# IS601: Python for Web API Development

Matt Toegel – matthew.toegel@njit.edu

#### **Table of Contents**

- 1. Description
- 2. Prerequisites
- 3. Instructor
- 4. General Information
- 5. Course Learning Outcomes
- 6. Resources
- 7. Grading
- 8. Exams
- 9. Project
- 10. Course Schedule
- 11. Policies
  - 11.1. Academic Integrity
  - 11.2. Requesting Accommodations
  - 11.3. Resources for NJIT Students
  - 11.4. Class Etiquette
  - 11.5. Proctoring
  - 11.6. Late Work/Makeups
- 12. Closing Notes

## 1. Description

This course is meant as an introduction to the Python programming language through the lens of transferring data on the web. Students will learn the general structures and syntax of Python programs and be able to generate their own programs to move data via HTTP. Students will work with common web interchange formats to both make requests and provide responses. By the conclusion of the course, students will have built a backend web program that supports creating, reading, updating, and deleting objects stored on a database.

## 2. Prerequisites

While this is an introductory course, students are still expected to have some computing background. Students should be familiar with file system structure, running programs from the command line, installing programs, editing text files, and basic networking concepts.

If you are not well versed in these subjects, here are some resources that may help you get up to speed:

- <u>GCFGlobal Basic Computer Skills</u> (https://edu.gcfglobal.org/en/basic-computer-skills/)
- <u>Debugging: The 9 Indispensable Rules for Finding Even the Most Elusive Software and Hardware Problems</u> (https://www.amazon.com/Debugging-Indispensable-Software-Hardware-Problems/dp/0814474578)
- <u>codeacademy Operating Systems: Filesystems</u> (https://www.codecademy.com/learn/operating-systems-filesystems)
- djangogirls Introduction to the command-line interface (https://tutorial.djangogirls.org/en/intro\_to\_command\_line/)

- <u>Visual Studio Code Getting started with Visual Studio Code</u> (https://code.visualstudio.com/docs/introvideos/basics)
- <u>simplilearn</u> What is Client-Server Architecture? Everything You Should Know (https://www.simplilearn.com/what-is-client-server-architecture-article)

#### 3. Instructor

#### Matt Toegel

Email: matthew.toegel@njit.edu Discord: MattToegel Github: MattToegel

## 4. General Information

Course Number:	IS601
Course Title:	Python for Web API Development
Section(s):	001
Semester:	Fall 2024
Date & Time:	<b>001:</b> Tuesday/Thursday 2:30 pm - 3:50 pm
Modality:	Face-to-Face
Credits:	3
Office Hours:	<b>CKB Public Area/Lounge Main Floor:</b> Monday/Wednesday 10 am - 11:20 am General availability via Discord via a provided communication channel

## 5. Course Learning Outcomes

- 1. Students can use Python as a general purpose programming language
  - a. Students can install and run Python
  - b. Students can design an algorithm to solve a problem
  - c. Students can write and run a Python script
  - d. Students can follow standard conventions for software development
- 2. Students can utilize Python packages
  - a. Students can install a Python package
  - b. Students can use Python modules to break a project into components
  - c. Students can utilize the modules provided by a Python package in a Python script
- 3. Students can use Python to send and receive HTTP traffic.
  - a. Students can use a Python package to make an HTTP request

- b. Students can use a Python package to parse an HTTP response
- c. Students can use a Python package to listen for an HTTP request
- d. Students can use a Python package to send an HTTP response for a received request
- 4. Students can use SQL for general database operations.
  - a. Students can construct a SQL query to create a table
  - b. Students can construct a SQL query to read rows from a table
  - c. Students can construct a SQL query to add rows to a table
  - d. Students can construct a SQL query to update rows in a table
- 5. Students can use Python to interface with a database via SQL.
  - a. Students can use Python to send a SQL query to a database
  - b. Students can use Python to read a SQL query response from a database
- 6. Students can use Python to create a basic backend web API
  - a. Students can use Python to create an object in a database in response to an HTTP request
  - b. Students can use Python to read an object in a database in response to an HTTP request and return it in an HTTP response
  - c. Students can use Python to update an object in a database in response to an HTTP request
  - d. Students can use Python to delete an object in a database in response to an HTTP request
  - e. Students can utilize security best practices when creating a backend web API

#### 6. Resources

- To better match industry trends and standards, this class will utilize online resources in lieu of a traditional textbook. Resources will be provided as the course progresses, but some primary resources are listed here:
  - <u>w3schools Python Tutorial</u> (https://www.w3schools.com/python/default.asp)
  - <u>Python The Python Tutorial</u> (https://docs.python.org/3/tutorial/)
  - <u>w3schools SQL Tutorial</u> (https://https://www.w3schools.com/sql/default.asp)
  - FastAPI Introductory Sections and Tutorials (https://fastapi.tiangolo.com/learn/)
- This class utilizes the <u>Canvas learning management system</u> (https://canvas.njit.edu). You can find assignments, assessments, and learning resources there.
- This class will utilize the Respondus proctoring system for the midterm and final.
- This class also has a <u>GitHub page</u> (https://github.com/MattToegel/is601) where you can find out how the course materials are built and have changed over time.
- Students are required to have a device that meets the <u>YWCC minimum specifications</u> (https://ist.njit.edu/student-computers-recommended-specs). They will need administrative access to this device to install software. The device should be functional (charged, working) and brought to class.

## 7. Grading

Assignments and assessments are graded via SpeedGrader with comments left using the same system. Feedback on assignments, midterms, finals, and project deliverables will be delivered within two weeks of their due date.

The course grade is weighted based on five assignment categories:

Exercises	20%
Midterm	20%
Midterm Project	20%
Final	20%
Final Project	20%

## 8. Exams

All exams and quizzes will use Respondus, so be sure to bring a compatible device with you on the day of the assessment. The midterm will take place during a regular class period and will cover the material from weeks one to six. The final exam will be during finals week. The final is not strictly cumulative and largely covers the material from the midterm until the end of the course. That being said, a complete understanding of the second half topics will require using some knowledge from the beginning of the course. Exams and quizzes will be closed book and must be taken in the classroom if the class is meeting face-to-face.

## 9. Project

This course will include a midterm and final project to be completed individually. The goal of the project is to give students a more long-term development experience than is possible with the weekly exercises. In order to maintain academic integrity, students will be asked to use a version control system so they can demonstrate through their commit log that they completed the project individually over the course of several weeks.

## 10. Course Schedule

Each class, students will receive a class lecture and perform an exercise Students are expected to use the lessons learned in class to continue their project work between each class session.

Date	Торіся	Learning Outcomes
Week 1	<ul> <li>Python's Purpose, Usage, and History</li> <li>Installing Python</li> <li>Basic Input and Output</li> <li>Interactive Mode</li> <li>Indentation and Basic Syntax</li> <li>Variables and Types</li> <li>Conditionals</li> <li>Loops</li> <li>Problem Solving in Steps</li> <li>Installing/Using VSCode</li> <li>Writing a Python Script</li> </ul>	<ul> <li>Students can install and run Python</li> <li>Students can design an algorithm to solve a problem</li> <li>Students can write and run a Python script</li> </ul>
Week 2	<ul> <li>Python Conventions</li> <li>Lists, Dictionaries and Iteration</li> <li>Strings</li> <li>Functions</li> <li>Python Modules, Packages, and Virtual Environments</li> <li>Version Control and git</li> <li>Pytest</li> </ul>	<ul> <li>Students can design an algorithm to solve a problem</li> <li>Students can write and run a Python script</li> <li>Students can follow standard conventions for software development</li> <li>Students can install a Python package</li> <li>Students can use Python modules to break a project into components</li> <li>Students can utilize the modules provided by a Python package in a Python script</li> </ul>
Week 3	<ul><li>Classes</li><li>Exceptions</li><li>File Operations</li><li>JSON</li></ul>	<ul> <li>Students can design an algorithm to solve a problem</li> <li>Students can write and run a Python script</li> <li>Students can follow standard conventions for software development</li> <li>Students can install a Python package</li> <li>Students can utilize the modules provided by a Python package in a Python script</li> </ul>

Date	Topics	Learning Outcomes
Week 4	<ul> <li>Namespaces and Scope</li> <li>Elements of Pythonic Style</li> <li>Linters and Formatters</li> <li>Project Structure</li> <li>GitHub</li> </ul>	<ul> <li>Students can design an algorithm to solve a problem</li> <li>Students can write and run a Python script</li> <li>Students can follow standard conventions for software development</li> <li>Students can install a Python package</li> <li>Students can use Python modules to break a project into components</li> <li>Students can utilize the modules provided by a Python package in a Python script</li> </ul>
Week 5	<ul><li>Lambda</li><li>Dates</li><li>Sets and Tuples</li><li>Casting</li></ul>	<ul> <li>Students can design an algorithm to solve a problem</li> <li>Students can write and run a Python script</li> <li>Students can follow standard conventions for software development</li> <li>Students can utilize the modules provided by a Python package in a Python script</li> </ul>
Week 6	Midterm Review	<ul> <li>Students can install and run Python</li> <li>Students can design an algorithm to solve a problem</li> <li>Students can write and run a Python script</li> <li>Students can follow standard conventions for software development</li> <li>Students can install a Python package</li> <li>Students can use Python modules to break a project into components</li> <li>Students can utilize the modules provided by a Python package in a Python script</li> </ul>

Date	Topics	Learning Outcomes
Week 7	Midterm	<ul> <li>Students can install and run Python</li> <li>Students can design an algorithm to solve a problem</li> <li>Students can write and run a Python script</li> <li>Students can follow standard conventions for software development</li> <li>Students can install a Python package</li> <li>Students can use Python modules to break a project into components</li> <li>Students can utilize the modules provided by a Python package in a Python script</li> </ul>
Week 8	<ul> <li>The 9 Indispensable Rules for Debugging</li> <li><u>Python Debugging</u> (https://www.debuggingbook.org/)</li> <li>How to Google Error Messages</li> </ul>	<ul> <li>Students can design an algorithm to solve a problem</li> <li>Students can follow standard conventions for software development</li> </ul>
Week 9	<ul> <li>Structured Query Language</li> <li>Data Schema</li> <li>SQLite</li> <li>Python sqlite3</li> </ul>	<ul> <li>Students can write and run a Python script</li> <li>Students can install a Python package</li> <li>Students can utilize the modules provided by a Python package in a Python script</li> <li>Students can construct a SQL query to create a table</li> <li>Students can construct a SQL query to read rows from a table</li> <li>Students can construct a SQL query to add rows to a table</li> <li>Students can construct a SQL query to update rows in a table</li> <li>Students can use Python to send a SQL query to a database</li> <li>Students can use Python to read a SQL query response from a database</li> </ul>

Date	Topics	Learning Outcomes
Week 10	<ul> <li>The Hypertext Transport Protocol</li> <li>URIs</li> <li>HTTP Headers</li> <li>Cookies</li> <li>Python Requests Package</li> </ul>	<ul> <li>Students can write and run a Python script</li> <li>Students can install a Python package</li> <li>Students can utilize the modules provided by a Python package in a Python script</li> <li>Students can use a Python package to make an HTTP request</li> <li>Students can use a Python package to parse an HTTP response</li> </ul>
Week 11	<ul> <li>REST APIs</li> <li>OpenAPI Schema</li> <li>JSON Schema</li> <li>Python FastAPI</li> <li>FastAPI Tests</li> </ul>	<ul> <li>Students can follow standard conventions for software development</li> <li>Students can write and run a Python script</li> <li>Students can install a Python package</li> <li>Students can utilize the modules provided by a Python package in a Python script</li> <li>Students can use a Python package to listen for an HTTP request</li> <li>Students can use a Python package to send an HTTP response for a received request</li> </ul>
Week 12	<ul><li>Web Security</li><li>OAuth2 and Bearer Tokens</li><li>Injection Attacks</li><li>CORS</li></ul>	<ul> <li>Students can follow standard conventions for software development</li> <li>Students can utilize security best practices when creating a backend web API</li> </ul>

Date	Торіся	Learning Outcomes
Week 13	<ul> <li>Integration</li> <li>Project Organization</li> <li>Documentation</li> <li>FastAPI and sqlite</li> </ul>	<ul> <li>Students can follow standard conventions for software development</li> <li>Students can write and run a Python script</li> <li>Students can utilize the modules provided by a Python package in a Python script</li> <li>Students can construct a SQL query to create a table</li> <li>Students can construct a SQL query to read rows from a table</li> <li>Students can construct a SQL query to add rows to a table</li> <li>Students can construct a SQL query to update rows in a table</li> <li>Students can construct a SQL query to update rows in a table</li> <li>Students can construct a SQL query to update rows in a table</li> <li>Students can use Python to send a SQL query to a database</li> <li>Students can use Python to read a SQL query response from a database</li> <li>Students can use a Python package to listen for an HTTP request</li> <li>Students can use a Python package to send an HTTP response for a received request</li> <li>Students can use Python to read an object in a database in response to an HTTP request and return it in an HTTP response</li> <li>Students can use Python to update an object in a database in response to an HTTP request</li> <li>Students can use Python to delete an object in a database in response to an HTTP request</li> <li>Students can use Python to delete an object in a database in response to an HTTP request</li> <li>Students can use Python to delete an object in a database in response to an HTTP request</li> <li>Students can use Python to delete an object in a database in response to an HTTP request</li> </ul>

Date	Торіся	Learning Outcomes		
Week 14 Troubleshooting / Work Session	<ul> <li>Students can design an algorithm to solve a problem</li> <li>Students can follow standard conventions for software development</li> </ul>			
	<ul><li>Students can write and run a Python script</li><li>Students can install a Python package</li></ul>			
		• Students can utilize the modules provided by a Python package in a Python script		
		• Students can construct a SQL query to create a table		
		• Students can construct a SQL query to read rows from a table		
		• Students can construct a SQL query to add rows to a table		
		• Students can construct a SQL query to update rows in a table		
		• Students can use Python to send a SQL query to a database		
				• Students can use Python to read a SQL query response from a database
		<ul> <li>Students can use a Python package to listen for an HTTP request</li> </ul>		
		<ul> <li>Students can use a Python package to send an HTTP response for a received request</li> </ul>		
		• Students can use Python to create an object in a database in response to an HTTP request		
		<ul> <li>Students can use Python to read an object in a database in response to an HTTP request and return it in an HTTP response</li> </ul>		
		• Students can use Python to update an object in a database in response to an HTTP request		
		• Students can use Python to delete an object in a database in response to an HTTP request		
		• Students can utilize security best practices when creating a backend web API		

Date	Торіся	Learning Outcomes
Week 15	Final Review	• Students can design an algorithm to solve a problem
		• Students can follow standard conventions for software development
		• Students can write and run a Python script
		• Students can install a Python package
		<ul> <li>Students can utilize the modules provided by a Python package in a Python script</li> </ul>
		• Students can construct a SQL query to create a table
		• Students can construct a SQL query to read rows from a table
		• Students can construct a SQL query to add rows to a table
		• Students can construct a SQL query to update rows in a table
		<ul> <li>Students can use Python to send a SQL query to a database</li> </ul>
		• Students can use Python to read a SQL query response from a database
		<ul> <li>Students can use a Python package to listen for an HTTP request</li> </ul>
		• Students can use a Python package to send an HTTP response for a received request
		• Students can use Python to create an object in a database in response to an HTTP request
		<ul> <li>Students can use Python to read an object in a database in response to an HTTP request and return it in an HTTP response</li> </ul>
		<ul> <li>Students can use Python to update an object in a database in response to an HTTP request</li> </ul>
		• Students can use Python to delete an object in a database in response to an HTTP request
		• Students can utilize security best practices when creating a backend web API

Date	Торіся	Learning Outcomes
Final (Date TBD)	Final Exam	<ul><li>Students can design an algorithm to solve a problem</li><li>Students can follow standard conventions for software</li></ul>
		development
		• Students can write and run a Python script
		Students can install a Python package
		<ul> <li>Students can utilize the modules provided by a Python package in a Python script</li> </ul>
		• Students can construct a SQL query to create a table
		• Students can construct a SQL query to read rows from a table
		• Students can construct a SQL query to add rows to a table
		• Students can construct a SQL query to update rows in a table
		• Students can use Python to send a SQL query to a database
		• Students can use Python to read a SQL query response from a database
		<ul> <li>Students can use a Python package to listen for an HTTP request</li> </ul>
		• Students can use a Python package to send an HTTP response for a received request
		• Students can use Python to create an object in a database in response to an HTTP request
		<ul> <li>Students can use Python to read an object in a database in response to an HTTP request and return it in an HTTP response</li> </ul>
		<ul> <li>Students can use Python to update an object in a database in response to an HTTP request</li> </ul>
		• Students can use Python to delete an object in a database in response to an HTTP request
		• Students can utilize security best practices when creating a backend web API

## 11. Policies

#### 11.1. Academic Integrity

Academic Integrity is the cornerstone of higher education and is central to the ideals of this course and the university. Cheating is strictly prohibited and devalues the degree that you are working on. As a member of the NJIT community, it is your responsibility to protect your educational investment by knowing and following the <u>academic code of integrity policy</u>.

(http://www5.njit.edu/policies/sites/policies/files/academic-integrity-code.pdf)

Please note that it is my professional obligation and responsibility to report any academic misconduct to the Dean of Students Office. Any student found in violation of the code by cheating, plagiarizing or using any online software inappropriately will result in disciplinary action. This may include a failing grade of F, and/or suspension or dismissal from the university. If you have any questions about the code of Academic Integrity, please contact the Dean of Students Office at dos@njit.edu.

#### 11.2. Requesting Accommodations

If you are in need of accommodations due to a disability please contact the <u>Office of Accessibility Resources & Services (OARS)</u> (https://www.njit.edu/studentsuccess/accessibility), Fenster Hall Room 260 to discuss your specific needs. A Letter of Accommodation Eligibility from the OARS authorizing your accommodations will be required.

#### 11.3. Resources for NJIT Students

<u>NJIT Service for Students</u> (https://docs.google.com/document/d/1xGO2qcVEF1tsOgZn-\_W1LjSOKn\_jhEVs9IWI\_6jeuPs/edit?usp=sharing), including Technical Support.

#### 11.4. Class Etiquette

Students who are the most successful attend and participate in class. If you have questions, please ask them. This makes the class more dynamic and interesting for everyone.

#### 11.5. Proctoring

NJIT policy requires that all midterm and final exams must be proctored, regardless of delivery mode, in order to increase academic integrity. Note that this does not apply to essay or authentic based assessments. Effective beginning Fall semester 2019, students registered for a fully online course section (e.g., online or Hyflex mode) must be given the option to take their exam in a completely online format, with appropriate proctoring.

Exams will be given in-person using <u>Respondus</u> (https://njit.instructure.com/courses/21706/pages/respondus-lockdown-browser-and-monitor). Be sure to bring your charged laptop and charger on the day of exams.

#### 11.6. Late Work/Makeups

Late work can be turned in before the end of the semester for half credit.

## 12. Closing Notes

Syllabus is subject to change, attend class to stay current.

Last updated 2024-09-10 01:09:22 UTC