

## INTRODUCING PROGRAMMING TO OCEAN MAPPING STUDENTS

By Giuseppe Masetti<sup>1,3</sup>, Semme Dijkstra<sup>1</sup>, Rochelle Wigley<sup>1</sup> and Tyanne Faulkes<sup>2</sup>

<sup>1</sup> Center for Coastal and Ocean Mapping and NOAA-UNH Joint Hydrographic Center, University of New Hampshire, Durham, NH 03824, USA

<sup>2</sup> NOAA National Ocean Service, Office of Coast Survey, Hydrographic Survey Division, Pacific Hydrographic Branch, Seattle, WA 98115-6349, USA

<sup>3</sup> Danish Geodata Agency, Danish Hydrographic Office, Nørresundby 9400, Denmark



### Abstract

The next generation of ocean mappers need to master programming skills to meet increasingly higher expectations for timely ping-to-public data workflows. As such, the e-learning Python for Ocean Mapping (ePOM) project was established at the Center for Coastal and Ocean Mapping/NOAA-UNH Joint Hydrographic Center (University of New Hampshire). The project aims to support new ocean mapping students and professionals in reaching a minimum level of programming skills. These skills are then expanded with further powerful capabilities by leveraging the open-source Python scientific stack and the NOAA (National Oceanic & Atmospheric Administration) Office of Coast Survey's Pydro distribution. To the best of our knowledge, the ePOM project represents the first attempt at creating a set of comprehensive open-source courses providing students with the required initial coding skills for a career in the ocean mapping field.

**Keywords:** Ocean mapping, programming, scientific computing, education, ocean data science



### Résumé

La prochaine génération de spécialistes en cartographie océanique doit maîtriser les compétences en programmation pour répondre aux attentes de plus en plus élevées en matière de flux de travail de données ping-to-public en temps opportun. En tant que tel, le projet d'apprentissage en ligne Python for Ocean Mapping (ePOM) a été créé au *Center for Coastal and Ocean Mapping/NOAA-UNH Joint Hydrographic Center* (Université du New Hampshire). Le projet vise à aider les nouveaux étudiants et professionnels de la cartographie océanique à atteindre un niveau minimum de compétences en programmation. Ces compétences sont ensuite étendues avec d'autres capacités puissantes en tirant parti du stack scientifique de Python et de la distribution *Pydro* du Service des levés côtiers de la NOAA (*National Oceanic & Atmospheric Administration*). À notre connaissance, le projet ePOM représente la première tentative de création d'un ensemble de cours open-source complets offrant aux étudiants les compétences de codage initiales requises pour une carrière dans le domaine de la cartographie océanique.

**Mots-clés:** Cartographie océanique, programmation, calcul scientifique, éducation, science des données océaniques



## Resumen

La próxima generación de cartógrafos oceánicos necesitará dominar las habilidades de programación para cumplir las expectativas crecientes de flujos de datos rápidos de-ping-al-público. Por eso se creó el proyecto de formación a distancia en Python para Cartografía Oceánica (ePOM) en el Centro de Cartografía Costera y Oceánica / Centro Hidrográfico Conjunto NOAA-UNH (Universidad de New Hampshire). El objetivo del proyecto es ayudar a los nuevos estudiantes y profesionales de cartografía oceánica a alcanzar un nivel mínimo de habilidades de programación. Después se amplían estas habilidades con capacidades más potentes aprovechando el stack científico Python de código abierto y la distribución de Pydro de la Oficina de Levantamientos Costeros de NOAA (Administración Nacional del Océano y la Atmósfera). Hasta donde sabemos, el proyecto ePOM representa el primer intento de crear un conjunto de cursos completos de código abierto que proporcionan a los estudiantes las habilidades de codificación inicial necesarias para hacer carrera en el campo de la cartografía oceánica.

**Palabras clave:** Cartografía oceánica, programación, informática científica, educación, ciencia de datos oceánicos

## 1. INTRODUCTION

Computing has revolutionized research over the past several decades. Contemporary studies in earth science, oceanography, ocean engineering and other marine sciences now rely on computer programs. Computation is essential for data acquisition, analysis, automation, and hypothesis testing via modeling and simulation. As such, the ability to read, understand and write computer programs is now part of the list of indispensable skills that a modern researcher needs to have.

The role of programming in ocean mapping is not limited to research. Programming is already an important part of STEM (Science, Technology, Engineering, and Mathematics) education, with several curricula incorporating it as a core competence. Specifically, ocean mappers are increasingly expected to have a basic understanding of the computational tools that they are using to collect and process data daily (Masetti et al., 2020). Following this trend, the next generation of ocean mappers will need to have, and constantly expand, a set of solid programming skills to face the variety of new sensors and data types, and complexity of newly collected data (Mayer and Roach, 2021). At the same time, based on current trends in the technological evolution of ocean mapping, datasets of unprecedented size are becoming the norm (Hughes Clarke, 2018; Le Deunf et al., 2020). This situation will be exacerbated by even higher expectations for the timely transposition of data from “ping” (data acquisition) to “public” (e.g., updates to nautical charts, pipeline inspection, evaluation of areas for windmill farms).

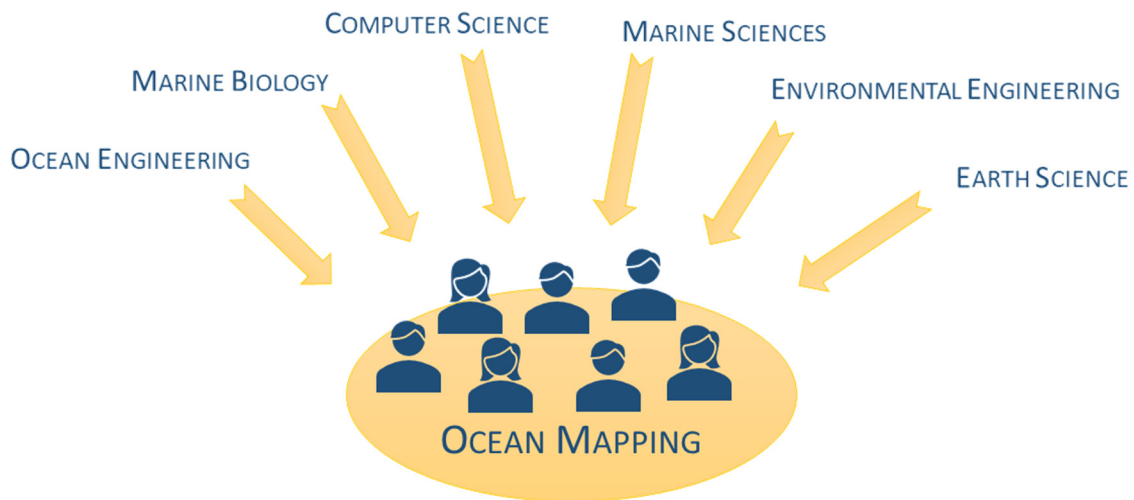
To address these challenges, a new kind of ocean mapping training is required to educate specialists who can collect, structure, process, and analyze (both numerically and visually) large datasets (Wigley et al., 2017). Specialized processing software is available for these datasets in the form of small tools and/or large software suites. However, due to the multitude of ways in which ocean mapping sensors are integrated, it is unrealistic to assume that these software applications will solve the variety of challenges that are present in ocean mapping. The ability to program can provide the ocean mapper with a more efficient mechanism to use available tools in new ways, allowing control for every single step of data analysis (Masetti et al., 2019). Software implementation of algorithms – paired with standards-compliant data-exchange formats – may represent a universal and portable solution for a specific task.

Since newcomers entering the ocean mapping field often have limited programming experience and come from diverse backgrounds (**Figure 1**), the need arises to help them in reaching a minimum level of programming skills to successfully embark on such a highly specialized career. During their undergraduate studies, many students often did not take more than a basic programming class since such courses are seen as challenging not only for students to take, but for staff to deliver (Reades, 2020). However, the basics of the needed skills can be learned quickly by providing a basic knowledge of a programming language and the comprehension of a few core notions like, for instance, data structures. Once students obtain these essential programming skills, they may expand them with more advanced topics (e.g., numerical methods) on an as-needed basis. These programming skills are useful not only in an academic setting, but as a powerful tool in their professional career.

Although many programming courses are available, with a subset of them having a geospatial focus like (Anbaroğlu, 2021) and (Petrelli, 2021), none of them were judged to fulfill all the requirements of:

- Designed for being remotely accessible and executed in a cloud solution.
- Providing the coding and geospatial notions at the necessary pace for initially being taken asynchronously.
- Uniformly covering the necessary coding skills for preparing the student to face their subsequent ocean mapping courses.

The option of combining two or more existing courses was rejected due to the lack of consistency and cohesion that such an approach would have introduced (Bowlick, Bednarz and Goldberg, 2020).



**Figure 1.** Incoming students in ocean mapping come from diverse backgrounds: ocean engineering, computer science, marine biology, earth science, etc.

Based on these considerations, the e-learning Python for Ocean Mapping (ePOM) project was designed and implemented at the Center for Coastal and Ocean Mapping/NOAA-UNH Joint Hydrographic Center (CCOM/JHC) at the University of New Hampshire. This document explains the rationale of selecting Python as the primary programming language at CCOM/JHC, illustrates the programming courses and the platform used to offer them, and describes the improvements based on student and instructor feedback. Finally, an evaluation of the results achieved with the ePOM project is presented together with potential future extensions to the current courses.

## 2. PYTHON AS FIRST PROGRAMMING LANGUAGE

There are a lot of programming languages to choose from, each with their own strengths and weaknesses depending on their use cases (Bogdanchikov et al., 2013). Although, in principle, any programming language can be taught as a first language, there are good reasons for careful evaluation/comparison before selecting one.

The change in thinking that occurs when students learn programming is their ability to view problems in an algorithmic way. When working on such a shift, the student should ideally focus on the core concepts of programming without being overwhelmed by a language's specific nuances. From this point of view, Python represents a natural choice for algorithm realization. When compared to other popular languages, like C++ or Java, it has less terms and elements related to the specifics of the language. For instance, the block of statements in the body of a for-loop is limited solely by indentation (Fangohr, 2004). This makes Python easy to learn, but also easy to read. This latter becomes a critical feature when writing complex programs.

These considerations are confirmed by the observed trend at dozens of universities around the world where the Python programming language is gaining a prominent role for teaching introductory programming courses to students of computer science departments (Fernandes et al., 2018). In the search for a good programming language for data science applications, Python has clearly emerged as a comprehensive solution (Nagpal and Gabrani, 2019). Leveraging such characteristics, Python has steadily climbed the list of the most popular programming languages (Cass, 2022).

Python has powerful features that reflect the way humans think, and it approaches plain English. It is a 'high level' programming language, this in contrast to low level languages that more closely resemble machine language. Python is expressive for object-oriented programming as well as other modern paradigms. These characteristics, coupled with the considerable number of availa-

ble standard and third-party libraries, make it particularly simple to prototype an algorithm in Python. If needed, a fully or partial rewrite of the code may be done in a lower-level language (e.g., C/C++ or Java) once the prototyping stage is completed.

Python is among the programming languages with a higher level of abstraction from the computer's details. In fact, Python avoids the programmer having to deal with memory management, variable declarations, etc. that are common in other programming languages. At the same time, Python is a very dynamic language: e.g., class attributes can be created on the fly (this cannot be done in C/C++ and JAVA). Python also minimizes the number of extra keywords (e.g., security keywords like *public*, *private* and *protected*) required to write syntactically correct programs. In addition, polymorphism and operator overloading gives extra power to Python classes without the limitations of other languages (e.g., C++'s virtual functions, JAVA's restricted syntax for operators). Python is clearer and has greater functionality than MATLAB, another language popular for prototyping code in academic settings (Fangohr, 2004; Ozgur et al., 2022).

Python allows the user to begin programming (i.e., the Hello World program) without several requirements (e.g., importing dependencies, declaring functions or methods) that other programming languages have. Code indentation also plays a significant role in writing programs that are easily read and understood. Additionally, there are many freely available and easy-to-use Integrated Development Environment (IDE) software suites available (e.g., Spyder, PyCharm) to ease coding tasks.

The programming skills acquired in Python may be transferred to learn other programming languages (Shrestha, Barik and Parnin, 2018). Being an interpreted language, it is easy to understand how code works by interacting directly with the Python command-line interpreter. The built-in help module contributes to facilitate the understanding of specific functions and classes.

There are also a few limitations in using Python that should be mentioned. The execution time of an algorithm written in Python is usually slower than in a compiled language like C or Fortran (Bassi, 2007). There are tools (e.g., Numba, Cython) that may significantly reduce the execution time by compiling the critical section of the Python code into efficient machine code, comparable in performance to a traditional compiled language (Behnel et al., 2011; Lam et al., 2015). Another major drawback is related to the number of numerical and statistical functions available in Python when compared to specialized languages (like R or MATLAB) (Bassi, 2007). However, the Python community is constantly working to reduce this gap by improving popular packages like NumPy and SciPy (Harris et al., 2020; Virtanen et al., 2020). The two mentioned limitations and others (e.g., low usage in mobile applications, high memory consumption) will be reduced over time, due to the continuous efforts to improve the language in itself and extend the community-developed packages (Nagpal and Gabrani, 2019; Stančin and Jović, 2019).

As may be summarized from the above, Python represents a simple, compact, and popular language, which makes it particularly attractive for education in general. There are then additional motivations that make Python the natural choice when it comes to providing programming skills to ocean mapping students and professionals alike. Python is quite popular for academic work in ocean mapping, and it is publicly available at no cost to ocean mapping agencies around the world. At the same time, Python is a popular solution among embedded scripting languages for GIS (Geographic Information System) applications (e.g., both QGIS and ESRI ArcMap support Python scripting), with a large community of experts welcoming new users.

Furthermore, the NOAA Office of Coast Survey (OCS) has made Pydro publicly available (Riley et al., 2001). Pydro is a Python distribution and a suite of several libraries and applications developed to support effective OCS hydrographic workflows. Although primarily driven by OCS specifications and requirements, the Python-based tools in Pydro can easily be extended/adapted to general good practices in the ocean mapping field (Masetti et al., 2022).

### 3. DESIGN AND IMPLEMENTATION OF THE PROGRAMMING TRAINING

In 2018, CCOM/JHC decided to include Python-based programming courses into the ocean mapping curriculum. This decision triggered the creation of the e-learning Python for Ocean Mapping (ePOM) project in January 2019 (<https://www.hydrooffice.org/epom>). The project name is based on the facts that:

- A large part of the training is done remotely (e-Learning).
- Python is the programming language used.
- The introduced concepts, listed in **Table 1** and **Table 2**, are tailored to ocean mapping students, offering examples and exercises with practical applications in the ocean mapping field.

#### 3.1 Teaching goals

From the beginning, the target has not been to put together a full course on how to program in Python. Instead, the teaching goals have been to familiarize ocean mapping students and professionals with key programming concepts (see **Table 1** and **Table 2**) and, at the same time, to teach them programming habits and skills that can be easily translated to other programming languages (i.e., learning transfer).

Given the popularity of Python, there is an abundant amount of information about the language available online. This can be overwhelming. The ePOM project aims to help navigate through the vast number of online resources. Last, but not least, ePOM aims to familiarize users with the popular open-source Python scientific stack and the NOAA OCS' Pydro distribution.

#### 3.2 Adopted technologies

The project was developed using Jupyter notebooks. This allows for the creation of computational documents i.e., documents are textbook like, but contain live code and visualizations (**Figure 2**).

To facilitate the learning, key concepts in the notebooks are visually marked (**Figure 3**, pane A). Similarly, the notebooks clearly identify supplementary information that may be skipped at a first reading (**Figure 3**, pane B). Finally, many notebooks contain one or more exercises, with solutions accessible by clicking on the related button (**Figure 3**, pane C).

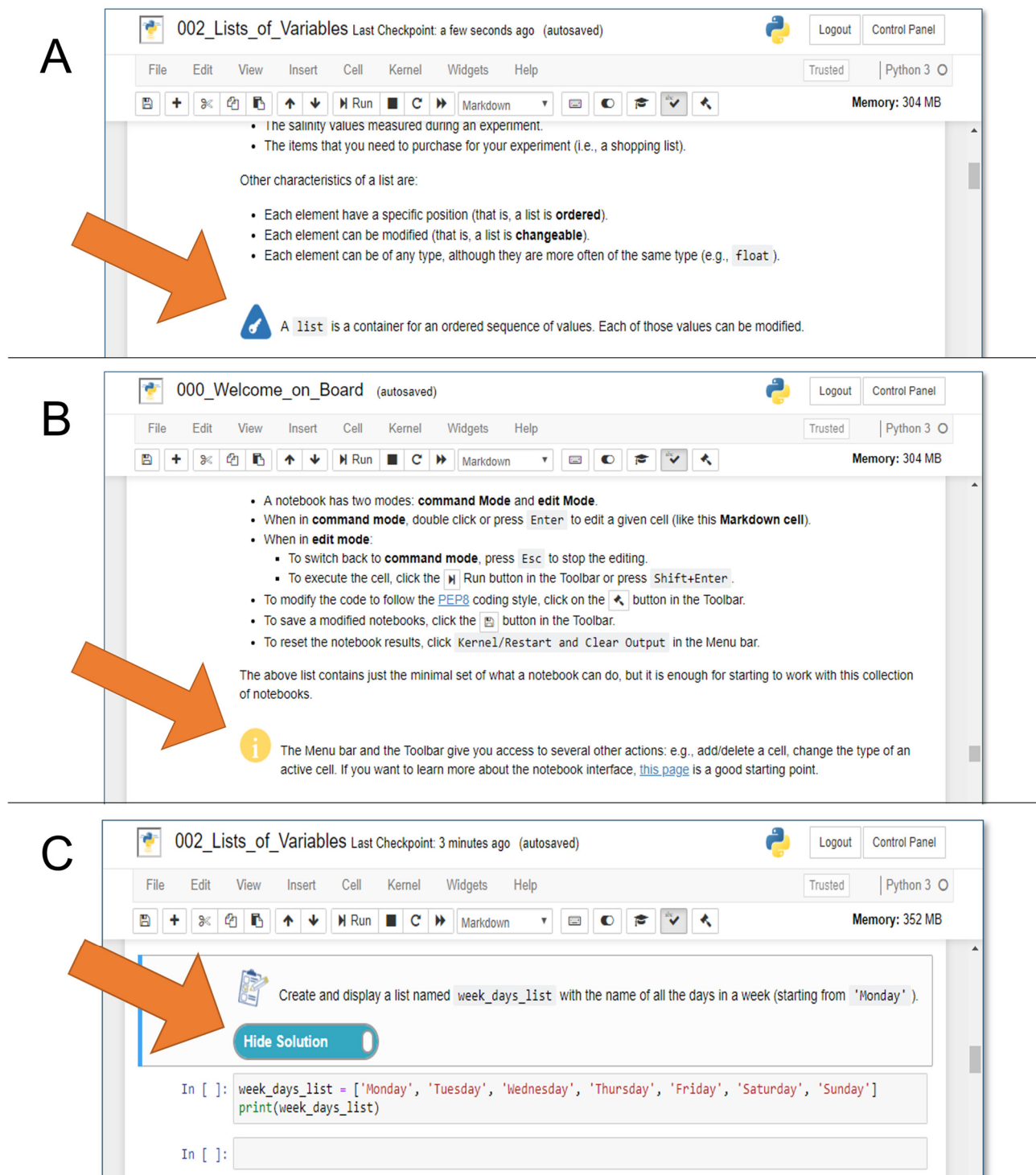
Blended learning has generated an increasing interest in educational research. This interest was strongly increased due to the COVID-19 pandemic (Tadlaoui and Chekou, 2021). As such, the courses have been tailored to be fully attended remotely and asynchronously (Zinovieva et al., 2021). The platform selected to provide students with access to the Jupyter notebooks is JupyterHub, a Python environment that runs on a multi-user server. With JupyterHub, all the students get an identical computing environment. JupyterHub has a significant user community and has been adopted for teaching by several universities (Elhayany et al., 2022).

#### 3.3 Notebooks for learning programming

Two main sets of notebooks are provided:

- *Programming Basics with Python*, with a focus on basic programming concepts with ocean mapping applications; and
- *Introduction to Ocean Data Science*, providing elements of object-oriented programming, data visualization, scientific computing, and geospatial data formats.

During the summer of 2019, both sets of notebooks went through an incubation phase and received feedback and comments from several volunteer beta testers.



**Figure 2.** Examples of notebook's feature: key concepts (pane A), supplementary information (pane B), and coding exercises with accessible solutions (pane C). The adoptions of specific icons for these features have the intent to facilitate the learning process as well as the later review of critical elements in each notebook.

A key feature of this solution is that there is no need to install Python (and other third-party libraries) on the student's computer. The students will only need a modern browser and an Internet connection. Once logged in, the landing page provides a list of notebooks with a brief description of the teaching objectives (**Figure 2**). The student then can click on the notebook of choice.

**List of Notebooks**

Notebook Name	Topics
<a href="#">Welcome on Board</a>	Python: pros and cons. Notebook interaction. <code>print()</code>
<a href="#">Variable and Types</a>	Variables, <code>int</code> , <code>float</code> , <code>str</code> , <code>type()</code> , Operators and type casting.
<a href="#">Lists of Variables</a>	
<a href="#">Conditional Execution</a>	
<a href="#">Loops</a>	
<a href="#">Write Your Own Functions</a>	
<a href="#">Dictionaries</a>	
<a href="#">Read and Write Text Files</a>	
<a href="#">First Steps of a Class</a>	
<a href="#">More About Classes</a>	
<a href="#">Wrapping Up Notions</a>	

**Welcome on Board!**

If you are reading this notebook, you are likely to be a student at the [Center for Coastal and Ocean Mapping / NOAA-JNH Joint Hydrographic Center \(CCOM/JHC\)](#). Welcome!

As a student, you will have assignments for the courses that you will be taking. These assignments can come in the form of practical laboratory exercises, presentations, essays, and many other forms. A large number of exercises have a significant 'coding' component, meaning that you are expected to write some short computer programs (code) to achieve certain ocean-mapping related tasks.

We do not assume that you are familiar with computer programming, and we have therefore created this short training on programming basics. Through a set of notebooks, we will provide you with the basic coding knowledge required for the successful completion of your first assignments. The notebooks use Python, the preferred programming language at CCOM/JHC. This document is an example of a Python notebook and is the first of the series in **Programming Basics with Python for Ocean Mapping**.

The overall task is to lead you through some basic concepts of programming using the Python language, with a focus on their application to the Ocean Mapping field.

**Figure 3.** Example of student experience through the CCOM/JHC's JupyterHub server. Once logged (left pane), the student is presented with a list of notebooks (middle pane) from which to access specific topics (right pane). The notebooks are interconnected so that topics already treated in the past can be easily accessed for review.

### 3.3.1 Programming Basics with Python

As with human languages, programming fluency can only be developed actively, not passively. Based on this consideration, the course starts with basic concepts, such as “variable” and “type”, then the notebooks methodically advance the student to the point of writing a class to read and manipulate files with geospatial data.

The course has 2 phases:

- The first phase that is asynchronous providing online learning through a remotely hosted collection of Jupyter notebooks (**Table 1**).
- The second phase consists of an in-person meeting (during the student orientation week). This meeting is devoted to answer students' questions, to evaluate their understanding of the main concepts, and to stimulate collaboration among students.

The first phase primer offers a basic introduction to coding via Python with a focus on ocean mapping applications. It includes concrete examples and exercises to illustrate Python's usage and capabilities, culminating with a final notebook that connects several concepts from the past notebooks (see **Table 1**).

### 3.3.2 Introduction to Ocean Data Science

The course is intended to act as a connector to the ocean mapping classes offered to graduate students at CCOM/JHC. It consists of two modules that are part of the Applied Tools for Ocean Mapping class:

- The first module is taught during the first 2 weeks of the course. It has a focus on data analysis and visualization using popular Python libraries (**Table 2**).
- The second module introduces the student to popular tools (like git) and approaches on how to develop research code. The module is taught at the end of the Fall Term as it allows the students time to develop their coding skills allowing more complex skills to be taught.



**Table 1.** The “Programming Basics with Python” notebooks with related topics.

Notebook Name	Topics
Welcome on Board	Python: pros and cons. Notebook interaction. print()
Variable and Types	Variables. int, float, str. type(). Operators and type casting.
Lists of Variables	List. dir(). len(). Methods.
Conditional Execution	bool. #. if, elif, and else. len(). and, or, not.
Loops	for loop. while loop. List pop().
Write Your Own Functions	def. Parameters. return. None.
Read and Write Text Files	os module. raise. open()
Dictionaries and Metadata	dict, OrderedDict. Metadata.
A Class as a Data Container	class and __init__().
Summing-Up	Review. str.split(). range().

**Table 2.** The “Introduction to Ocean Data Science” notebooks with related topics.

Section	Notebook Name	Topics
Object-Oriented Programming	First Steps of a Class	class, __init__, self, __str__, class attributes
	More About Classes	class interface, how to add a method, datetime.fromtimestamp
Data Visualization	Intro to Matplotlib	matplotlib.plot(), cartopy
	Adding a Plotting Method	how to add a plot() method
Scientific Computing	Intro to NumPy	numpy.ndarray, Matplotlib support, indexing, slicing
	Adopting NumPy in Class Methods	numpy integration in a class
	NumPy Array Operations	view vs. copy, reshaping, split, concatenate, dot product
Data Formats	Intro to GDAL	raster and vector data formats
Supplementary	The Zen of Python	guiding principles for Python's design

### 3.4. Beta Testing and First Cycle

Before adopting the notebooks in actual teaching, a group of 10 volunteers (both students and researchers at the CCOM/JHC) acted as beta testers in the spring 2019, with coding skills spanning from beginner level to proficient level. They were asked to provide granular feedback on the quick start guide and the notebooks, but also a general evaluation of the entire course. Specifically, questions were asked about the general evaluation of the whole notebook, the relevance of each notebook section, the estimated time of completion for a student with no programming background, and suggested changes. Several adjustments (e.g., reductions of verbose explanation, additions of more exercises) have been adopted based on the testers’ feedback. More details are available in (Masetti et al., 2020).

Following the beta testing phase, student evaluations have been collected after the first teaching cycle that occurred at UNH during Fall Term 2019. Although there were diverse levels of programming skills among the students (spanning from no experience to intermediate level), all of them have provided quite positive reactions (based on the feedback collected by the instructors) for both clarity of the content and its accessibility. The first cycle of ePOM courses – as well as

the results of the beta testing phase – have been presented at the Canadian Hydrographic Conference 2020 (Masetti et al., 2020). This presentation has triggered the interest of the ocean mapping community, with many persons (working for both public agencies and in the private sector) asking to evaluate the ePOM notebooks.

#### **4. CONCLUSION AND FUTURE WORKS**

The ePOM courses aim to provide ocean mapping students with the programming and confidence to optimize solutions for specific issues through coding. As such, it is critical to apply an adequate teaching strategy like the one adopted in the ePOM project. The positive reactions collected during the first few years of the ePOM project, combined with an initial increase in quality of the work produced by enrolled students as observed by the CCOM faculty, supports the validity of the training solution currently adopted. However, a more comprehensive evaluation of the impact of the ePOM courses will only be possible after the conclusion of a few graduation cycles.

A key aspect of the project has been the decision of using a cloud solution, with Jupyter notebooks running on a pre-configured Python environment. This solution has the advantage for the students to focus from the very first moment on learning coding rather than to struggle with setting up a functional Python environment (and all the required packages) for the specific tasks in the assignments. However, this may also represent a disadvantage in case that the student will later face the need of developing Python code outside from a pre-configured environment (e.g., Pydro) or a cloud solution (like the ePOM server). Early attempts providing the students with generic guidance on this latter task has shown that this is not a trivial problem. Future work should focus on creating a set of optional lectures to be activated only on an as-needed basis.

Many of the courses taught through CCOM/JHC in the latest academic years now include labs provided to the students in the same format as used by ePOM. These courses include ESCI 872 Tools for Ocean Mapping, OE/ESCI 874 Integrated Seabed mapping Systems, OE/ESCI 771/871 Geodesy and Positioning for Ocean Mapping and, OE 875 Advanced topics in Ocean Mapping. Further modules may be developed in the context of the growing interest for programming in related subjects such as linear algebra, artificial intelligence, data mining and autonomous surveying.

The ePOM project has obtained the attention of agencies, companies, and individuals around the world, highlighting the potential impact of such an initiative for the ocean mapping field. The design characteristics of being open source may represent a crucial factor for having the ePOM notebooks being replicated, extended, and improved by other ocean mapping programs. To the best of our knowledge, the ePOM project represents the first attempt for a set of comprehensive open-source courses providing students with the required initial coding skills for anyone operating in the ocean mapping field.

#### **5. SUPPLEMENTARY MATERIALS**

The source code – with example scripts and data samples – are publicly available at: <https://github.com/hydrooffice>. Future updates on the ePOM initiative can be retrieved at: <https://www.hydrooffice.org/epom>.

#### **6. ACKNOWLEDGMENTS**

This research was partially funded by NOAA, grant numbers NA15NOS4000200 and NA20NOS4000196.

Many beta testers from all around the world contributed to this work by providing feedback and endless sources of inspiration. In particular, the authors would like to thank the NOAA Office of Coast Survey and the University of New Hampshire's CCOM/JHC for actively supporting the de-

velopment of the programming course. Finally, we also acknowledge the Office of Coast Survey's Hydrographic Systems and Technology Branch for the continuous help in the integration within the Pydro software distribution.

## 7. REFERENCES

- Anbaroğlu, B. (2021). A collaborative GIS programming course using GitHub Classroom. *Transactions in GIS*, 25(6), pp. 3132-3158, doi:10.1111/tgis.12810.
- Bassi, S. (2007). A Primer on Python for Life Science Researchers. *PLOS Computational Biology*, 3(11), pp. 2052-2057, doi:10.1371/journal.pcbi.0030199.
- Behnel, S., et al. (2011). Cython: The Best of Both Worlds. *Computing in Science & Engineering*, 13(2), pp. 31-39, doi:10.1109/MCSE.2010.118.
- Bogdanchikov, A., Zhaparov, M. and Suliyev, R. (2013). Python to learn programming. *Journal of Physics: Conference Series*, 423, pp. 1-5, doi:10.1088/1742-6596/423/1/012027.
- Bowlick, F. J., Bednarz, S.W. and Goldberg, D. W. (2020). Course syllabi in GIS programming: Trends and patterns in the integration of computer science and programming. *The Canadian Geographer / Le Géographe canadien*, 64(4), pp. 495-511, doi:10.1111/cag.12544.
- Cass, S. (2022). Top Programming Languages. *IEEE Spectrum*, viewed 17 September 2022, <https://spectrum.ieee.org/top-programming-languages-2022>.
- Elhayany, M., Nair, R.-R., Staubitz, T. and Meinel, C. (2022), A Study about Future Prospects of JupyterHub in MOOCs. *Proceedings of the Ninth ACM Conference on Learning @ Scale*, New York City, NY, USA, Association for Computing Machinery, pp. 275–279, doi:10.1145/3491140.3529537.
- Fangohr, H. (2004), A Comparison of C, MATLAB, and Python as Teaching Languages in Engineering. *Computational Science - ICCS 2004*, Berlin, Heidelberg, Springer Berlin Heidelberg, pp. 1210-1217, doi:10.1007/978-3-540-25944-2\_157.
- Fernandes, T. R., et al. (2018), Python Programming Language for Power System Analysis Education and Research. *IEEE PES Transmission & Distribution Conference and Exhibition - Latin America (T&D-LA)*, pp. 1-5, doi:10.1109/TDC-LA.2018.8511780.
- Harris, C. R., et al. (2020). Array programming with NumPy. *Nature*, 585(7825), pp. 357-362, doi:10.1038/s41586-020-2649-2.
- Hughes Clarke, J. E. (2018). *Multibeam Echosounders, Submarine Geomorphology*. A. Micallef, S. Krastel and A. Savini, Cham, Springer International Publishing, pp. 25-41, doi:10.1007/978-3-319-57852-1\_3.
- Lam, S. K., Pitrou, A. and Seibert, S. (2015). Numba: a LLVM-based Python JIT compiler. *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, Austin, Texas, Association for Computing Machinery, doi:10.1145/2833157.2833162.
- Le Deunf, J., Debese, N., Schmitt, T. and Billot, R. (2020). A Review of Data Cleaning Approaches in a Hydrographic Framework with a Focus on Bathymetric Multibeam Echosounder Datasets. *Geosciences*, 10(7), doi:10.3390/geosciences10070254.
- Masetti, G., et al. (2020). E-learning Python for Ocean Mapping: Empowering the next generation of ocean mappers with effective programming skills. *Canadian Hydrographic Conference 2020*, Québec City, Canada, doi:10.13140/RG.2.2.33921.40806.
- Masetti, G., Faulkes, T. and Calder, B. R. (2019). Opening the Black Boxes in Ocean Mapping: Design and Implementation of the HydrOffice Framework. *AMSA 2019*, Freemantle, WA, Australia, doi:10.13140/RG.2.2.18516.55681.

- Masetti, G., Faulkes, T., Wilson, M. and Wallace, J. (2022). Effective Automated Procedures for Hydrographic Data Review. *Geomatics*, 2(3), pp. 338-354, doi:10.3390/geomatics2030019.
- Mayer, L. and Roach, J. A. (2021). *The Quest to Completely Map the World's Oceans in Support of Understanding Marine Biodiversity and the Regulatory Barriers WE Have Created*. Leiden, The Netherlands, Brill | Nijhoff, pp. 149-166, doi:10.1163/9789004422438\_009.
- Nagpal, A. and Gabrani, G. (2019). Python for Data Analytics, Scientific and Technical Applications. *Amity International Conference on Artificial Intelligence (AICAI)*, pp. 140-145, doi:10.1109/AICAI.2019.8701341.
- Ozgur, C., Colliau, T., Rogers, G. and Hughes, Z. (2022). MatLab vs. Python vs. R. *Journal of Data Science*, 15(3), pp. 355-372, doi:10.6339/JDS.201707\_15(3).0001.
- Petrelli, M. (2021). *Introduction to Python in Earth Science Data Analysis: From Descriptive Statistics to Machine Learning*. Springer.
- Reades, J. (2020). Teaching on jupyter. *Region*, 7(1), pp. 21-34, doi:10.18335/region.v7i1.282.
- Riley, J., Gallagher, B. and Noll, G. (2001). *Hydrographic Data Integration with PYDRO*. 2nd Int. Conf. On High Resolution Survey in Shallow Water, Portsmouth NH.
- Shrestha, N., Barik, T. and Parnin, C. (2018). It's Like Python But: Towards Supporting Transfer of Programming Language Knowledge. *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pp. 177-185, doi:10.1109/VLHCC.2018.8506508.
- Stančin, I. and Jović, A. (2019), An overview and comparison of free Python libraries for data mining and big data analysis. *42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 977-982, doi:10.23919/MIPRO.2019.8757088.
- Tadlaoui, M. A. and Chekou, M. (2021). A blended learning approach for teaching python programming language: towards a post pandemic pedagogy. *International Journal of Advanced Computer Research*, 11(52), pp. 13-22, doi:10.19101/IJACR.2020.1048120.
- Virtanen, P., et al. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python, *Nature Methods*, 17(3), pp. 261-272, doi:10.1038/s41592-019-0686-2.
- Wigley, R., et al. (2017), Nippon foundation / GEBCO ocean mapping training program at the university of New Hampshire: 13 years of success and alumni activities. *OCEANS 2017 - Aberdeen*, pp. 1-5, doi:10.1109/OCEANSE.2017.8084900.
- Zinovieva, I. S., et al. (2021). The use of online coding platforms as additional distance tools in programming education. *Journal of Physics: Conference Series*, 1840(1), pp. 1-16, doi:10.1088/1742-6596/1840/1/012029.

## 8. AUTHORS BIOGRAPHIES

**Giuseppe Masetti** received an M.S. degree in Ocean Engineering from the University of New Hampshire in 2012, a Master in Marine Geomatics and a Ph.D. degree in System Monitoring and Environmental Risk Management from the University of Genoa in 2008 and 2013, respectively. After having served with the hydrographic service of the Italian Navy, he was a Tyco Post-Doctoral Fellow with the Center for Coastal and Ocean Mapping where he is currently an adjunct research faculty. In 2019 he has joined the Danish Hydrographic Office. Email: [giuseppe.masetti@unh.edu](mailto:giuseppe.masetti@unh.edu)

**Semme Dijkstra** holds a Bachelor of Science in Hydrographic Surveying from the Hogere Zeevaartschool Amsterdam and PhD in Geodesy and Geomatics Engineering from the University of New Brunswick. He was a postdoctoral fellow at the Alfred Wegener Institute and has been at the Center for Coastal and Ocean Mapping at the University of New Hampshire, where he currently is a clinical professor and is responsible for the curriculum. Email: [semmed@ccom.unh.edu](mailto:semmed@ccom.unh.edu)

**Rochelle Wigley** has a PhD in Geology from the University of Cape Town. Dr. Wigley is Project Director for Nippon Foundation / GEBCO Training Program in Ocean Mapping at the Center for Coastal and Ocean Mapping/Joint Hydrographic Center, University of New Hampshire. She oversees all administrative, educational and management issues related to the Training Program, mentors the current students and is actively involved in maintaining the international network of the Training Program alumni through the Map the Gaps Non-Profit. She was also co-lead of the GEBCO-Nippon Foundation Alumni team, grand winners of the Shell Ocean Discovery XPRIZE challenge. Email: [rochelle@ccom.unh.edu](mailto:rochelle@ccom.unh.edu)

**Tyanne Faulkes** is a physical scientist with the NOAA Office of Coast Survey. She works at the Pacific Hydrographic Branch in Seattle, Washington. She is primarily involved in data verification, software development, and ship support. Email: [tyanne.faulkes@noaa.gov](mailto:tyanne.faulkes@noaa.gov)