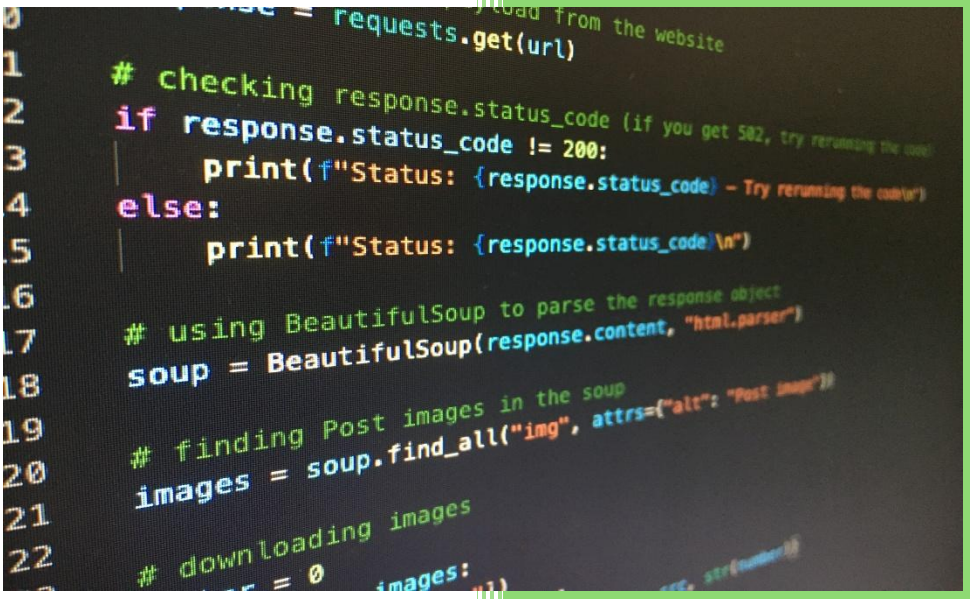
Python Basics for Beginners

Web Design – Project Proposal & HTML



Ramesh Rajagalgoda

10637942

Contents

1. Introduction

1.1 Background

Python has many potential programmers and hobbyists who are keen to learn but have difficulty locating easy and trustworthy material on the basics such as variables, loops, functions and basic syntax. This is aggravated by the fact that resources are usually dispersed across sources and written in technical jargon or are aimed at an advanced audience or commercial classes as opposed to practical and teaching advice. In this sense, a simple, easy to understand, beginner level site on the basics of Python in a non-commercial format is required.

1.2 Proposed Website: Python Basics for Beginners

The suggested site, Python Basics for Beginners, is a non-commercial, educational site that seeks to teach new users the major concepts in Python programming. The website follows well-defined categories (Home, Lessons, Exercises, and Contact/About) based on which the user can navigate to the high-level overview on the home page (/index.html) to various sub-pages like variable basics (/lessons/variables.html), loop examples (/lessons/loops.html), and practice tasks (/exercises/practice.html). This is done through a systematic procedure that promotes step-by-step learning and does not overwhelm the users with unorganized information.

1.3 Scope of the Project

It is also narrowed in its scope, as it only gives simple, didactic information about the basics of Python as a first-time user, but does not cover anything more advanced, or interactive environments. The site will consist of: basic syntax and data types; control structures such as loops and conditionals, basic functions, basic exercises and curated external resources. Neither will it provide region specific advice, product sales nor will it offer any more advanced features such as live code editors but it will give a clear information architecture and bare HTML page as a skeleton to a more visually rich fully developed site in the next development cycle.

2. Project Description and Goals

2.1 Overview of the Website

Python Basics for Beginners is a non-commercial, instructional website that explains the fundamental steps in learning Python, from basic syntax through variables, loops, functions, and simple exercises in a safe and responsible manner. The site is structured with clearly labeled sections - Home, Lessons, Exercises, and Contact/About - allowing beginners to progress gradually from simple explanations on the home page to more detailed advice on specific topics such as defining variables or writing loops. Information architecture is implemented through a series of content-only HTML pages (e.g., /lessons/variables.html, /exercises/practice.html) which together form a structured learning pathway that can be enhanced with visual design and interactivity in the next phase. (Python Software Foundation, 2026)

2.2 Project Goal

It is also narrowed in its scope, as it only gives simple, didactic information about the basics of Python as a first-time user, but does not cover anything more advanced, or interactive environments. The site will consist of: basic syntax and data types; control structures such as loops and conditionals, basic functions, basic exercises and curated external resources. Neither will it provide region specific advice, product sales nor will it offer any more advanced features such as live code editors but it will give a clear information architecture and bare HTML page as a skeleton to a more visually rich fully developed site in the next development cycle.

2.3 Project Objectives

Content and learning objectives

• Include simple explanations of key topics such as (syntax, variables, data types, loops, functions, exercises) using short sections and clear headings on the relevant pages.

• Allow beginner users to follow a basic sequence from introduction to practice (e.g., via /lessons/index.html and /exercises/index.html) so they understand how foundational concepts build upon each other.

Usability and accessibility objectives

• Consistent navigation system across all pages (Home, Lessons, Exercises, Contact) so users can access any main section with one or two clicks, lowering cognitive load for new users.

• Use structure (headings, lists, meaningful link text, simple layout) for easy readability, accessibility, and easy extension with styling during the stage of development.

UX and engagement objectives

• Make the learning experience easy to access by making thorough programming topics into small, understandable pages that align with to how beginners normally ask questions (e.g., “How do I use variables?” or “How do I practice loops?”).

• Support continued exploration by including external links to reputable Python resources, encouraging users to deepen their knowledge beyond the introductory material.

3. User Research

3.1 Target Users (Demographic)

Beginners with interest in programming are the primary target users, which are students, young adults, and hobby learners who use Python to start a personal project or career change, as opposed to professional developers. Research on new programmers indicates that a high number of new entrants are young, usually part-time workers or students, and do not necessarily have formal training in computer science. These are users who usually have minimum digital access (laptops or smartphones) and are familiar with searching online on how to guides but may lack prior coding experience.

3.2 Motivations and Preferences (Psychographic)

Career development, personal interest in technology, self-development and practical skills, such as automation or data analysis, are the motives to motivate target users. Like novice gardeners wishing to adopt healthier lifestyles, they like simple, step-by-step instruction, vivid examples, and hints (e.g., beginner-friendly programs, famous mistakes) more than complex theory, content that is easy to read, look at, and absorb in brief bursts that fit hectic schedules.

3.3 Behaviour and Context of Use (Ethnographic)

Practically, these consumers study in bits: watching videos, reading blogs, joining online forums to get immediate responses during the coding process on their devices, and in most cases, they multitask or access materials during their commuting time. They act in the way of trial and error - small beginnings, trying code, and required resources to a particular query such as software assignment, or syntax of loops, instead of full-fledged books.

3.4 Key Findings and UX Requirements

Key findings

Key Finding	**What it means for users**
Beginners have limited time and formal programming knowledge but strong interest in practical Python basics.	They need simple, direct answers (e.g., "how to use variables") rather than long theoretical chapters.
They are motivated by self-improvement and career skills, preferring trustworthy explanations over promotional content.	They trust neutral, educational guidance that helps them build skills affordably and safely.
Learning is context-driven: they seek info just before or during coding tasks.	They visit pages briefly on mobile and need quick answers.
Users get confused when syntax, data types, and exercises are mixed without clear structure.	Without separation, they can't discern advice for basics vs. practice.
Users cautious about complex or outdated advice, prefer neutral sources emphasizing safe practices.	They avoid ads and favor clear, beginner-first explanations without jargon.

UX requirements for the website

UX Requirement	**How the site will implement it (With examples)**
Map each major user question to a specific section or subpage so answers are easy to find	"What are variables?" → Lessons (/lessons/variables.html); "How do I practice?" → Exercises (/exercises/practice.html); "Where to learn more?" → External links on Contact.
Provide focused pages for each main topic (syntax, loops, exercises) instead of mixing them.	Separate sections: Lessons, Exercises, each with overview and subpages as in site map.
Support quick, task-based reading with scannable content.	Use short paragraphs, headings, bullet lists, and code placeholders on every page for mobile scanning.
Keep navigation shallow and consistent so users never feel lost.	Same top menu (Home, Lessons, Exercises, Contact) on all pages, with overview hubs (e.g., /lessons/index.html).
Build trust with a neutral, educational tone and beginner-first messaging.	Use short paragraphs, headings, bullet lists, and code placeholders on every page for mobile scanning.

4. Project Research (Precedent Study)

4.1 Research Approach

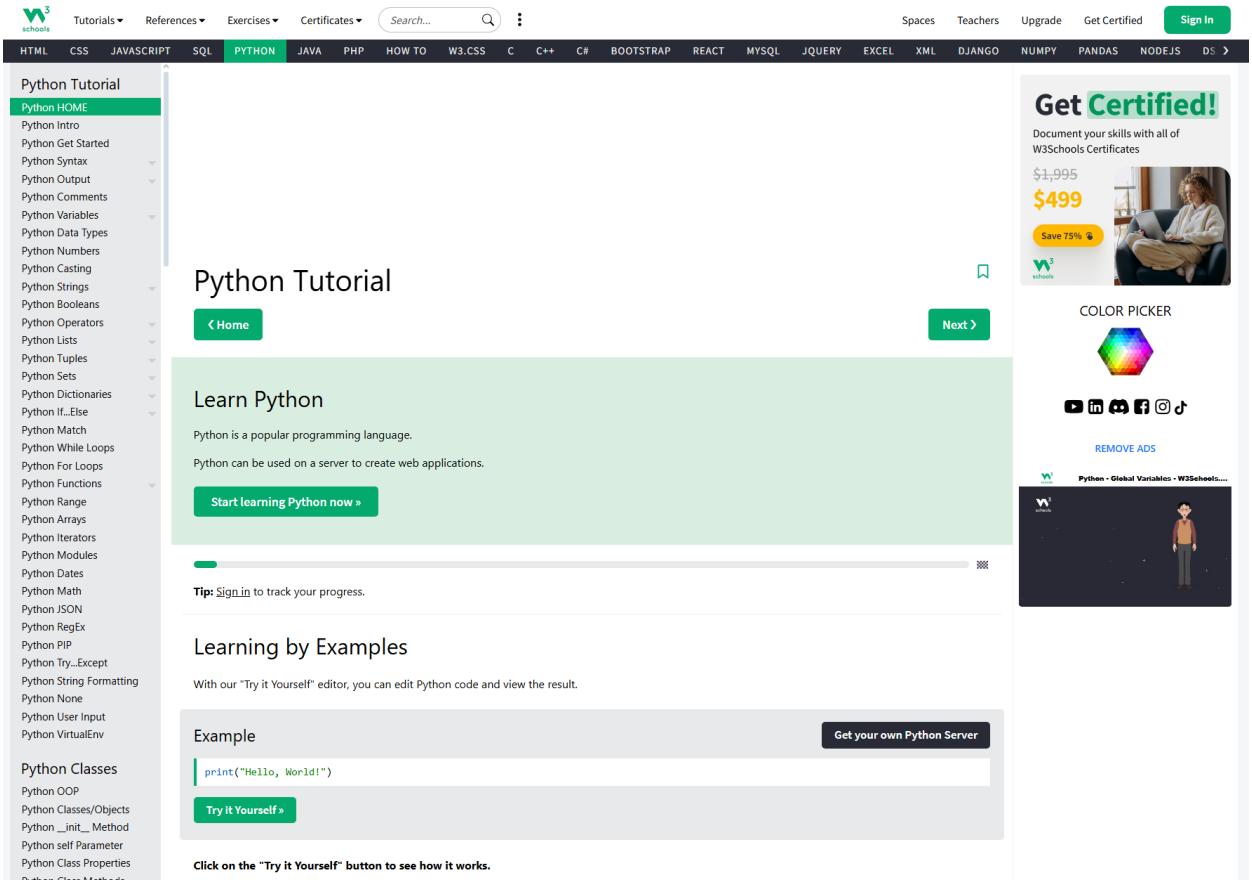
The research focused on identifying existing educational resources for beginner Python learners, including tutorials, wikis, and coding platforms that present practical advice on syntax, variables, loops, and exercises. These were reviewed to understand content structure, emphasized topics (e.g., step-by-step guides, easy starters), and presentation for non-experts. Inspirational educational sites were analyzed for interface patterns, visual hierarchy, and self-paced learning modules to inform the content architecture and UX strategy of Python Basics for Beginners. (Duolingo, 2026)

4.2 Review of Similar Educational Programming Websites

Several Python learning resources provide useful precedents:

W3Schools Python Tutorial

Website: https://www.w3schools.com/python/



The Python Tutorial at W3Schools gives an in-depth and organized reference on learning Python programming. It includes the basic concepts of syntax, variables, the data type, operators, control structures (if-else, loops), functions, modules, file manipulation, and object-oriented programming (OOP). Inherent features Advanced features consist of data structures (lists, tuples, sets, dictionaries), database integration (MySQL, MongoDB), data science tools (NumPy, Pandas, Matplotlib), machine learning concepts, and data structures and algorithms (DSA). The tutorial is focused on the practical learning by means of examples, exercises, quizzes and a Try it Yourself editor where you can immediately execute the code. It also contains references, inbuilt functions and documentations of modules to look up easily.

For this project, these characteristics directly influence the decision to:

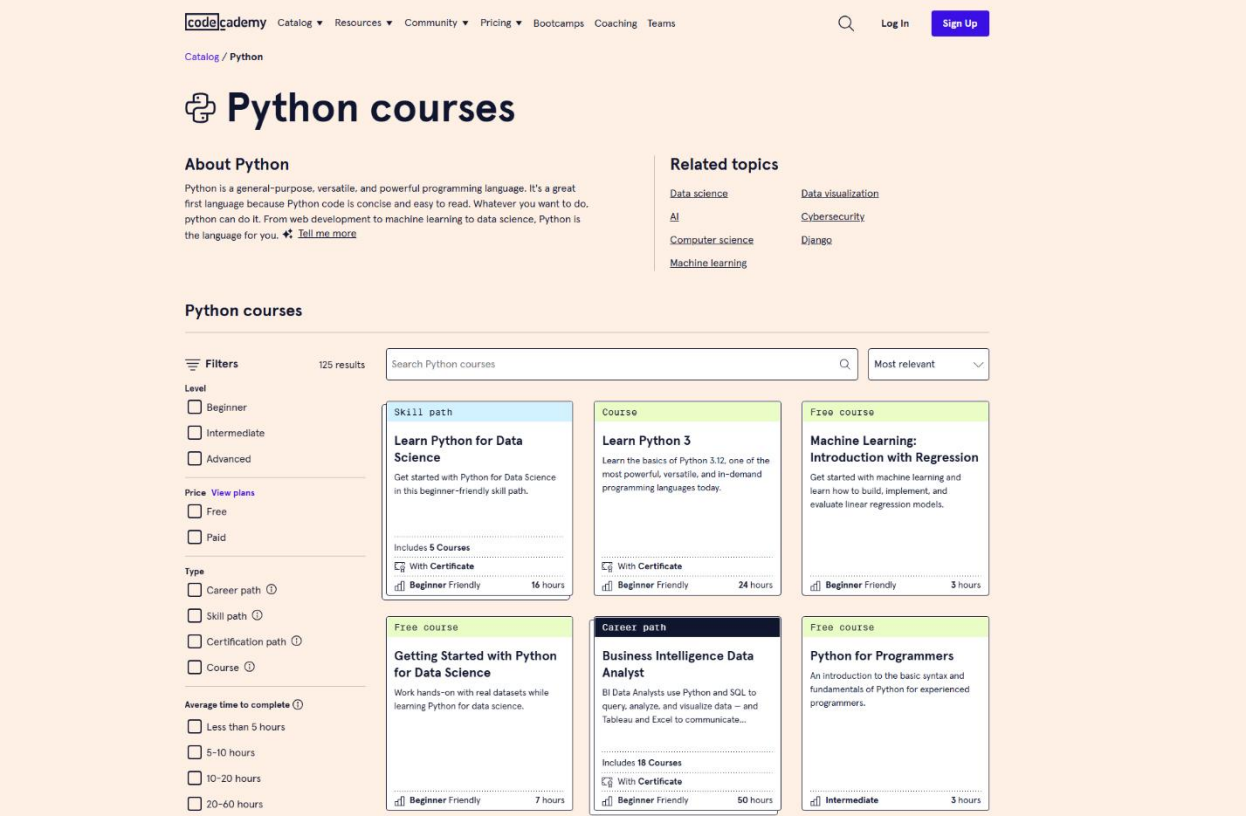
• Represent learning as a clear sequence on the Lessons page (/lessons/index.html), where each concept (variables, loops) is a separate block.

• Include external links on Contact to aggregate resources, as W3Schools does for deeper info.

W3Schools' role as a curated center for programming knowledge supports keeping on-site content introductory and linking to specialist topics.

Codecademy Python Catalog

Website: https://www.codecademy.com/catalog/language/python



The catalog offers entry-level and intermediate-level Python courses in data analysis and ML; badges and course card icons contain icons of duration/certificate. Level/price filters enhance the findability, clean card layout with disclosure progression, users are guided by the UX by recommendation and become engaged with the certificates.

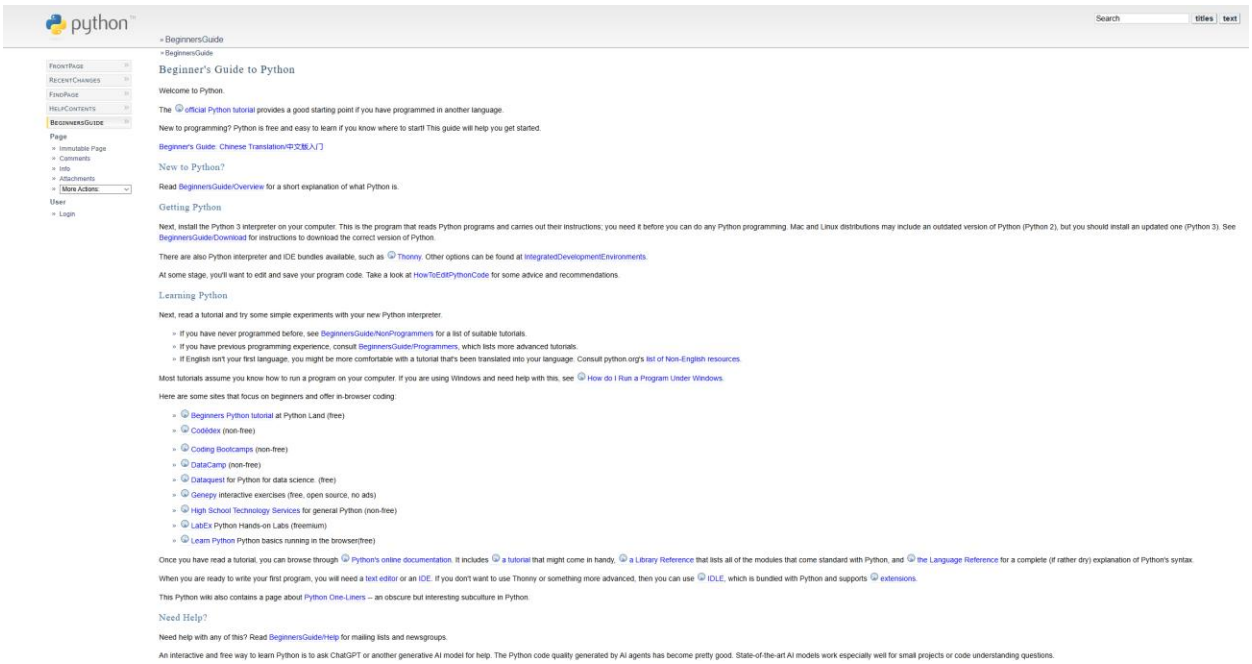
How it influenced this project:

• As a beginner-friendly guide with clear simple language.

• Supports the Lessons and Exercises sections (/lessons/index.html, /exercises/index.html) emphasizing easy starters.

Python Wiki Beginners Guide

Website: https://wiki.python.org/moin/BeginnersGuide



It is a newcomer guide to Python, so it provides easy steps and instructions to beginners, assigns them to install Python 3 and suggests using such tools as Thonny or IDLE to write the code. The guide is divided into learning materials based on the user experience and language, even non-English tutorials. It includes in-browser code platforms and links to the official documentation of Python. Other tools include help resources, beginner quizzes, AI and prompt engineering, and contribution guidelines.

How it influenced this project:

• The project has specific areas devoted to the basics and exercises, the application of principles to syntax and practice to beginners..

• Emphasis on natural learning encourages the inclusion of non-complex options in Lessons.

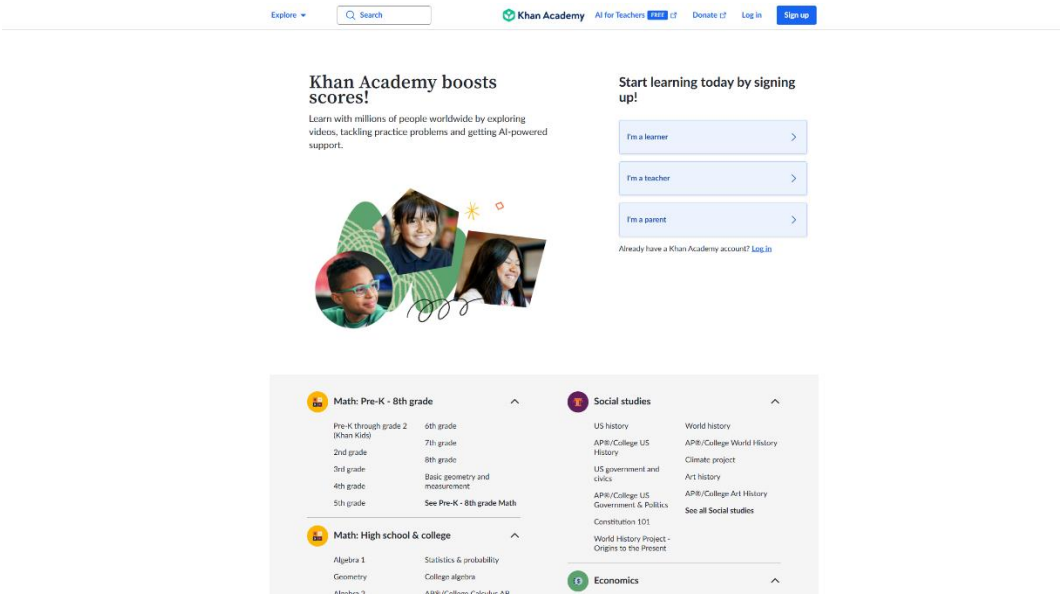
Collectively, these websites demonstrate the use of good programming pedagogy when delivered to beginners as stepwise procedures, with the foundational subjects segregated, and the materials described in simple language.

4.3 Review of Inspirational UI/UX Websites

UI/UX qualities were studied in inspirational educational websites:

• Khan Academy: It is free online learning in a variety of disciplines that focus on personalized learning, professionally created content, and teaching tools. Well-organized form with sign-up buttons; UX cultivates confidence through testimonials and artificial intelligence.

Website: https://www.khanacademy.org/



Considering the UI/UX, this site is an example of consistent navigation, modular blocks and self-paced elements, used in the design of the planned HTML pages.

4.4 Design Insights for This Project

The findings of the research resulted in tangible solutions:

• Organized design: It is designed after tutorials, sections of the site are aligned to the learner tasks (lessons, exercises).

• Practice: Based on the interactive platform, Exercises section emphasizes practical.

• Helpful navigation: On portals, beginners like predictable menus, thus stick with top nav.

• Curated resources: Use Contact to connect outside contents, which are kept on site.

5. Information Architecture

5.1 Summary of Content for Each Main Section

• Home (/index.html): Introduces site purpose, briefly describes sections with links.

• Lessons: Covers fundamentals (variables, loops, functions) with examples.

• Exercises: Provides practice tasks to reinforce learning.

• Contact/About: Email link, external resources to Python docs.

6. UX Design Strategy

6.1 Usefulness: Content to Meet User Needs

• Content addresses practical queries (syntax, variables, exercises) per beginner needs.

• Each question maps to a section (e.g., loops to /lessons/loops.html).

6.2 Usability and Accessibility

• Consistent global nav with clear labels, best for educational sites.

• Semantic HTML (headings, lists) supports readability and WCAG.

6.3 Desirability and Overall Experience

• Wireframes use simple layouts: titles, short paras, code blocks echoing chunked material.

• Positions site as responsible resource appealing to motivated learners.

7. Navigation Structure

7.1 Navigation Approach

• Persistent top nav to all sections. • Overviews as hubs for subpages. • Contextual links support flows.

7.2 Navigation Flowchart with Filenames

Home (index.html) → Lessons (lessons.html) → Exercises (exercises.html) → Contact (contact.html), with bidirectional links.

9. Wireframes

9.1 Wireframe Conventions (Grid, Breakpoints, Patterns)

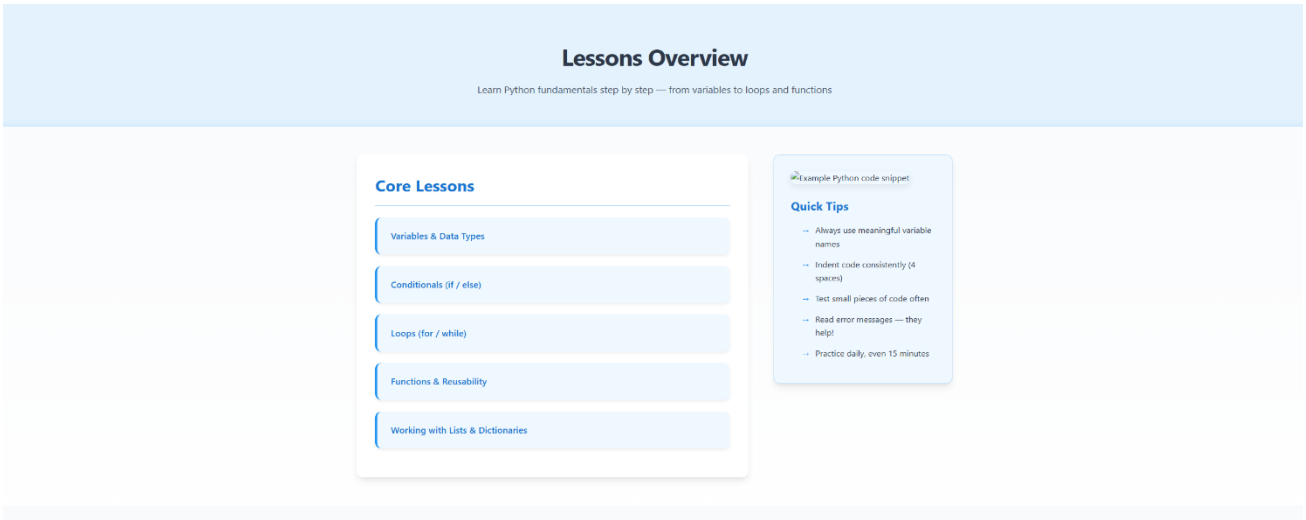
Wireframes use a simple grid: one-column mobile, two-column desktop. Patterns: consistent nav, title blocks, modular content.

9.2 Home Page Wireframe

Header with title, nav menu. Hero with intro, links to sections. Key cards for Lessons, Exercises. Footer with copyright.

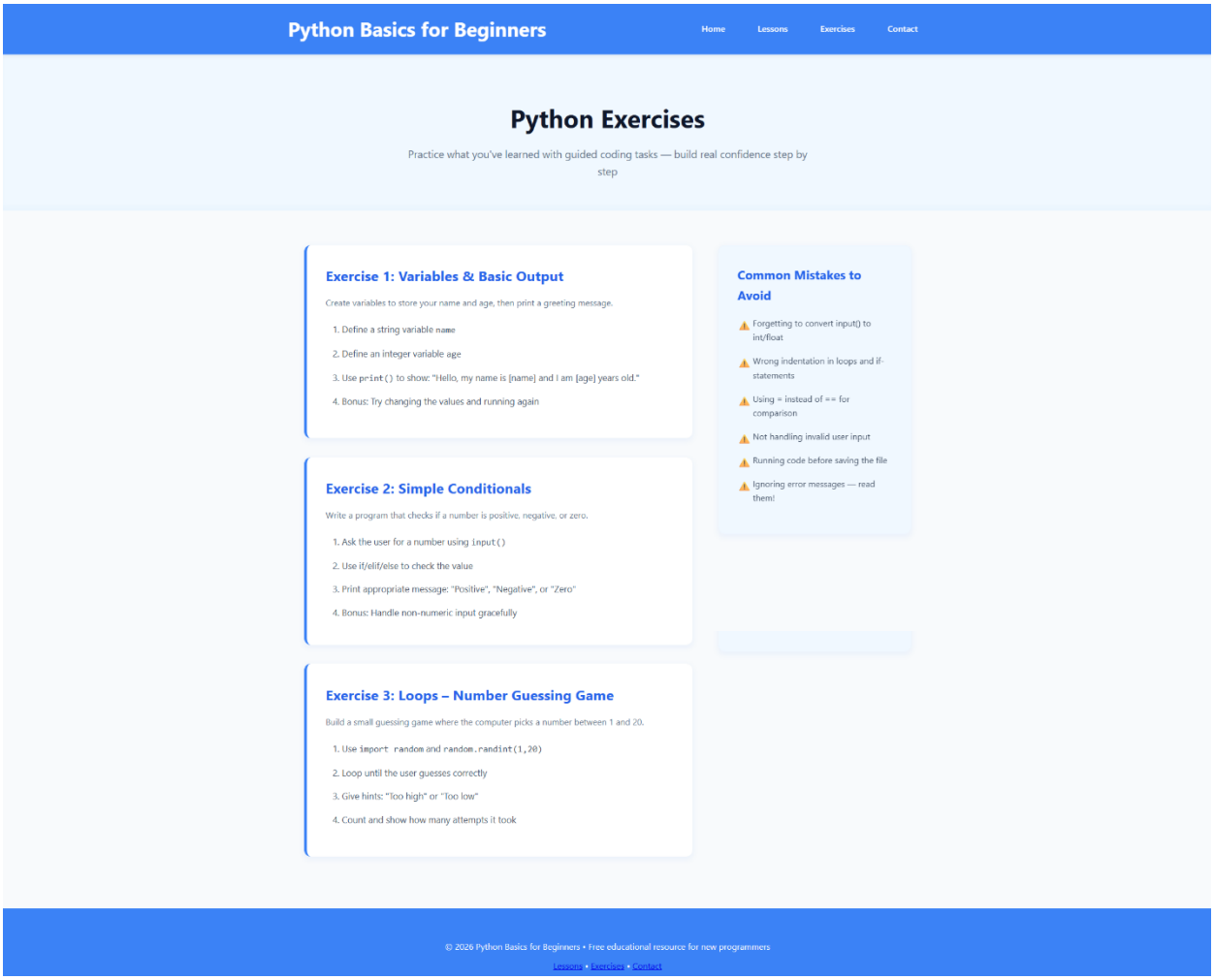
9.3 Lessons Overview Wireframe

Header, title block with intro. Left: subtopic links (variables, loops). Right: code placeholder, tips box. Footer.



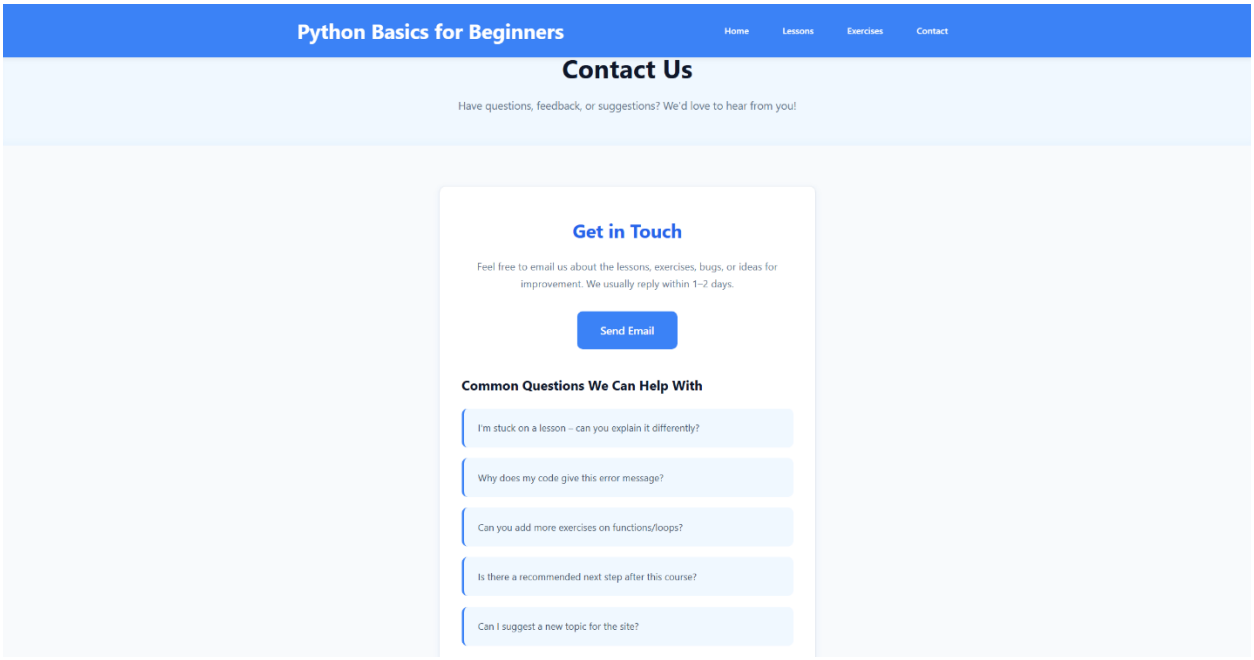
9.4 Exercises Overview Wireframe

Header, title. Vertical step cards for exercises. Sidebar with common errors. Footer.



9.5 Contact Page Wireframe

Header, title. Centered card with email link, external resources. Footer.



10. Conclusion

The Python Basics for Beginners project proposes a focused educational site providing newcomers with fundamentals for building skills, with sections matching user questions for logical pathways. Based on research, it transforms findings into UX specs: clear headings, shallow nav, semantic HTML, external resources. This blueprint can be implemented in Assignment 1 and developed further.

References

Python Software Foundation. (2026). BeginnersGuide

(Org, 2026) **https://wiki.python.org/moin/BeginnersGuide**

W3Schools. (2026). Python Tutorial

(W3Schools, 2026) **https://www.w3schools.com/python/?**

Khan Academy. (2026). Free Online Courses

(Academy, n.d.) **https://www.khanacademy.org/**