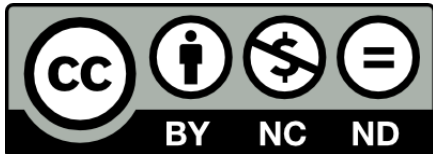


# Εισαγωγή στην Python

1





### *Copyright*

Το παρόν εκπαιδευτικό υλικό προσφέρεται ελεύθερα υπό τους όρους της άδειας Creative Commons:

- Αναφορά Δημιουργού - Μη Εμπορική Χρήση - Όχι Παράγωγα Έργα 3.0.

Για να δείτε ένα αντίγραφο της άδειας αυτής επισκεφτείτε τον ιστότοπο

<https://creativecommons.org/licenses/by-nc-nd/3.0/gr/>

Στ. Δημητριάδης, 2015

# Περιεχόμενα

- Τί είναι η Python;
- Ιστορία
- Η Python είναι Open Source Project
- Γιατί είναι δημοφιλής σήμερα η Python
- Python και άλλες γλώσσες
- Python 2.x & 3.x - Αρχεία εγκατάστασης
- Python Integrated Development Environment (IDLE)
  - Κέλυφος (Shell)
  - Συντάκτης (Editor)

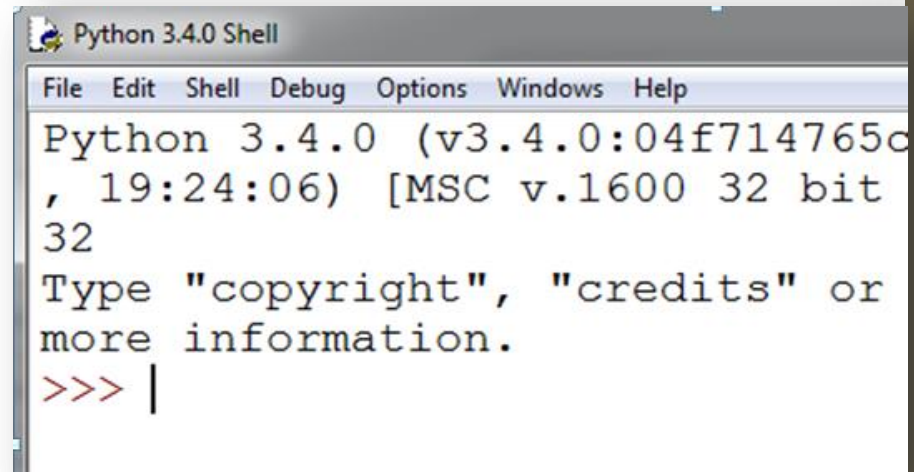
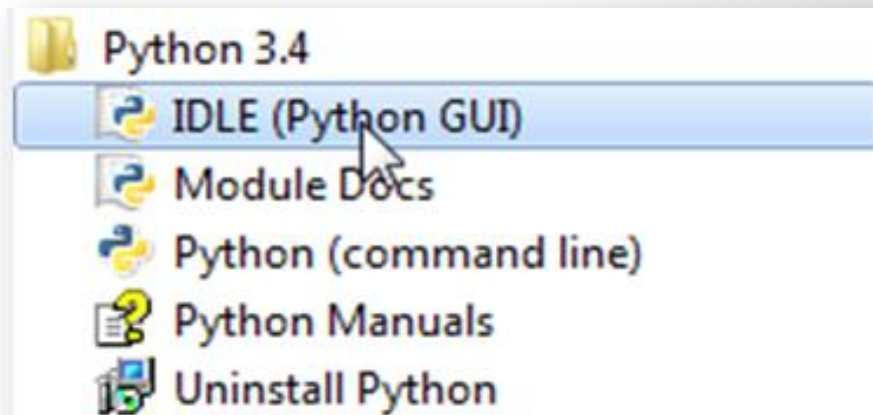
# print('Hello World!')

- 1) Ανοίξτε το Python **IDLE**
- 2) Βεβαιωθείτε ότι βλέπετε το **prompt >>>**

(αν όχι, τότε πηγαίνετε Run / Python Shell)

- 3) Πληκτρολογήστε:  
print('Hello World!') και  
πατήστε Enter

*Καλώς ήρθατε στην Python*



```
>>> print('Hello World!')
Hello World!
>>> |
```

# Τι είναι η Python

- 1

- Η **Python** είναι μια γλώσσα προγραμματισμού ευρέως χρησιμοποιούμενη:
- Γενικού σκοπού (general-purpose)
- Υψηλού επιπέδου (high-level programming language)
- Η σχεδίασή της δίνει έμφαση στην **αναγνωσιμότητα** κώδικα (readability)
- Η σύνταξή της επιτρέπει στους προγραμματιστές να εκφράζουν τις ιδέες τους γράφοντας **λιγότερες γραμμές κώδικα** (lines of code) σε σχέση με γλώσσες όπως η C++ ή Java.

# Η απλότητα της Python: 'Hello World!'

```
#include <iostream.h>
```

```
main()
```

```
{
```

```
    cout << "Hello World!";
```

```
    return 0;
```

```
}
```

```
.....
```

```
public class HelloWorld {
```

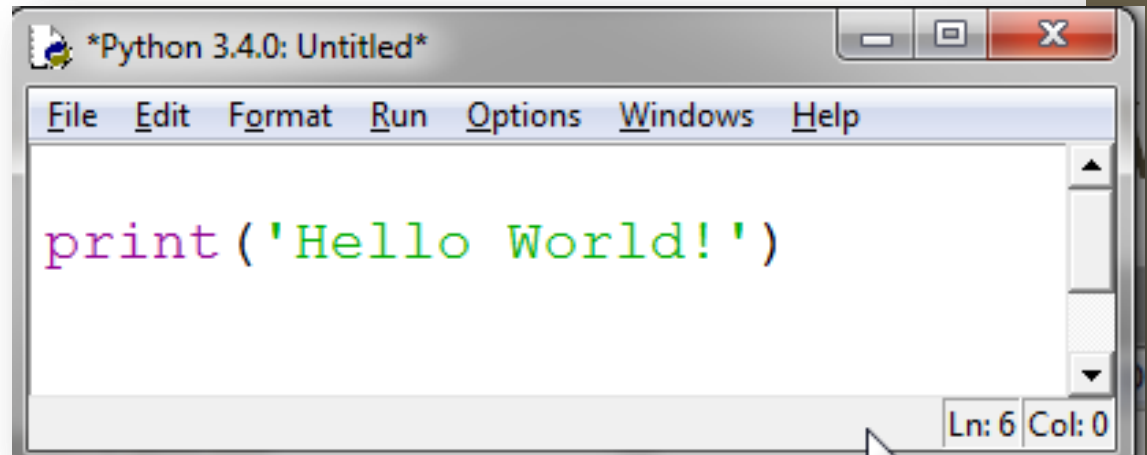
```
    public static void main(String[] args) {
```

```
        System.out.println("Hello, World");
```

```
    }
```

```
}
```

```
.....
```



# Τι είναι η Python

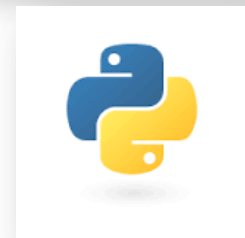
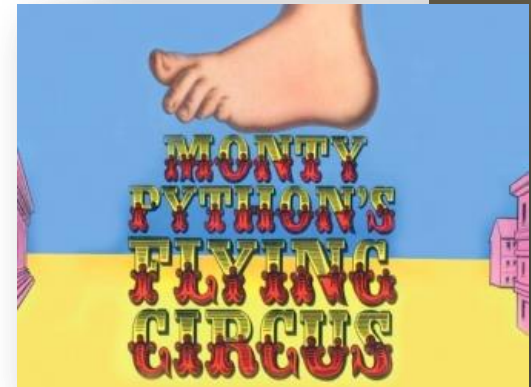
- 2

- Η Python υποστηρίζει πολλαπλά “παραδείγματα” (τεχνικές, υποδείγματα, στυλ) προγραμματισμού ([programming paradigm](#)):
  - Αντικειμενοστρεφής προγραμματισμός ([object-oriented](#))
  - Προστακτικός προγραμματισμός ([imperative](#))
  - Συναρτησιακός προγραμματισμός ([functional programming](#))
  - Διαδικαστικός προγραμματισμός ([procedural](#) styles)
- Άλλα σημαντικά χαρακτηριστικά:
  - Δυναμικός καθορισμός τύπων δεδομένων ([dynamic type](#))
  - Αυτοματοποιημένη διαχείριση μνήμης ([Memory management](#) & [Garbage collection](#))
  - Μεγάλη και ευκολόχρηστη ενσωματωμένη βιβλιοθήκη ([standard library](#))
- [Python at Wikipedia](#)

# Ιστορία

-1

- Η ανάπτυξη της Python ξεκίνησε το Δεκέμβριο του 1989
- Η έκδοση 2.0 εμφανίστηκε το 2000
- Η έκδοση 3.0 εμφανίστηκε το 2008
- Δημιουργός: Guido van Rossum στο ερευνητικό κέντρο CWI της Ολλανδίας
- Ο GvR εξακολουθεί να παίζει ένα κεντρικό ρόλο στην ανάπτυξη της γλώσσας - Σε αναγνώριση της συνεισφοράς του η κοινότητα Python του έχει απονείμει τον τίτλο-διάκριση 'BDFL' (benevolent dictator for life).
- Το όνομα της γλώσσας προέρχεται από τους Monty Python - Monty Python's Flying Circus
  - ...το λογότυπο είναι το ...φιδάκι





- Η Python δημιουργήθηκε αρχικά ως 'scripting' γλώσσα
- Scripting γλώσσα: μια γλώσσα για γράψιμο κώδικα 'ειδικού' σκοπού σε συγκεκριμένες συνθήκες,
  - Πχ. κώδικας για εργασίες διαχείρισης αρχείων σε συγκεκριμένο λειτουργικό σύστημα
- Λόγω του ευσύνοπτου κώδικά της έγινε γρήγορα δημοφιλής και εξελίχθηκε ώστε σήμερα να περιγράφεται ως:
  - Δυναμική (dynamic),
  - υψηλού επιπέδου (high level)
  - και γενικού σκοπού (general-purpose) γλώσσα

# Η Python είναι Open Source Project

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

[GO](#)[Socialize](#)[Sign In](#)[About](#)[Downloads](#)[Documentation](#)[Community](#)[Success Stories](#)[News](#)[Events](#)

```
# Python 3: List comprehensions
>>> fruits = ['Banana', 'Apple', 'Lime']
>>> loud_fruits = [fruit.upper() for fruit in
fruits]
>>> print(loud_fruits)
['BANANA', 'APPLE', 'LIME']
```



```
# List and the enumerate function
>>> list(enumerate(fruits))
[(0, 'Banana'), (1, 'Apple'), (2, 'Lime')]
```

## Compound Data Types

Lists (known as arrays in other languages) are one of the compound data types that Python understands. Lists can be indexed, sliced and manipulated with other built-in functions. [More about lists in Python 3](#)

[1](#)[2](#)[3](#)[4](#)[5](#)

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)



# Γιατί είναι δημοφιλής σήμερα η Python

- Ποιότητα λογισμικού (Software quality)
- Παραγωγικότητα ανάπτυξης (Developer productivity)
- Μεταφερσιμότητα κώδικα (Program portability)
- Βιβλιοθήκες (Support libraries)
- Ενσωμάτωση/Σύνδεση τμημάτων λογισμικού (Component integration)
- ...Enjoyment !!!
- **PYPL PopularitY of Programming Language**
- **RANKED: The Highest-Paying Programming Languages**
- **More proof that it's Python's world and we're just living in it**
- **10 Programming Languages You Should Learn Right Now**
- **Which Programming Language Should You Learn To Make Money?**
- **Examples of jobs and salaries in Greece**
- **Game Developer Salary Survey 2014**

# Τι μπορώ να κάνω με την Python;

- Η Python χρησιμοποιείται **παντού** όπου υπάρχουν ψηφιακές τεχνολογίες...
- Systems & Application Programming
- GUIs
- Internet Scripting
- Component Integration
- Database Programming
- Mobile Programming
- Rapid Prototyping
- Numeric and Scientific Programming
- Special apps: Gaming, Images, Data Mining, Robots, Excel...
- Special hardware: Arduino, Raspberry Pi, IPython for Lego Mindstorms

# Ποιος χρησιμοποιεί Python σήμερα;

- Python Success Stories
- Οργανισμοί που χρησιμοποιούν Python
- Εταιρείες στο Διαδίκτυο
  
- You Used Python to Write WHAT?
- What is Python Used For?
- Why We Choose Python

- **Απλούστερη** και **ευκολότερη** στη χρήση από τις **Java** και **C#**.
  - Η Python ξεκίνησε (και είναι και σήμερα σε σημαντικό βαθμό) ως γλώσσα scripting, αλλά η Java και η C# έχουν κληρονομήσει μεγάλο μέρος της πολυπλοκότητας και της σύνταξης μεγαλύτερων αντικειμενοστρεφών γλωσσών όπως η C++.
- **Απλούστερη** και **ευκολότερη** στη χρήση από την **C++**.
  - Ο κώδικας Python είναι απλούστερος σε σχέση με τον ισοδύναμο σε C++ και συχνά φτάνει το 1/3 ως και 1/5 σε μέγεθος, αλλά φυσικά η Python σαν scripting γλώσσα που είναι, εξυπηρετεί συχνά διαφορετικούς σκοπούς προγραμματισμού
- **Απλούστερη** και **υψηλότερου επιπέδου** από τη **C**.
  - Η αποσύνδεση της Python από την πλατφόρμα υλικού όπου εκτελείται καθιστά τον κώδικα απλούστερο, καλύτερα δομημένο, και περισσότερο κατανοητό σε σχέση με τη C (τον γεννήτορα της C++)

- Πιο **ισχυρή, γενικού-σκοπού** και διατίθεται σε **περισσότερες πλατφόρμες** υλικού σε σχέση με τη **Visual Basic**.
  - Η Python είναι πλουσιότερη γλώσσα που χρησιμοποιείται ευρύτερα και ο "ανοικτού κώδικα" ('open source') χαρακτήρας της την αποδεσμεύει από τον έλεγχο μιας και μόνης εταιρίας
- Είναι πιο **ευανάγνωστη** και **γενικού-σκοπού** από την **PHP**.
  - Η Python χρησιμοποιείται επίσης στην ανάπτυξη δικτυακών τόπων, αλλά επίσης χρησιμοποιείται σχεδόν σε κάθε περιοχή χρήσης ψηφιακών τεχνολογιών (όχι μόνον «υπολογιστών»), από τη ρομποτική μέχρι πχ. τη σχεδιοκίνηση σε ταινίες (animation) και ψηφιακά παιχνίδια (games)
- Είναι πιο **ισχυρή** και **γενικού-σκοπού** από τη **JavaScript**.
  - Η Python έχει μεγαλύτερο πλήθος εργαλειοθηκών/βιβλιοθηκών και δεν είναι τόσο στενά δεσμευμένη με την ανάπτυξη ιστοσελίδων. Χρησιμοποιείται επίσης σε επιστημονικές εφαρμογές (scientific modeling), έλεγχο οργάνων (instrumentation), κλπ.

Python **2.x** & **3.x**

**Αρχεία εγκατάστασης** Python στον  
υπολογιστή σας



# Python **2.x** & **3.x**

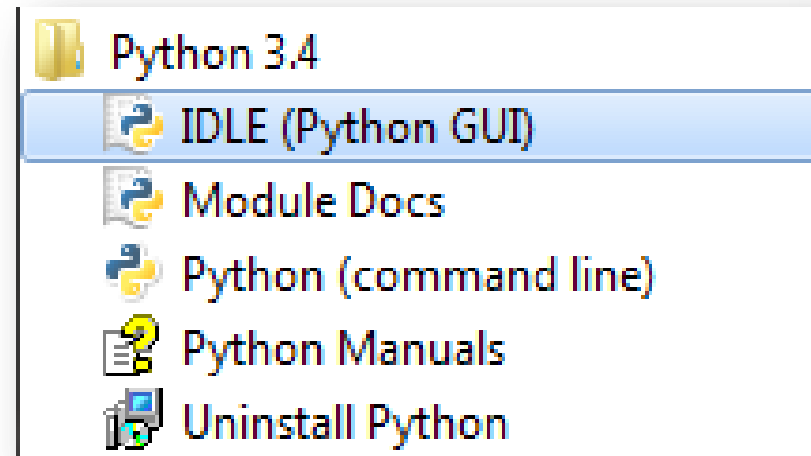
- Ο κόσμος της Python είναι ...χωρισμένος στα δύο!
- Η παλαιότερη έκδοση **2.x** (σήμερα διαθέσιμη η **2.7**)
- Η νεότερη έκδοση **3.x** (σήμερα διαθέσιμη η **3.4**)
  - “Python 2.x is legacy, Python 3.x is the present and future of the language.”
- Η 2.x έχει ακόμη και σήμερα περισσότερες **βιβλιοθήκες** (Libraries) για διάφορες εφαρμογές
- Η 3.x είναι η εξέλιξη με περισσότερα **ενσωματωμένα καινοτόμα χαρακτηριστικά**
- Όμως η **μεταγραφή** του κώδικα από τη 2 στην 3 (ή αντίστροφα) μπορεί να γίνει αυτόματα με κατάλληλα προγράμματα και **όλο και περισσότερες** βιβλιοθήκες προσφέρονται σε εκδόσεις συμβατές με την 3

# Python & Βιβλιοθήκες

- Ένα χαρακτηριστικό της ισχύος της Python είναι πως έχουν αναπτυχθεί **πολυάριθμες βιβλιοθήκες** για κάθε είδους προγραμματιστική εργασία
- Έτσι ο προγραμματιστής της Python για να κάνει μια συγκεκριμένη εργασία επιλέγει και εγκαθιστά και την **αντίστοιχη βιβλιοθήκη**
  - (για την έκδοση της γλώσσας με την οποία εργάζεται)
- Πχ. για την διαχείριση της οθόνης, διαχείριση πολυμέσων και ανάπτυξη παιχνιδιών μια «κλασσική» βιβλιοθήκη είναι η **Pygame**

# Εγκατάσταση Python

- Στο μάθημα χρησιμοποιούμε την έκδοση 3.4.0 της γλώσσας για λειτουργικό Windows.
- Μπορείτε να κατεβάσετε και να εγκαταστήσετε την πιο πρόσφατη έκδοση της γλώσσας για το λειτουργικό σύστημα που χρησιμοποιείτε από [εδώ](#)
- Στο φάκελο όπου αποθηκεύθηκε τρέξετε το αρχείο και ακολουθήστε τις απλές οδηγίες εγκατάστασης
- Για να ξεκινήσετε το Ολοκληρωμένο Περιβάλλον Ανάπτυξης της Python (Integrated DeveLopment Environment) επιλέγετε IDLE (Python GUI) στον κατάλογο επιλογών που θα δημιουργηθεί, όπως βλέπετε στην εικόνα.

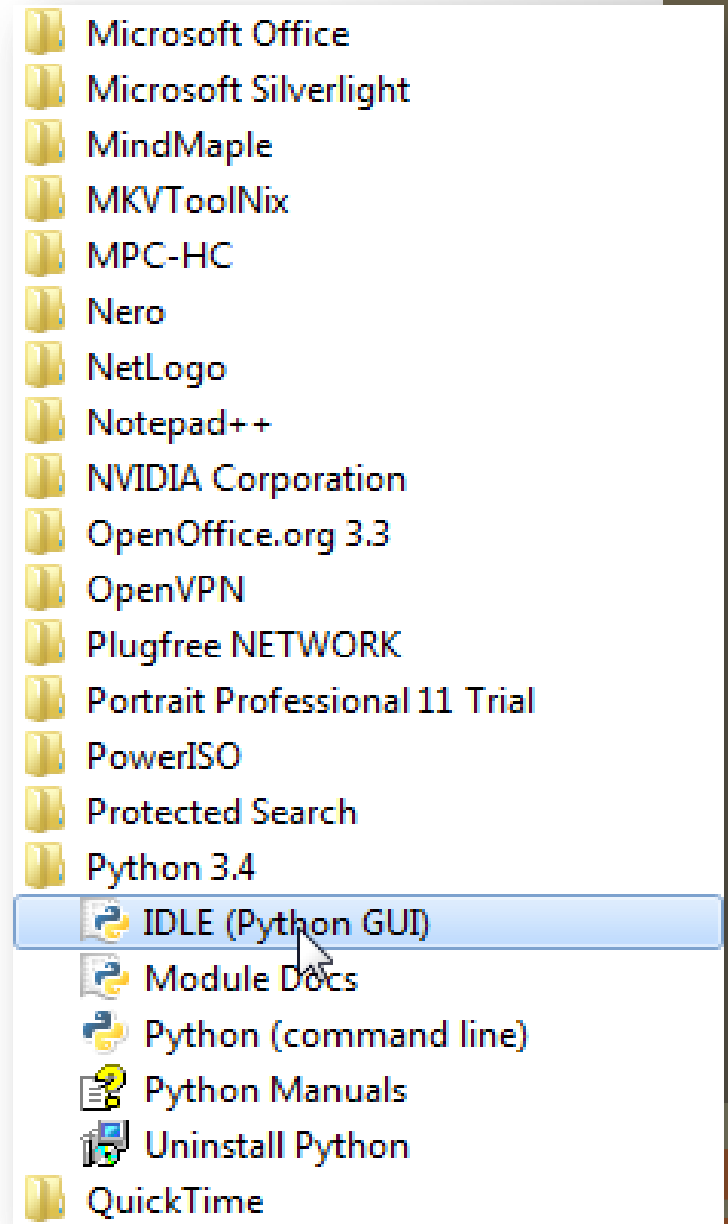


# Python **IDLE** Integrated Development Environment

Το ολοκληρωμένο περιβάλλον ανάπτυξης  
της Python

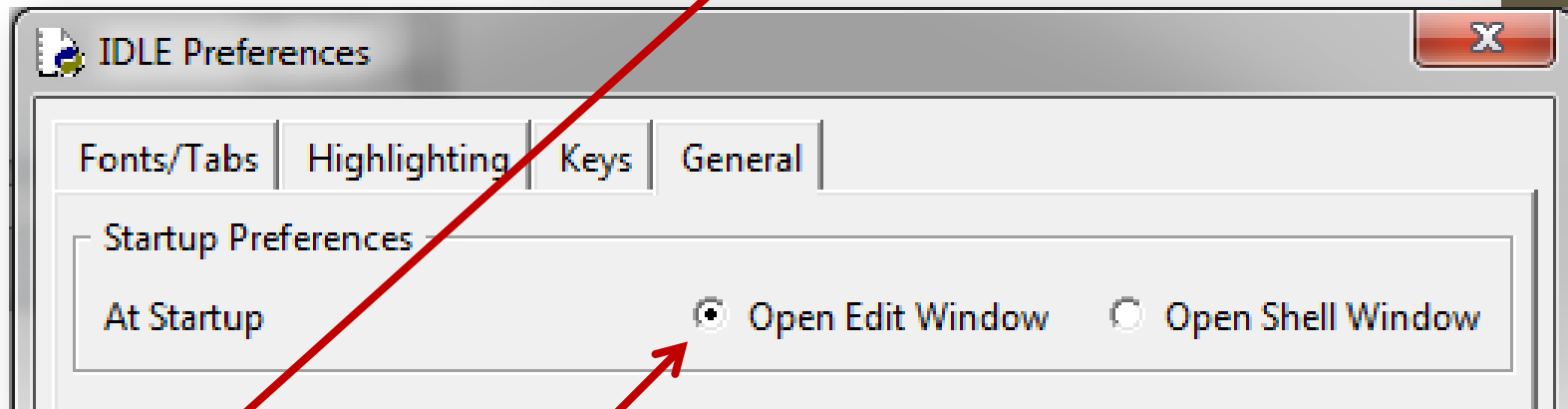
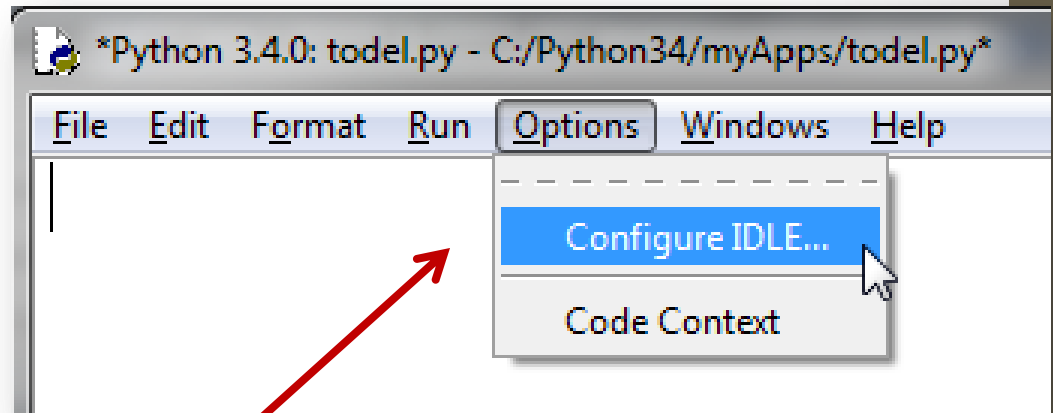
# Python IDLE

- Μετά την εγκατάσταση της Python\* βρείτε στον υπολογιστή σας το Ολοκληρωμένο Περιβάλλον Ανάπτυξης
- Python **IDLE**
- Integrated **DeveLopment Environment**
- \* Σε MacOS & Linux η Python είναι προεγκατεστημένη



# Διαμορφώστε (Configure) το IDLE

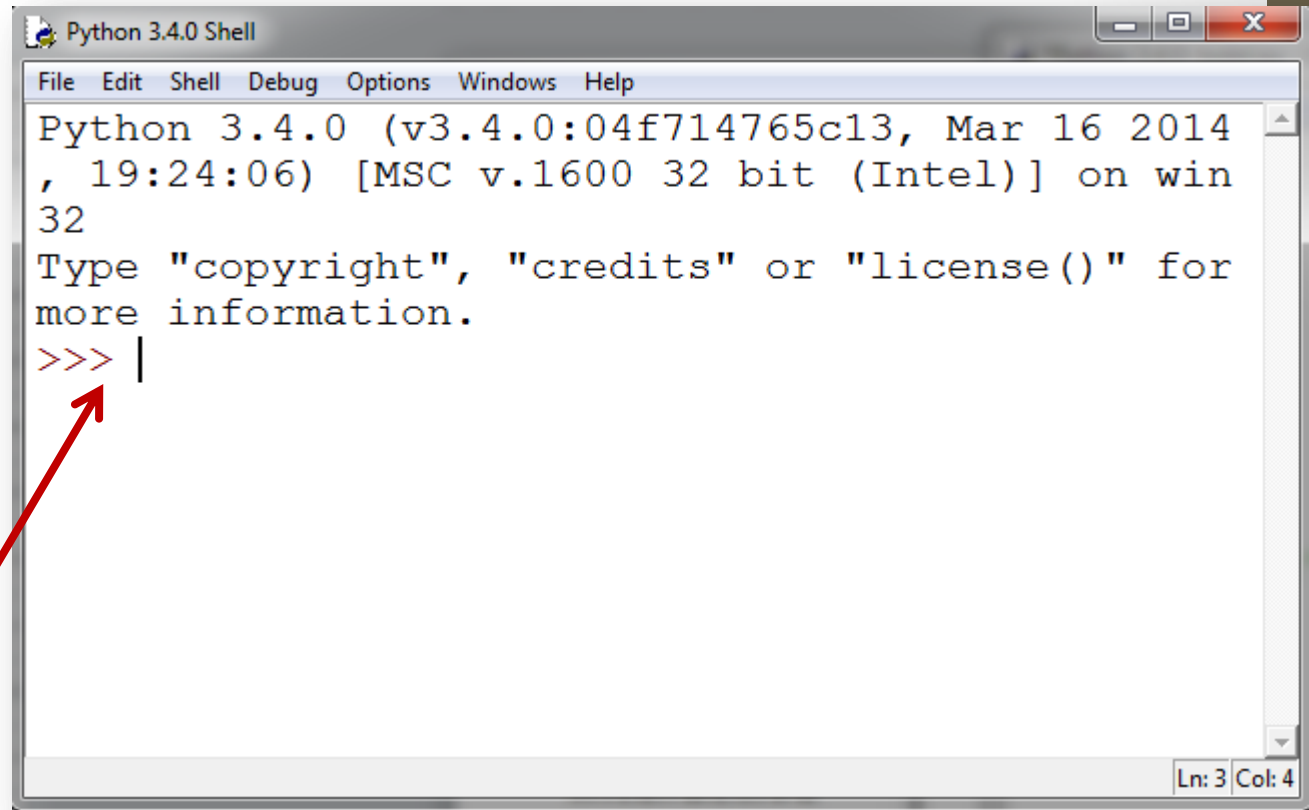
- Το IDLE μπορεί να εμφανίζεται με δύο μορφές/περιβάλλοντα:
- **Edit** Window (συντάκτης)
- **Shell** Window (κέλυφος)



- Διαμορφώστε το περιβάλλον ώστε να ανοίγει στη μορφή Edit

# Python IDLE: the **Shell** window (κέλυφος)

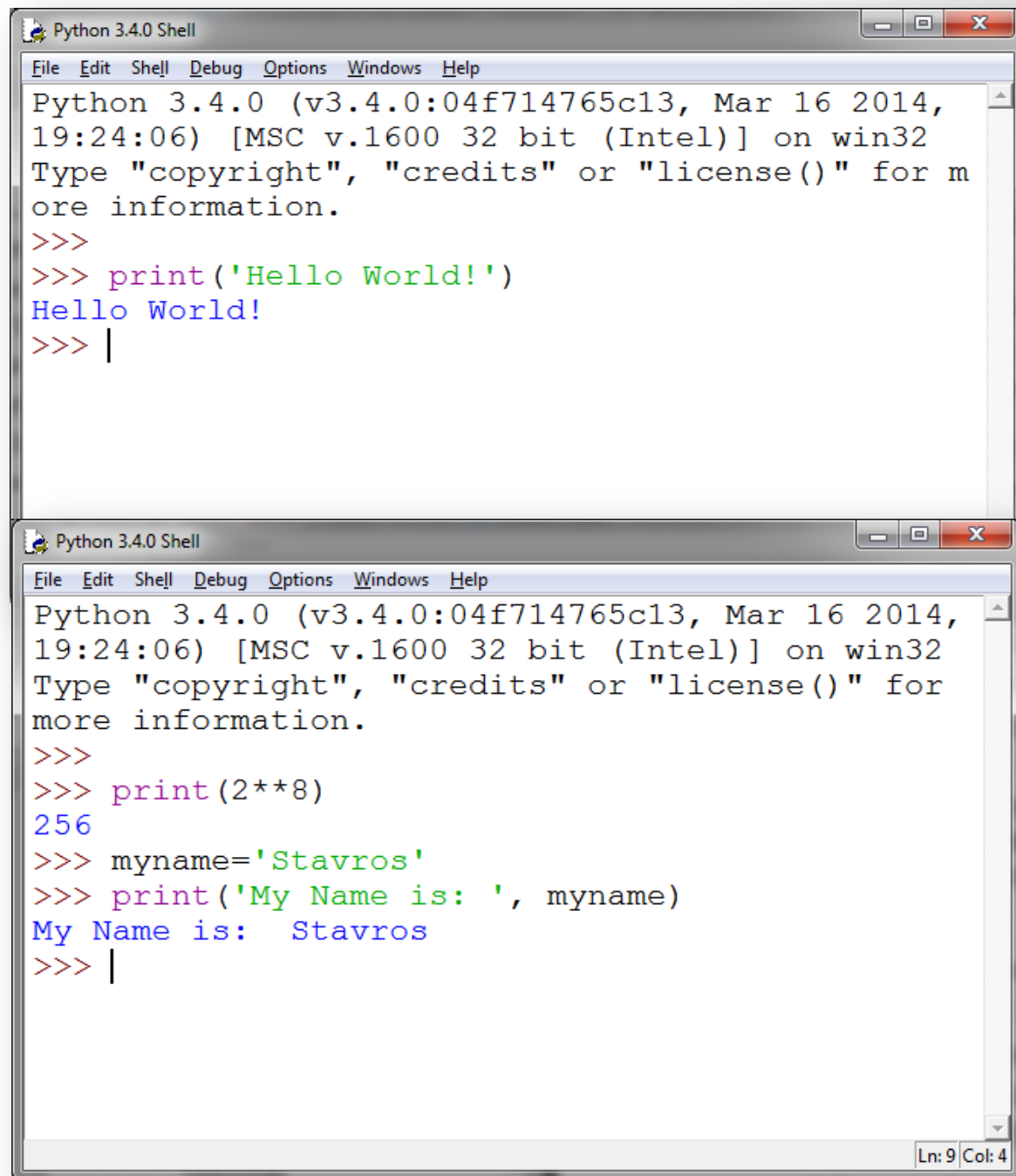
- Το κέλυφος (Shell) προσφέρει έναν **άμεσο τρόπο** εκτέλεσης εντολών
- **Interactive prompt**



```
Python 3.4.0 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.0 (v3.4.0:04f714765c13, Mar 16 2014
, 19:24:06) [MSC v.1600 32 bit (Intel)] on win
32
Type "copyright", "credits" or "license()" for
more information.
>>> |
Ln: 3 Col: 4
```

# Interactive prompt

- Δοκιμάστε να γράψετε στο κέλυφος τον κώδικα που βλέπετε δεξιά
- (Enter μετά από κάθε εντολή)
- Πώς αποκρίνεται;



The image shows two screenshots of the Python 3.4.0 Shell window. The top screenshot shows the initial prompt and the execution of `print('Hello World!')`, which outputs `Hello World!`. The bottom screenshot shows the execution of `print(2**8)` (outputting `256`) and `print('My Name is: ', myname)` (outputting `My Name is: Stavros`).

```
Python 3.4.0 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.0 (v3.4.0:04f714765c13, Mar 16 2014, 19:24:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
>>> print('Hello World!')
Hello World!
>>> |

Python 3.4.0 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.0 (v3.4.0:04f714765c13, Mar 16 2014, 19:24:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
>>> print(2**8)
256
>>> myname='Stavros'
>>> print('My Name is: ', myname)
My Name is: Stavros
>>> |
```

Ln: 9 Col: 4



# Γιατί κέλυφος και interactive prompt?

- Το κέλυφος προσφέρει δύο βασικές λειτουργίες
- (1) **Πειραματισμός**
- Πειραματιστείτε δοκιμάζοντας εντολές που δεν γνωρίζετε και κατανοήστε τη λειτουργία τους, πχ.:
  - `>>> 'Spam!' * 8` **# Learning by experimentation**  
`'Spam!Spam!Spam!Spam!Spam!Spam!Spam!Spam!'`
- (2) **Άμεσος έλεγχος βιβλιοθηκών**
- Ελέγξτε γρήγορα τη διαθεσιμότητα βιβλιοθηκών και τον κώδικά τους, πχ.:
  - `>>> import os`
  - `>>> os.getcwd()` **# Testing on the fly**  
`'C:\\Python34'`

# Δείτε...

- α) ... ποια έκδοση Python έχετε εγκατεστημένη

```
>>> import sys
>>> print(sys.version)
3.4.0 (v3.4.0:04f714765c13, Mar 16 2014, 19:24:06) [MSC v.1600 32 bit (Intel)]
>>>
```

- β) ... σε ποιο λειτουργικό σύστημα

```
>>> print(sys.platform)
win32
>>> |
```

# Άμεση βοήθεια στο κέλυφος

- Δοκιμάστε στο κέλυφος:
- **help( )** : ενεργοποιεί το help mode για άμεση παροχή βοήθειας
- **help > topics**
- Δείτε τα διαθέσιμα θέματα για βοήθεια
- **help>quit**
- Εγκαταλείπετε το help mode

```
>>> help()
```

```
Welcome to Python 3.4!
```

```
help> topics
```

```
Here is a list of available topics.  
help.
```

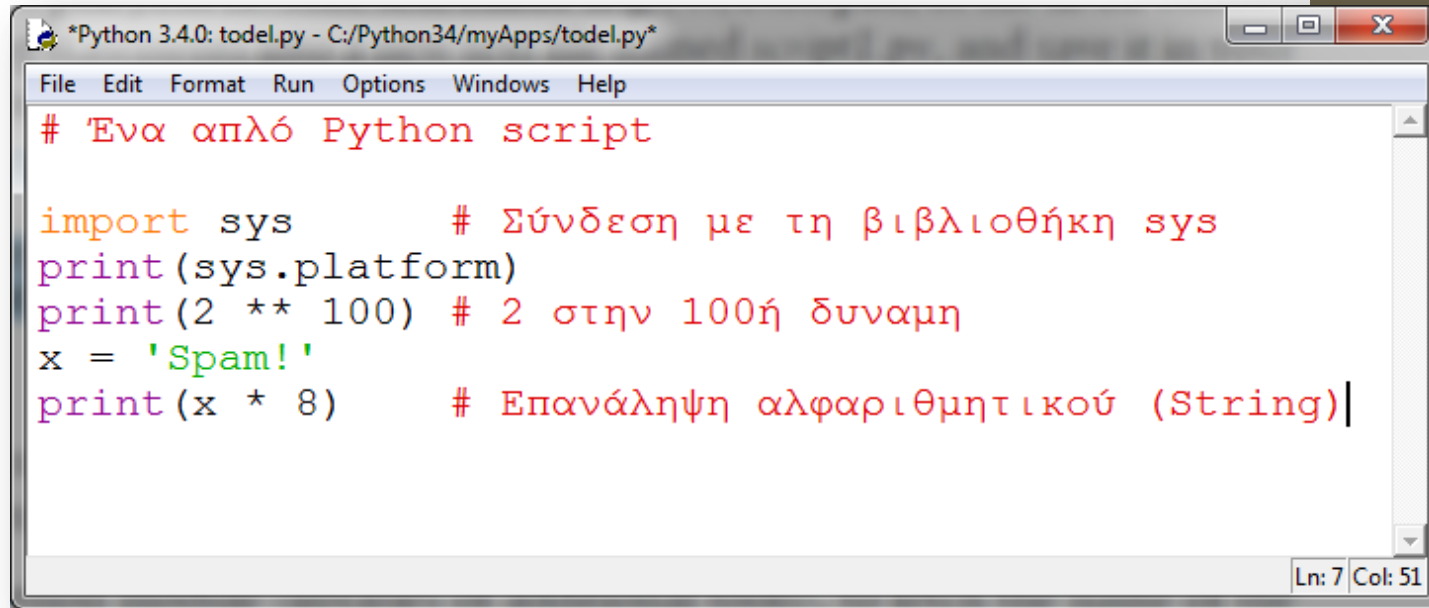
ASSERTION	DELETION
ASSIGNMENT	DICTIONARIES
ATTRIBUTEMETHODS	DICTIONARYLITERAL
ATTRIBUTES	DYNAMICFEATURES
RIBUTES	
AUGMENTEDASSIGNMENT	ELLIPSIS

```
help> quit
```

```
You are now leaving help
```

# Python IDLE: the **Edit** window (συντάκτης)

- Στον παράθυρο του συντάκτη (edit) **γράφουμε** κώδικα,
- τον **εκτελούμε** (**F5**) δοκιμαστικά,
- τον **αποθηκεύουμε**, κλπ.

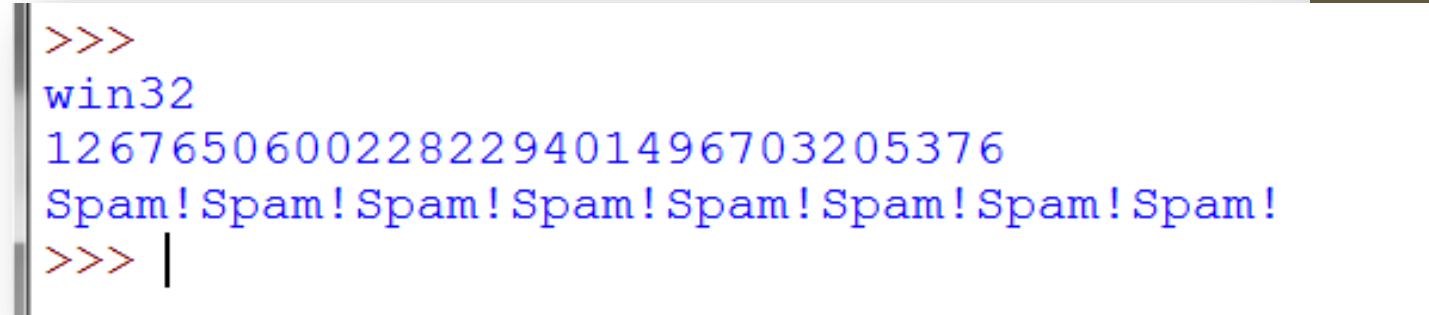


The screenshot shows the Python IDLE Edit window for a file named 'todol.py'. The menu bar includes File, Edit, Format, Run, Options, Windows, and Help. The code in the editor is as follows:

```
# Ένα απλό Python script

import sys          # Σύνδεση με τη βιβλιοθήκη sys
print(sys.platform)
print(2 ** 100)      # 2 στην 100ή δύναμη
x = 'Spam!'
print(x * 8)         # Επανάληψη αλφαριθμητικού (String)
```

The status bar at the bottom right indicates 'Ln: 7 Col: 51'.



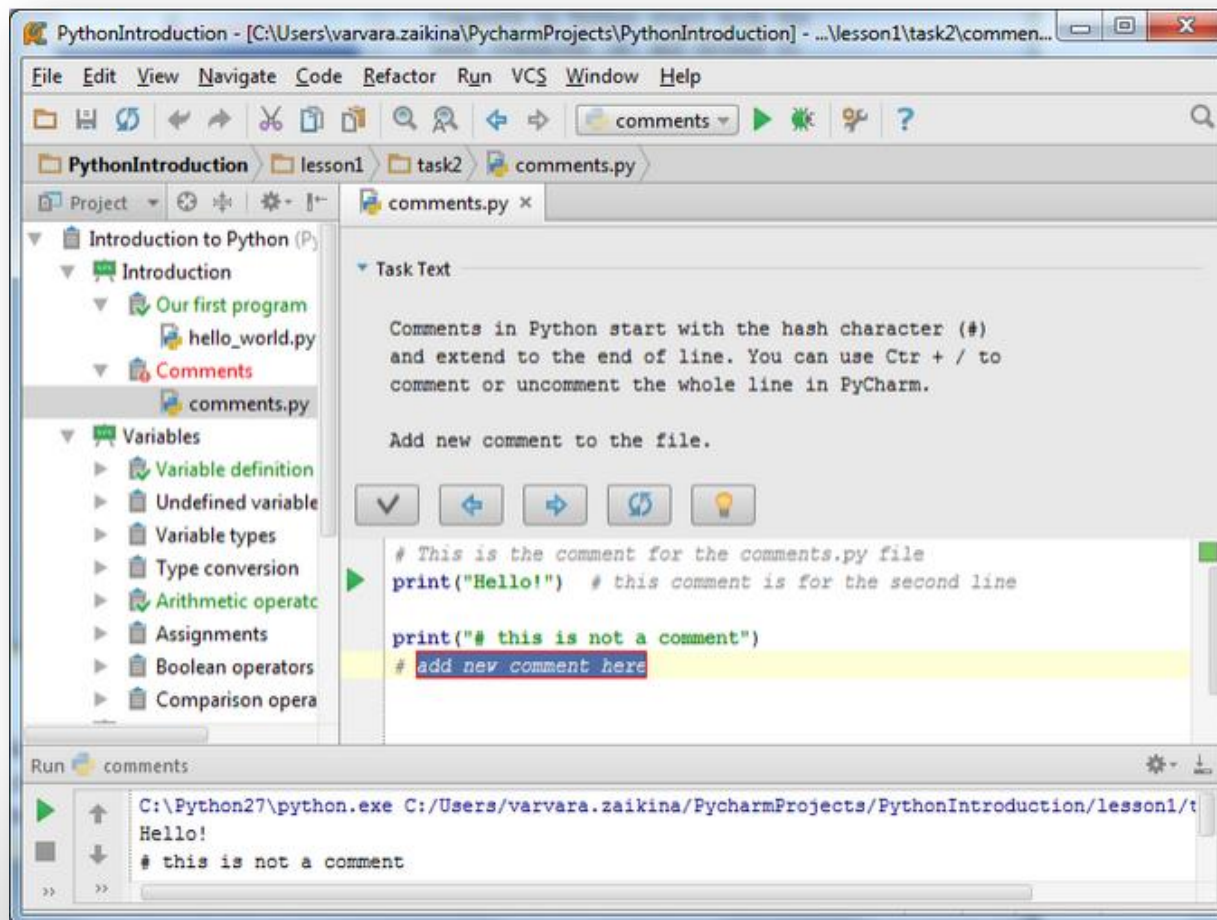
The screenshot shows the Python IDLE Shell window with the following output:

```
>>>
win32
1267650600228229401496703205376
Spam! Spam! Spam! Spam! Spam! Spam! Spam! Spam!
>>> |
```

- Όταν τρέχουμε τον κώδικα του **συντάκτη** τα αποτελέσματα εμφανίζονται στο **κέλυφος**

# PyCharm

- Αν θέλετε έναν περισσότερο υποστηρικτικό Python Editor δοκιμάστε την εκπαιδευτική έκδοση του PyCharm
- **PyCharm Educational Edition**



# Wingware Python IDE

The screenshot displays the Wingware Python IDE interface. The main window is titled "location.py (wingutils): ide-5.0-qt.wpr: Wing IDE". The code editor shows the following Python code:

```
return self

#-----
def __lt__(self, other):
    """ Override __lt__ (less-than) to define sort order. Note that it
    does not get called on equality or inequality testing. """

    if self == other or not isinstance(other, CResourceLocation):
        return False

    return self.furl < other.furl

#-----
def GetUrl(self):
    """ Get name of location in URL format """
    if self.
        #-----
        def
            #-----
            s
            #-----
            i
            #-----
            #-----
            raise IOError('Cannot open FIFOs')
            if 'w' not in mode and s.st_size > kMaxFileSize:
                raise IOError('File too large: size %d > %d' % (s.st_size, kMaxFileSize))
```

A context menu is open over the code, listing methods of the `Child` class:

- Child
- DeleteResource
- DoesResourceExist
- EnumerateChildren
- EnumerateFilesAndDirs
- fName
- furl
- GetByteCount
- GetLastModificationTime
- GetParentDir

The Source Browser on the right shows a list of methods, with `GetLocation(self)` selected. The Completion Assistant at the bottom right shows the following information:

Completion: `self.Child`

Runtime type: [callable method](#)  
[CLocalFileLocation.Child](#)  
`def CLocalFileLocation.Child(self)`

Join a location and one or more simple names. [PEP287](#)

Current Value: `<bound method CLocalFileLocation.Child of file:/Users/sdeibel/doc/woodworking/tables.tex>`

The bottom status bar shows "Line 659 Col 12 \* [Edit]".