
gnss
ACADEMY
ELEVATE YOUR STANDARDS
Python Guide

Python

- Python is an **interpreted language**, which can save you considerable time during program development because no compilation and linking is necessary. We will use **Anaconda** interpreter
- Python is simple to use, **available on Windows, Mac OS X, and Unix operating systems**, and will help you get the job done more quickly.
- Python is extensible: if you know how to program in C it is easy to add a new built-in function or module to the interpreter to perform critical operations at maximum speed
- All these features make Python an **excellent choice for a wide range of tools**, and **very appreciated on prototyping and testing tools of operational code**
- Important libraries (included in Anaconda): **numpy, matplotlib and pandas**

Python Basics

Etapios

gnss ACADEMY
ELEVATE YOUR STANDARDS

Python Official Documentation

https://docs.python.org/3.7/tutorial/index.html

- 1. Whetting Your Appetite
- 2. Using the Python Interpreter
  - 2.1. Invoking the Interpreter
    - 2.1.1. Argument Passing
    - 2.1.2. Interactive Mode
  - 2.2. The Interpreter and Its Environment
    - 2.2.1. Source Code Encoding
- 3. An Informal Introduction to Python
  - 3.1. Using Python as a Calculator
    - 3.1.1. Numbers
    - 3.1.2. Strings
    - 3.1.3. Lists
  - 3.2. First Steps Towards Programming
- 4. More Control Flow Tools
  - 4.1. `if` Statements
  - 4.2. `for` Statements
  - 4.3. The `range()` Function
  - 4.4. `break` and `continue` Statements, and `else` Clauses on Loops
  - 4.5. `pass` Statements
  - 4.6. Defining Functions
  - 4.7. More on Defining Functions
    - 4.7.1. Default Argument Values
    - 4.7.2. Keyword Arguments
    - 4.7.3. Arbitrary Argument Lists

Python Official Documentation

https://docs.python.org/3.7/tutorial/index.html

- 4.7.4. Unpacking Argument Lists
- 4.7.5. Lambda Expressions
- 4.7.6. Documentation Strings
- 4.7.7. Function Annotations
- 4.8. Intermezzo: Coding Style

5. Data Structures
- 5.1. More on Lists
  - 5.1.1. Using Lists as Stacks
  - 5.1.2. Using Lists as Queues
  - 5.1.3. List Comprehensions
  - 5.1.4. Nested List Comprehensions
- 5.2. The `del` statement
- 5.3. Tuples and Sequences
- 5.4. Sets
- 5.5. Dictionaries
- 5.6. Looping Techniques
- 5.7. More on Conditions
- 5.8. Comparing Sequences and Other Types

6. Modules
- 6.1. More on Modules
  - 6.1.1. Executing modules as scripts
  - 6.1.2. The Module Search Path
  - 6.1.3. "Compiled" Python files
- 6.2. Standard Modules

Python Official Documentation

https://docs.python.org/3.7/tutorial/index.html

- 6.3. The dir() Function
- 6.4. Packages
  - 6.4.1. Importing * From a Package
  - 6.4.2. Intra-package References
  - 6.4.3. Packages in Multiple Directories
- 7. Input and Output
  - 7.1. Fancier Output Formatting
    - 7.1.1. Formatted String Literals
    - 7.1.2. The String format() Method
    - 7.1.3. Manual String Formatting
    - 7.1.4. Old string formatting
  - 7.2. Reading and Writing Files
    - 7.2.1. Methods of File Objects
    - 7.2.2. Saving structured data with json
- 8. Errors and Exceptions
  - 8.1. Syntax Errors
  - 8.2. Exceptions
  - 8.3. Handling Exceptions
  - 8.4. Raising Exceptions
  - 8.5. User-defined Exceptions
  - 8.6. Defining Clean-up Actions
  - 8.7. Predefined Clean-up Actions
- 9. Classes
  - 9.1. A Word About Names and Objects

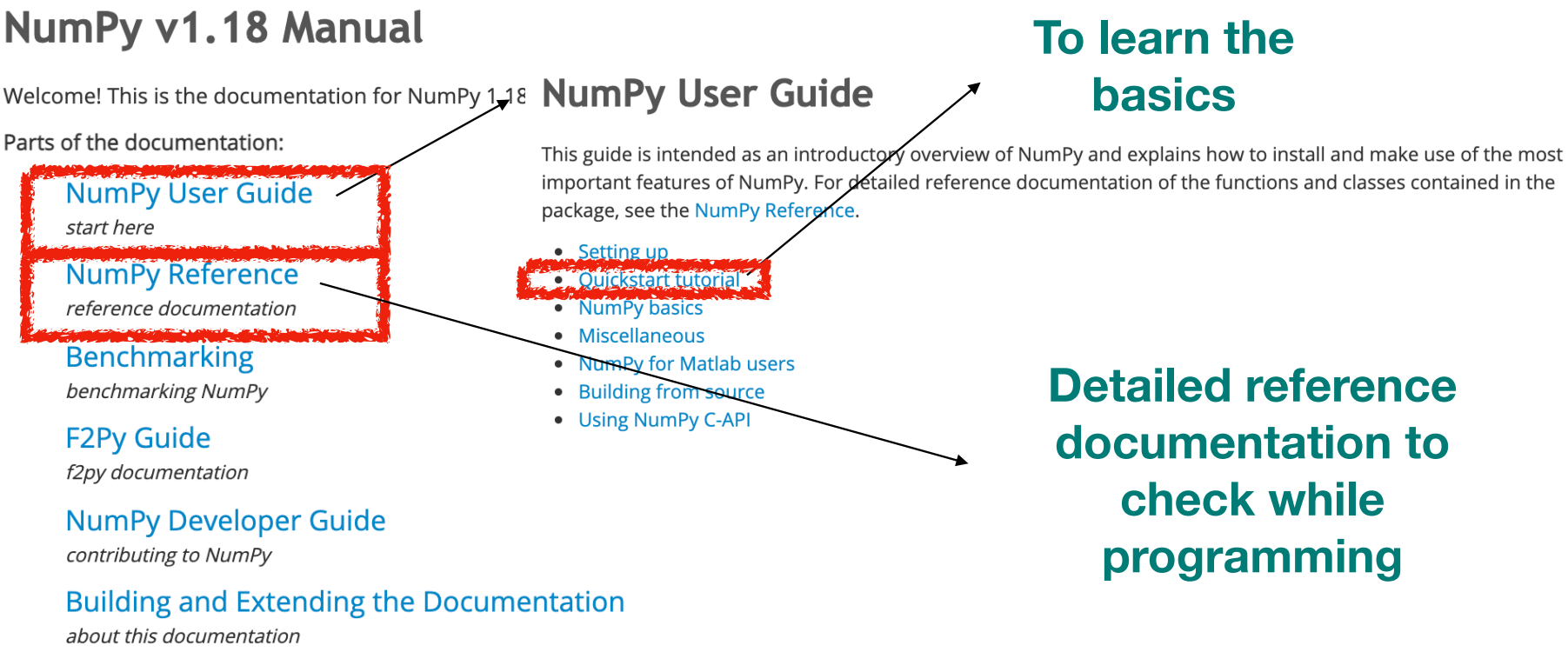
numpy Basics

Etapios

gnss ACADEMY
ELEVATE YOUR STANDARDS

numpy Manual

https://numpy.org/doc/1.18/index.html



To learn the basics

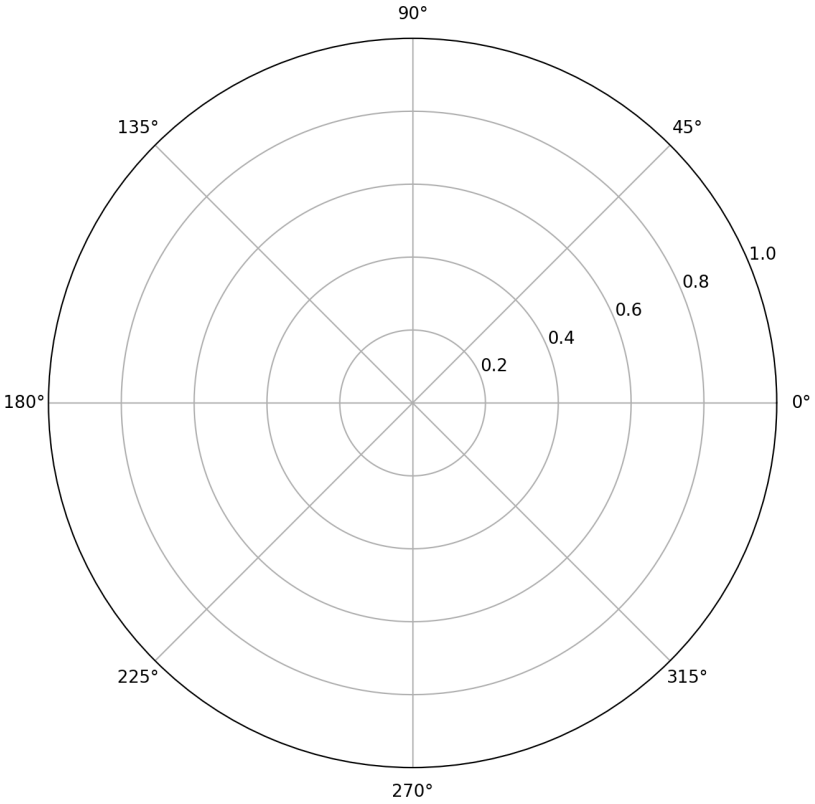
Detailed reference documentation to check while programming

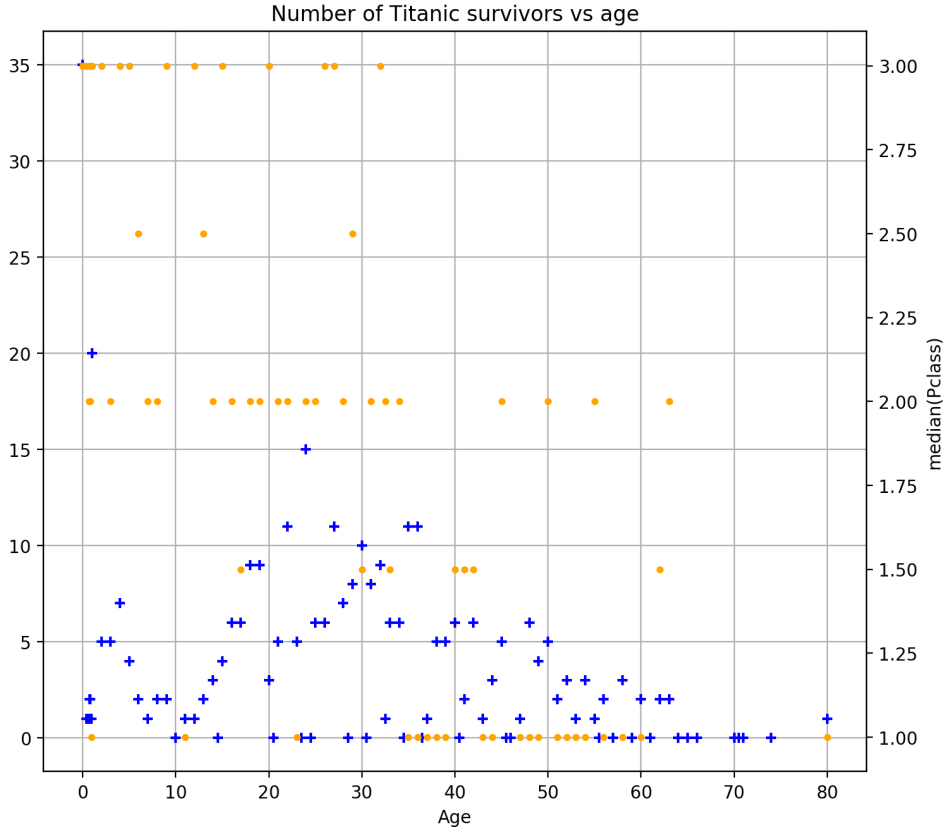
matplotlib Basics

Etapios

gnss ACADEMY
ELEVATE YOUR STANDARDS

matplotlib


90°
135°
45°
180°
0°
225°
315°
270°
0.2
0.4
0.6
0.8
1.0


Number of Titanic survivors vs age
Age
median(Pclass)
0
5
10
15
20
25
30
35
0
10
20
30
40
50
60
70
80
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00

matplotlib Reference

https://matplotlib.org/stable/tutorials/index.html

matplotlib Version 3.4.2

Installation | Documentation | Examples | Tutorials | Contributing

home | contents » User's Guide » Tutorials

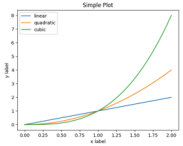
Tutorials

This page contains more in-depth guides for using Matplotlib. It is broken up into beginner, intermediate, and advanced sections, as well as sections covering specific topics.

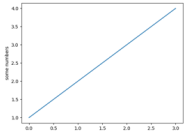
For shorter examples, see our examples page. You can also find external resources and a FAQ in our user guide.

Introductory

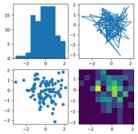
These tutorials cover the basics of creating visualizations with Matplotlib, as well as some best-practices in using the package effectively.



Usage Guide



Pyplot tutorial



Sample plots in

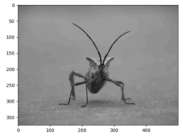


Image tutorial


End of Section