



**UNIVERSITY OF PUERTO RICO-  
MAYAGUEZ  
ENGINEERING FACULTY**



# **Health Care Protocol**

**Team Members:**

Rex J Reyes Rodriguez - [rex.reyes@upr.edu](mailto:rex.reyes@upr.edu)  
Emmanuel Mercado Centeno - [emmanuel.mercado@upr.edu](mailto:emmanuel.mercado@upr.edu)  
Carlos J Ayala Amorós - [carlos.ayala11@upr.edu](mailto:carlos.ayala11@upr.edu)  
Yetsiel S Aviles Bracero - [yetsiel.aviles@upr.edu](mailto:yetsiel.aviles@upr.edu)

Programming Languages  
Profesor: Wilson Rivera

## I. Introduction

The objective of this project was to create a programming language capable of facilitating process necessities to maintain the services and protocols of a hospital, in a more accessible way. Thinking as programmers, we made this programming language with the perspective that anyone could use it. Every method of the programming language was intended to be not just easy to learn, but also short and simple so that no programming knowledge is required. The user is able to perform multiple methods such as get access of a list of all the services that the hospital offers, protocols, list of doctors on the hospital and those that are available, a list of patients and beds available, and the feature of adding doctors and patients. This programming language was designed to be simple, but effective, and to be used as a base for all future programming languages regarding health protocols and hospital maintenance.

## II. Language Tutorial

### A. Installation

1. Make sure you have Python 3.7.2 or higher installed.
2. Download the .zip file or clone from the Github repository at <https://github.com/yetsielaviles/Health-Protocol>
3. Go to the directory where the project is located.
4. Once inside the Health-Protocol folder, run `py parser.py` to start.

**Note:** you may need to use `python3 parser.py` if you have other versions of python installed.

### B. Get Started

1. Once the project is running, you can start using any of the functions supported, refer to the Reference Manual for more details.

### III. Language Reference Manual

Command	Attribute	Attribute Description
create_protocol	<b>(NAME):</b>	String representing name of the protocol
protocols	:	Colon to symbolize end of method
addDoctor	<b>(NAME, BIRTHDAY, SSN, AVAILABILITY, ID):</b>	Name string representing name of doctor Birthday digits separated by / representing day of birth (example 01/16/1994) SSN 4 digits representing Social Security Number Availability (True or False) ID string representing ID of doctor
addPatient	<b>(NAME, BIRTHDAY, SSN, PROTOCOL, ID):</b>	Name string representing name of patient Birthday digits separated by / representing day of birth (example 01/16/1994) SSN 4 digits representing Social Security Number Protocol string representing protocol established to patient ID string representing ID of doctor
beds	:	Colon to symbolize end of method
beds	<b>(DIGIT)</b>	Digit representing integer to search for beds whose floor begins with said digit
doctors	<b>( AVAILABILITY):</b>	Availability (True or False)
services	:	Colon to symbolize end of method
update	<b>(ID, AVAILABILITY)</b>	ID string representing ID of doctor Availability (True or False)

info	(ID)	ID string representing ID of doctor
------	------	-------------------------------------

#### IV. Language Development

The Health Protocol Language was developed using Python, and was mainly implemented using five files, located inside the project. The `lexer.py` file implements the language lexer using the PLY library, used for keywords and commands. The language uses a variety of tokens and reserved words, which are also considered to be in the list of tokens.

The second file that was implemented was the `parser.py`, which implements the parser used in our language. This file has all the rules of our language and is in charge of executing each command, meaning the name of the command and also its required input tokens for each particular command. If a command has an incorrect input, our language will not execute and display an error message.

For our intermediate code, and all functions in charge of implementing our commands correctly, we implemented a class named `HPTools.py`. This class is responsible of managing all the dictionaries, used to save the doctors, doctors available, beds, and it also is in charge of adding and updating doctors, patients, protocols and services.

#### V. Conclusion

Our goal for this project was to apply what we learned in the Programming Language class, and also to create a programming language that was very simple, and easy to use. We kept in mind that not anyone has basic programming knowledge, therefore, our goal was to implement a basic programming language that anyone could use and learn. We had to learn

about lexer analyzers, syntax analysis, and how to use existing libraries in Python about lexers and parsers that could help us develop our programming language. The main goal was to make the life and workplace of people with no knowledge about programming languages easier, by helping them organize and optimize their already existing health protocols.