Θα χρησιμοποιήσουμε ως code editor για την Python το Jupyter Notebook.

[Jupyter notebook basics webpage](#)

[Jupyter notebook basics video](#)





# Η γλώσσα προγραμματισμού Python

Για ποια πράγματα θα μιλήσουμε:

1. Γενική εισαγωγή Variables & Operators
2. Loops\_Control Statements (For, While, If, else κλπ)
3. Functions (Συναρτήσεις)
4. Classes και Modules

Για περισσότερα:

(**Eric Matthes** - Python Crash Course, A Hands-On, Project-Based Introduction to Programming)



## Python libraries

Τα βασικά πακέτα (modules) που θα χρησιμοποιηθούν:

**NumPy** : Βιβλιοθήκη για την υποστήριξη πολυδιάστατων πινάκων (arrays) και μαθηματικών λειτουργιών υψηλής απόδοσης.

**Scipy** : Επέκταση του NumPy που προσφέρει πιο εξειδικευμένες επιστημονικές και τεχνικές συναρτήσεις.

**Matplotlib** : Βιβλιοθήκη της Python για τη δημιουργία γραφικών και διαγραμμάτων.

**Sympy** : Το SymPy είναι μια βιβλιοθήκη της Python για συμβολικούς μαθηματικούς υπολογισμούς (symbolic mathematics).



ΤΜΗΜΑ ΦΥΣΙΚΗΣ Α.Π.Θ.

Πρόγραμμα Μεταπτυχιακών Σπουδών

**Υπολογιστικής Φυσικής**

Google colab

colab

- Δυνατότητα δημιουργίας-χρήσης jupyter notebooks (Python) απομακρυσμένα

(Δεν χρειάζεται καμία εγκατάσταση και μπορεί να χρησιμοποιηθεί από οπουδήποτε)

- Αποθήκευση των δεδομένων μας στο cloud (google drive, max =15Gb)
- Χρήση πόρων της google (RAM, Αποθηκευτικός χώρος, Κάρτα γραφικών για απαιτητικούς υπολογισμούς).



## Google colab (Εφαρμογή)

1. Μπείτε στο google λογαριασμό σας κατά προτίμηση από το google chrome browser.
2. Το φάκελο που αποσυμπιέσατε για να τρέξετε τα python αρχεία στο περιβάλλον anaconda αντιγράψτε τον στο google drive σας. (Drive -> + Νέο (Εικονίδιο) -> Μεταφόρτωση Φακέλου (Dropdown menu))
3. Ανοίξτε ένα οποιοδήποτε αρχείο μέσα από το φάκελο με διπλό κλικ από το google drive
4. Αρχίστε να εκτελείτε μια-μια τις εντολές του αρχείου από το browser σας.



## Google colab (εγκατεστημένες βιβλιοθήκες)

1. Για να δουμε ποιες βιβλιοθήκες υπάρχουν ήδη εγκατεστημένες στο google colab:

```
!pip -list
```

2. Για να εγκαταστήσουμε κάποια που δεν έχουμε:

```
!pip install όνομα της βιβλιοθήκης
```



## Τι είναι και τι κάνει το github



- Το GitHub είναι μια (cloud based) υπηρεσία φιλοξενίας που παρέχει στους χρήστες-προγραμματιστές ένα φιλικό περιβάλλον για τη διαχείριση αποθετηρίων Git.
- Προσφέρει διάφορα εργαλεία, όπως παρακολούθησης σφαλμάτων (bugs), διαχείρισης εργασιών και συνεχούς ενσωμάτωσης, καθιστώντας την μια πλατφόρμα κατάλληλη για συνεργασία κατασκευής λογισμικού.



## Τι θα μάθουμε εμείς

- Θα φτιάξουμε προσωπικό λογαριασμό στο github
- Θα μάθουμε πώς να φτιάξουμε (ανεβάσουμε) δικό μας repository στην πλατφόρμα.
- \*Πώς να ελέγχουμε και να αποθηκεύουμε τοπικά (στο δικό μας σκληρό δίσκο) προγράμματα στο git και πώς να τα στέλνουμε και να τα αποθηκεύουμε τα αντίστοιχα στο github.
- \*Πώς να φτιάχνουμε branches (versions) του κώδικα για να ελέγχουμε τη ροή και την χρονική εξέλιξη του κώδικα καθώς και να δίνουμε την δυνατότητα σε εμάς και τους συνεργάτες μας να κατασκευάζουμε versions του αρχικού προγράμματος για έλεγχο και πειραματισμό πάνω σ' αυτόν.



## Τι κερδίζουμε από το github

- Ο καλύτερος τρόπος αποθήκευσης των προγραμμάτων μας χωρίς το φόβο να χαθεί κάτι.
- Έλεγχος της χρονικής πορείας του προγράμματος χρησιμοποιώντας τα branches.
- Δυνατότητα προβολής του κώδικα και των projects με τη δημιουργία πορτφόλιου
- Δυνατότητα εξ αποστάσεως συνεργασίας με εξωτερικούς συνεργάτες.



## Github (εφαρμογή)

- Επισκεφτείτε την διεύθυνση [www.github.com](https://www.github.com) και ακολουθήστε βήμα-βήμα τη διαδικασία εγγραφής. (Sign up)  
(Προσοχή στην ορθή επιλογή του username σας)
- Όταν ολοκληρώσετε την εγγραφή σας (και εφόσον είστε συνδεδεμένοι στο λογαριασμό) πηγαίνετε στη διεύθυνση:  
[https://github.com/thdiakonidis/pqm\\_course](https://github.com/thdiakonidis/pqm_course)  
Και πατήστε το εικονίδιο fork πάνω δεξιά.  
Επιστρέψτε έπειτα και δείτε τα repositories σας. Τι παρατηρείτε;



## Github (εφαρμογή)

- Γνωρίζοντας το link που έχετε αποθηκεύσει το `rgm_course`, επιλέξτε να το ανεβάσετε στο google colab και να τρέξετε κάποιο(a) από τα notebooks.
- Πηγαίνοντας στα repositories φτιάξτε ένα καινούργιο, με το δικό του README αρχείο.

(Αν θέλετε ως πρότυπο του README δείτε το:

<https://github.com/thdiakon/screenmedR>)

(Χρησιμοποιήστε τις εντολές του Markdown που διδαχθήκατε)