DS2000 – Programming with Data

Introduction

Northeastern University

The *Programming with Data* Track

- Northeastern's Data Science curriculum includes three courses in Python programming:
  - DS 2000: Programming with data
  - DS 2500: Intermediate programming with data
  - DS 3500: Advanced programming with data

- All three courses leverage the python software development ecosystem.

- The three-course programming track follows a logical progression from learning first principles of software development in DS2000 (with no programming experience required) to building object-oriented libraries and data applications in DS2500, to developing more complex *enterprise-grade systems* for data engineering and data science in DS3500.

- The DS Programming with Data track provides an alternative to the traditional CS-oriented *fundamentals* sequence (Fundamentals 1 and 2 and Object-Oriented Design) with special emphasis on data-centric problem solving, analytics, and engineering.

Northeastern University
Khoury College of Computer Sciences

Pathways to Code @ Northeastern

Northeastern University
Khoury College of Computer Sciences

The Data Science Track

DS 2000: Programming with data
DS 2500: Intermediate programming with data
DS 3500: Advanced programming with data

The Computer Science Track

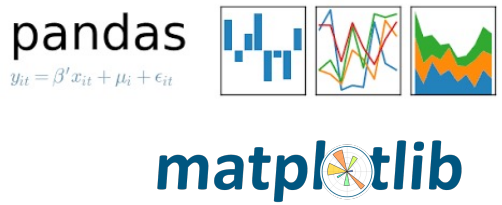
CS 2500: Fundamentals of Computer Science
CS 2510: Fundamentals of Computer Science II
CS 3500: Object-oriented Design

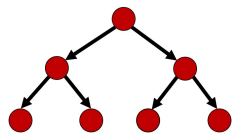
Take DS 2000 / 2500 / 3500 if:

- You are a DS major or DS minor
- You are a non-CS / non-DS major looking to gain practical experience with computer programming and data-centric problem solving.
- You want to become proficient in Python programming for data analysis and visualization, or for developing data-centric applications in business, finance, healthcare, or the sciences.

Take CS 2500 / 2510 / 3500 if:

- You are a CS major or CS minor
- You are a DS major interested in obtaining a deeper understanding of the fundamentals of computer science: data structures, algorithms, and programming paradigms.
- You want to become proficient in functional programming as well as object-oriented languages such as Java and C++ for general application development.





DS 2000: Programming with Data

What is DS 2000 about?

DS2000: Programming with Data is a first-course introduction to computer programming using the Python programming language. The course will introduce using various Python toolkits for manipulating and analyzing data in a broad range of applications including science, finance, and healthcare. The course assumes no prior programming experience.

Who should take DS 2000?

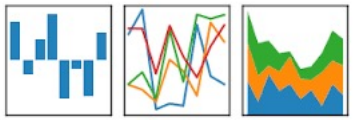
DS 2000 is aimed at students majoring in subjects outside of Computer Science or Data Science who wish to use computer programming as an essential *tool* for working with data in their respective fields. The course teaches students how to work with data algorithmically. *DS 2000 is the one computer programming course to take if you are only going to take one!*

Take DS 2000 if:	Alternatively, take CS 2500 if:
You are a DS major aiming to become a professional data scientists or you are a non-CS / non-DS major looking to gain some practical experience with computer programming as a tool for solving problems in various technical fields. Students with some prior programming skills should consider DS 2500 instead.	You are a CS major aiming to become a professional computer programmer or computer scientist. CS minors interested in the foundations of Computer Science and hoping to eventually tackle larger-scale software projects will also find CS 2500 to be a better fit.

python™

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

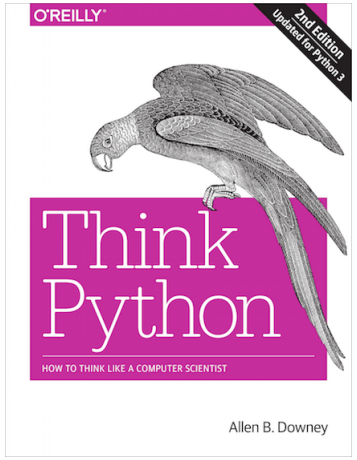


Northeastern University
Khoury College of Computer Sciences

Textbooks

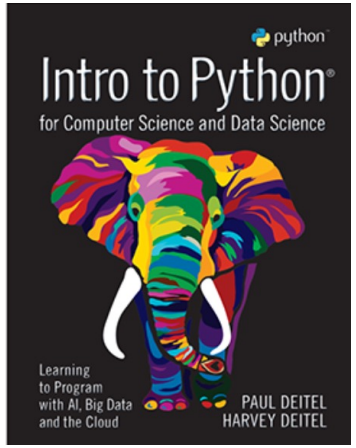
Recommended: Downey, 2016: *Think Python*, 2nd ed. O'Reilly

https://learning.oreilly.com/library/view/think-python-2nd/9781491939406/
https://greenteapress.com/thinkpython2/html/
https://www.amazon.com/Think-Python-Like-Computer-Scientist/dp/1491939362/



Recommended: Deitel and Deitel, 2019: Intro to Python for Computer Science and Data Science, 1st ed. Pearson

https://learning.oreilly.com/library/view/intro-to-python/9780135404799/
https://www.amazon.com/Intro-Python-Computer-Science-Data/dp/0135404673/



How to learn from the textbook: Practice and Experiment

GOOD

Read the book.

Read the book AND type in the examples.

BETTER

Tinker with the examples. Change them and observe the behavior. Try to break the examples and see what happens.

BEST

Write lots of programs while becoming practiced with editors, testing and debugging, and third-party libraries that can do powerful stuff for you.

Languages

Natural languages are languages that people speak: English, Spanish, French, etc.

Formal languages are languages that are *designed* for a specific purpose and having a well-defined syntax.

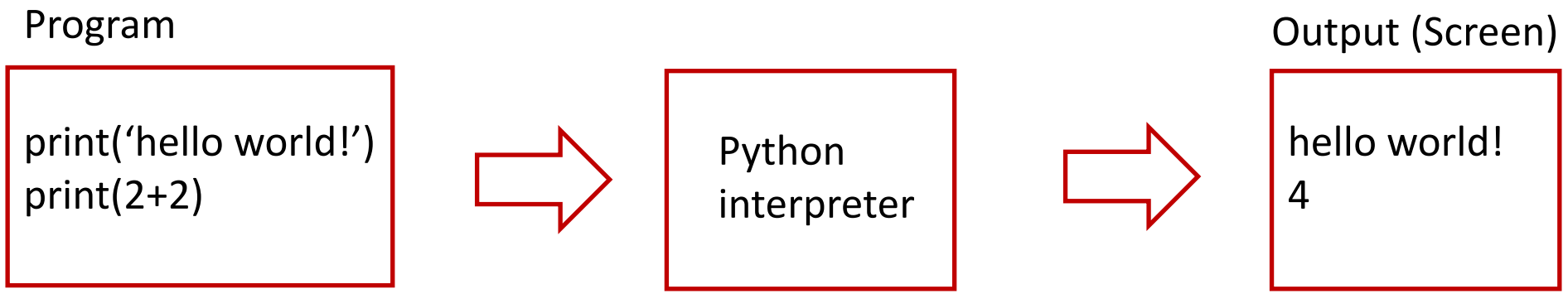
Mathematics: y = mx + b, 2 + 2 = 4, $\nabla \times \mathbf{B} = \frac{1}{c}\left(4\pi\mathbf{J} + \frac{\partial \mathbf{E}}{\partial t}\right)$

Programming languages are formal languages designed to express computations as a **program**.

What is a program?

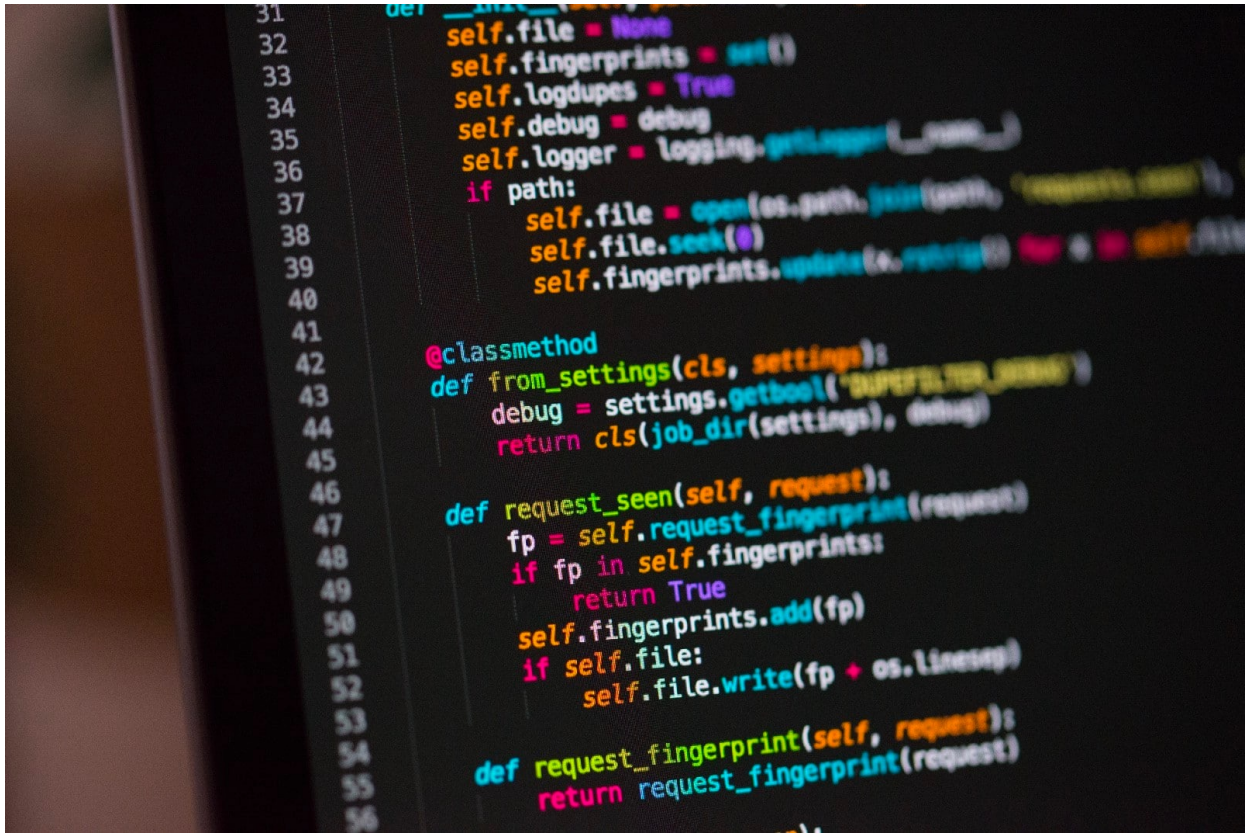
A **program** is an *ordered sequence* of instructions (rendered as one or more text files) that specifies how to perform a computation.

The **Python interpreter** parses a python program, validates the syntax, and runs the program on a computer.



What is programming?

Programming or **coding** is the *art* of writing programs designed to carry out the intended computations as concisely and efficiently as possible.



Programming languages are many and varied

LISP

```
(defun factorial (n)
  (if (= n 0)
      1
      (* n (factorial (- n 1))) ) )
```

Assembly

```
DATA SEGMENT
A DB 5
DATA ENDS
CODE SEGMENT
         ASSUME DS:DATA,CS:CODE
START:
      MOV AX,DATA
      MOV DS,AX
      MOV AH,00
      MOV AL,A
 L1:  DEC A
      MUL A
      MOV CL,A
      CMP CL,01
      JNZ L1
      MOV AH,4CH
      INT 21H
CODE ENDS
END START
```

Java

```
public static int factorial(int n) {
    if (n == 0) {
        return 1;
    } else {
        return n * factorial(n - 1);
    }
}
```

Prolog

```
factorial(0) => 1
factorial(n) => factorial(n-1) * n;
```

Python

```
def fact(n):
    if(n==0):
        return 1
    else:
        return n*fact(n-1)
```

High-Level vs. Low-Level Languages

High-Level Languages (Python, Java, C++, LISP …)

```
def fact(n):
    if(n==0):
        return 1
    else:
        return n*fact(n-1)
```

- Expressive
- Easier to learn
- Easier to read and maintain
- Slower execution
- Cross-platform (portable)

Low-Level (Machine language)

```
DATA SEGMENT
A DB 5
DATA ENDS
CODE SEGMENT
        ASSUME DS:DATA,CS:CODE
START:
      MOV AX,DATA
      MOV DS,AX
      MOV AH,00
      MOV AL,A
 L1:  DEC A
      MUL A
      MOV CL,A
      CMP CL,01
      JNZ L1
      MOV AH,4CH
      INT 21H
CODE ENDS
END START
```

- CPU-level instruction
- CPU-specific operations
- More difficult to understand
- Requires more lines of code
- *Super super fast*

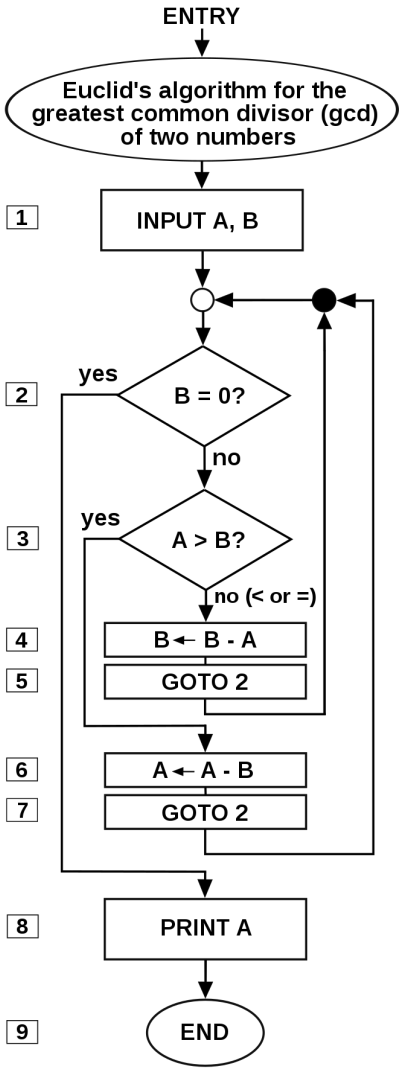
What is an algorithm?

An **algorithm** is a procedure, formula, or specification for solving a particular problem.

Algorithms are *implemented* by writing a program. The algorithm doesn't depend on the programming language, but the implementation does!

Examples:

- Sorting numbers: [3,1,6,4] → [1,3,4,6]
- Google search
- Encryption / Decryption



Advantages of Python

- A very popular language, widely used in industry, particularly for data science / machine learning applications.

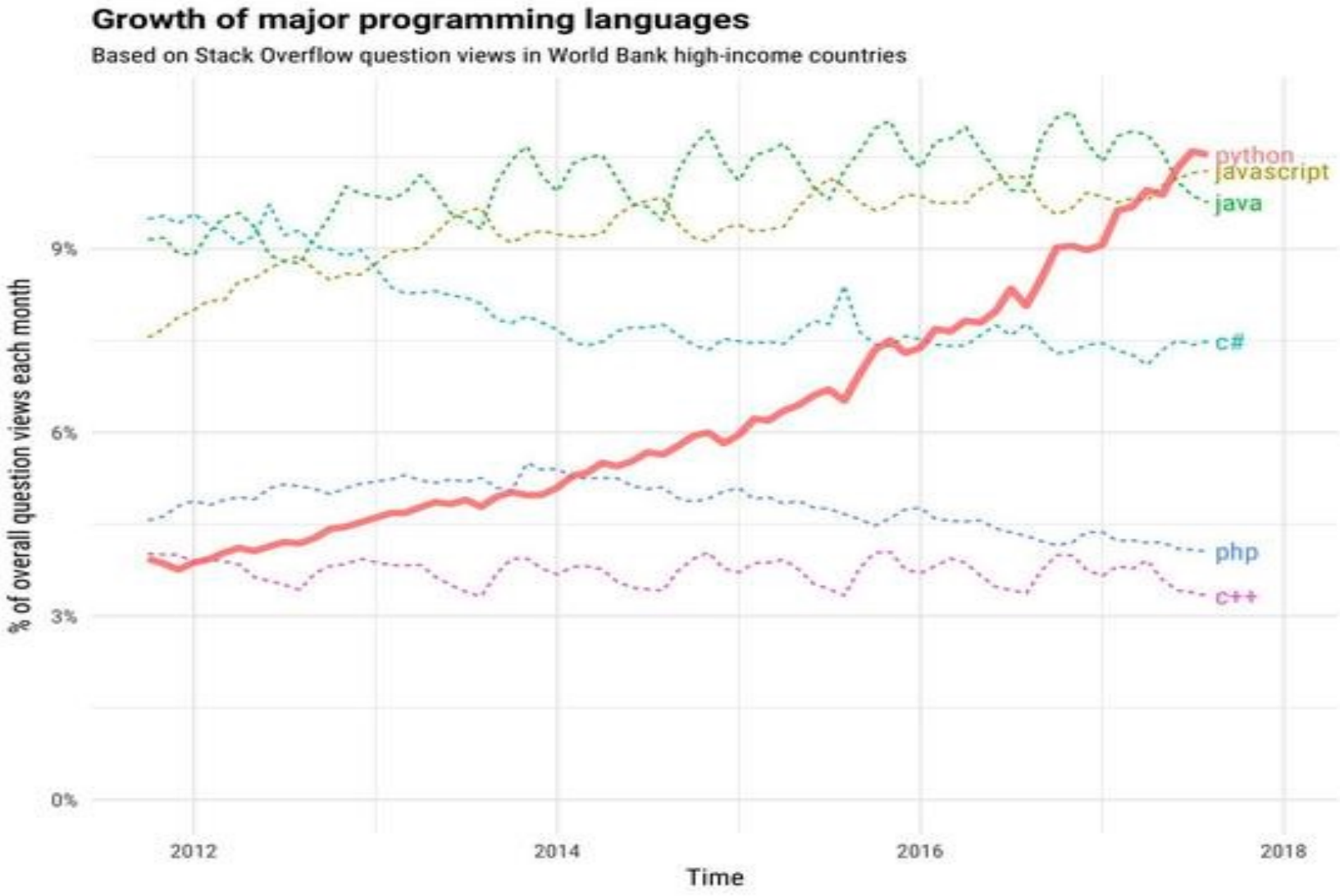
- A rich collection of re-usable libraries. (“*batteries included!*”)
  - Graphics and visualization
  - I/O (text, csv, json, database…)
  - AI / Machine learning / Data Science / Scientific computing
  - Web development frameworks
  - Graphics and User Interfaces
  - Etc.

- Easy to learn because of its expressive and clean syntax

The growing popularity of python

Source:

Nick Heath (2019): *Python is eating the world: How one developer's side project became the hottest programming language on the planet.* Tech Republic. (August 6, 2019)



Disadvantages of Python

- Slower performance because it is an *interpreted* language
- High memory consumption
- Not ideal for mobile development
- Runtime errors due to *dynamic typing* may make testing and debugging more challenging
- The use of whitespace / indentation to denote program *structure* is controversial

While loops

1. Determine whether the condition is true or false.

2. If false, exit the `while` statement and continue execution at the next statement.

3. If the condition is true, run the body and then go back to step 1.

```
def countdown(n):
    while n > 0:
        print(n)
        n = n - 1
    print('Blastoff!')
```

countdown(10) ➔

```
10
9
8
7
6
5
4
3
2
1
Blastoff!
```

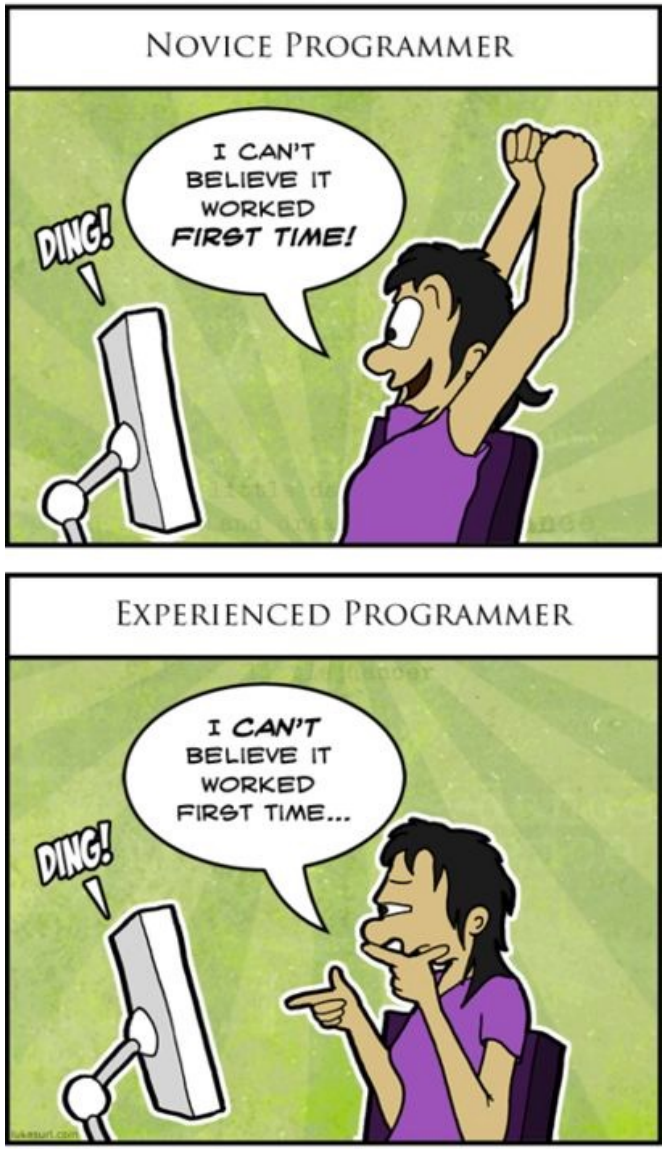
Debugging

Programming mistakes are called **bugs**.

Debugging is the process of eliminating bugs, i.e., figuring out why your program doesn't behave the way it is expected to.

With experience, programmers learn to:

- Write code with fewer bugs
- Track down existing bugs
- Systematically verify that code is bug-free



Types of programming errors: Can you identify?

Syntax Errors:

```
print(‘hello!)
Print(‘hello!’)
print(‘hello”)
```

Runtime Error:

```
x = 0
y = 12 / x
```

Semantic Error:

```
miles = kilometers * 1.609344
```

Python Keywords

```
False      class      finally    is         return
None       continue   for        lambda     try
True       def        from       nonlocal   while
and        del        global     not        with
as         elif       if         or         yield
assert     else       import     pass
break      except     in         raise
```

print("Hello, Students!")

urban DICTIONARY Browse Categories Vote Store

Type any word...

TOP DEFINITION

hello world

The easiest, and first program any **newbie** would write. Applies for any language. Also what you would see in the first **chapter** of most **programming** books.

programming noob: Hey I just attended my first programming lesson earlier!
*.NET **Veteran**: Oh? What can you do?*
*programming noob: I could make a **dialog** box pop up which says "**Hello World**!" !!!*
*.NET Veteran: lmao.. hey guys! look.. check out this "**hello world**" programmer*

Console.WriteLine("Hello World")

#programming #noobs #newbie #noob #hello #world

by The Black Jack October 26, 2006