Міністерство освіти і науки України

Національний університет «Запорізька Політехніка»

Кафедра програмних засобів

ЗВІТ

з лабораторної роботи №4

з дисципліни «Веб технології та Веб дизайн» на тему:

«Блочна модель документа»

Виконав:

Студент групи КНТ-122 О. А. Онищенко

Прийняли:

Старший викладач: С. Д. Леощенко

2023

4 БЛОЧНА МОДЕЛЬ ДОКУМЕНТА

Мета роботи

Вивчити способи групування та організації елементів. Навчитись керувати розташуванням елементів на сторінці.

Завдання до роботи

1. Ознайомитися з теоретичними відомостями, необхідними для виконання роботи.

2. Обрати персональну тему за варіантом з додатку В.

3. Підготувати сторінку, що є енциклопедичною довідкою з теми. За зразок можна взяти структуру сторінки Вікіпедії. При створенні сторінки використати отримані навички:

- структурувати сторінку, розділивши її на змістовні блоки (шапка, підвал, скорочена довідка в правій колонці, підрозділи);
- оформити текст, застосовуючи заголовки, списки, абзаци, зображення. Кожен елемент повинен мати стилі, всі стилі мають бути винесені в таблицю зв’язаних стилів;
- у підвал сторінки додати власне прізвище, що є посиланням на сторінку-резюме, розроблену в попередніх роботах.

4. Скопіювати проект та розділити сторінку, створену в пункті 4.3.3, на декілька сторінок, кожна з яких містить окремий підрозділ:

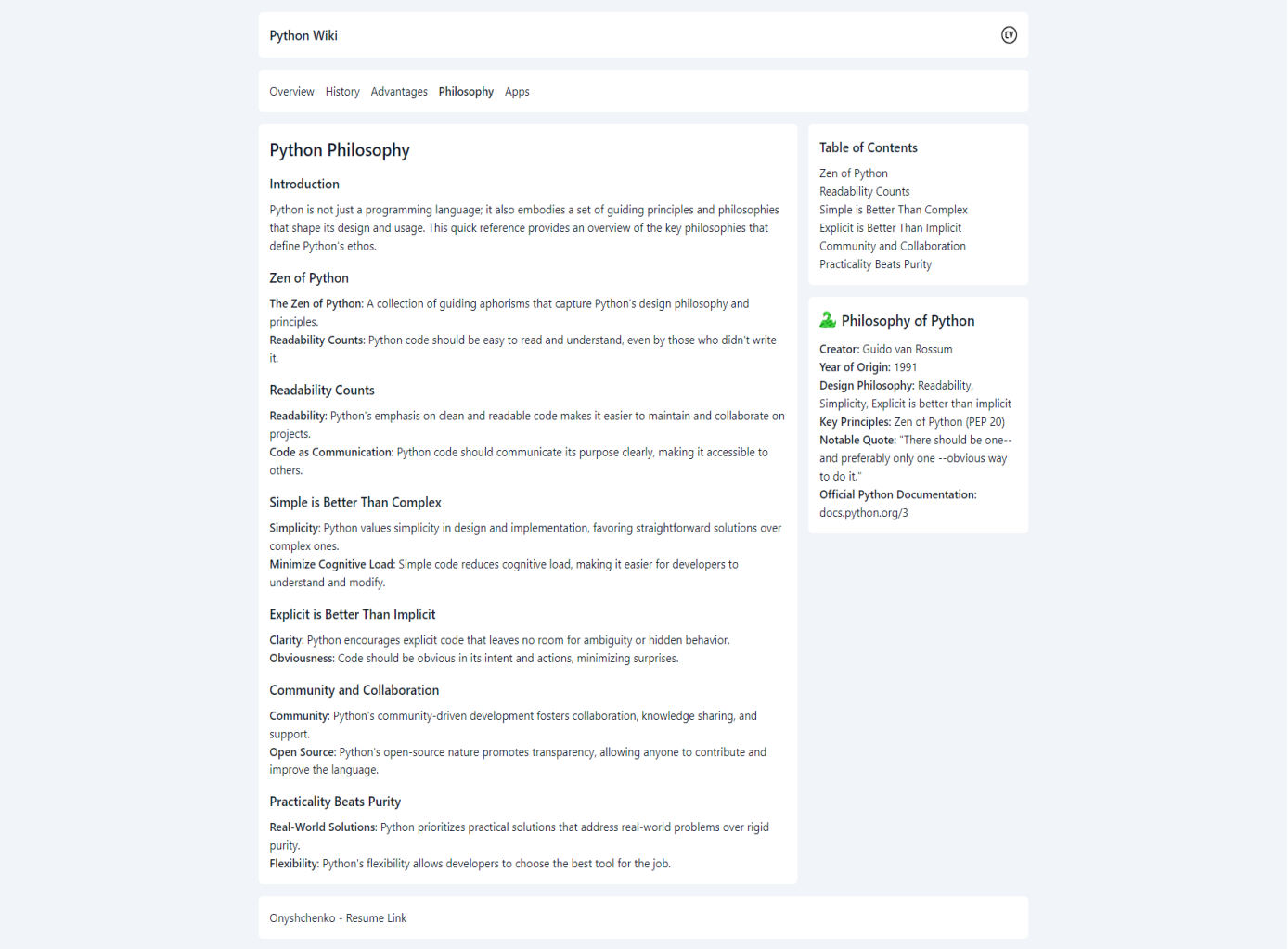
- на кожній зі сторінок зберегти бокову колонку зі скороченою довідкою;
- на всіх сторінках між шапкою та основним змістом додати горизонтальне навігаційне меню, що містить посилання на створені сторінки;

– виділяти стилем поточну сторінку в навігаційному меню.

5. Оформити звіт з роботи.

6. Відповісти на контрольні питання.

Результати виконання роботи



Python Wiki

Table of Contents

Variables and Data Types
Conditional Statements
Loops
Functions
Lists
Dictionaries
Strings
File Handling

Python Lanugage

Designed by: Guido van Rossum
First appeared: 1991
Paradigms: Object-oriented, procedural, functional
Typing discipline: Duck, dynamic, strong typing
Website: www.python.org

Python Overview

Introduction

Python is a high-level, versatile programming language known for its simplicity and readability. This quick reference provides an overview of essential Python concepts and syntax.

Variables and Data Types

Variables: In Python, you can assign values to variables using the = operator. Example: `x = 10`.
Data Types: Python supports various data types, including integers, floats, strings, booleans, and more.
Type Conversion: You can convert between data types using functions like `int()`, `float()`, and `str()`.

Conditional Statements

if Statement: Use `if` to execute code conditionally based on a Boolean expression.
elif and else: Extend `if` statements with `elif` (else if) and `else` for multiple branching.

Loops

for Loop: Iterate over a sequence (e.g., list, string) using a `for` loop.
while Loop: Execute code repeatedly as long as a condition is true with a `while` loop.

Functions

Defining Functions: Create reusable code blocks using the `def` keyword.
Parameters: Pass arguments to functions, which can be optional or required.
Return Values: Functions can return values using the `return` statement.

Lists

Creating Lists: Define lists using square brackets, e.g., `my_list = [1, 2, 3]`.
Indexing: Access list elements using their index (zero-based).
Slicing: Extract portions of a list using slicing, e.g., `my_list[1:3]`.

Dictionaries

Creating Dictionaries: Define key-value pairs using curly braces, e.g., `my_dict = {'name': 'John', 'age': 30}`.
Accessing Values: Retrieve values by specifying the key.

Strings

Creating Strings: Define strings with single or double quotes, e.g., `"Hello, World!"`.
String Methods: Use built-in string methods for operations like concatenation and splitting.

File Handling

Opening Files: Use `open()` to open files for reading or writing.
Reading and Writing: Read file contents with `read()`, write data with `write()`.

Read more

History
Advantages
Philosophy
Examples of programs

Python History

Introduction

Python is a widely-used high-level programming language with a rich history. This quick reference provides an overview of key milestones and events in the development of Python.

Inception

Creation of Python: Python was created by Guido van Rossum and first released in 1991.
Motivation: Guido aimed to create a language that emphasized code readability and allowed programmers to express concepts in fewer lines of code.

Major Versions

Python 2.x: The Python 2 series (e.g., 2.7) was widely used for many years but officially reached end-of-life in 2020.
Python 3.x: Python 3 introduced significant changes to improve consistency and eliminate some of the inconsistencies present in Python 2.

Python Enhancement Proposals (PEP)

PEP 8: This style guide for Python code has been influential in shaping Python's code formatting conventions.
PEP 20 (Zen of Python): A collection of guiding principles for writing computer programs in Python, providing insights into Python's design philosophy.

Notable Contributors

Guido van Rossum: Python's creator and "Benevolent Dictator For Life" (BDFL) until he stepped down in 2018.
Python Software Foundation (PSF): Established in 2001 to promote, protect, and advance Python and its community.

Community and Growth

Python Community: Python has a vibrant and welcoming community of developers, with conferences and user groups worldwide.
Python's Popularity: Python's simplicity and versatility have contributed to its popularity in various fields, including web development, data science, and artificial intelligence.

Table of Contents

Inception
Major Versions
Python Enhancement Proposals (PEP)
Notable Contributors
Community and Growth

The Journey of Python

Creator: Guido van Rossum
Year of Origin: 1991
Programming Paradigms: Object-oriented, Procedural, Functional
Type System: Duck-typing, Dynamic, Strong typing
Official Website: www.python.org

Table of Contents

Inception
Major Versions
Python Enhancement Proposals (PEP)
Notable Contributors
Community and Growth

The Journey of Python

Creator: Guido van Rossum
Year of Origin: 1991
Programming Paradigms: Object-oriented, Procedural, Functional
Type System: Duck-typing, Dynamic, Strong typing
Official Website: www.python.org

Python History

Introduction

Python is a widely-used high-level programming language with a rich history. This quick reference provides an overview of key milestones and events in the development of Python.

Inception

Creation of Python: Python was created by Guido van Rossum and first released in 1991.
Motivation: Guido aimed to create a language that emphasized code readability and allowed programmers to express concepts in fewer lines of code.

Major Versions

Python 2.x: The Python 2 series (e.g., 2.7) was widely used for many years but officially reached end-of-life in 2020.
Python 3.x: Python 3 introduced significant changes to improve consistency and eliminate some of the inconsistencies present in Python 2.

Python Enhancement Proposals (PEP)

PEP 8: This style guide for Python code has been influential in shaping Python's code formatting conventions.
PEP 20 (Zen of Python): A collection of guiding principles for writing computer programs in Python, providing insights into Python's design philosophy.

Notable Contributors

Guido van Rossum: Python's creator and "Benevolent Dictator For Life" (BDFL) until he stepped down in 2018.
Python Software Foundation (PSF): Established in 2001 to promote, protect, and advance Python and its community.

Community and Growth

Python Community: Python has a vibrant and welcoming community of developers, with conferences and user groups worldwide.
Python's Popularity: Python's simplicity and versatility have contributed to its popularity in various fields, including web development, data science, and artificial intelligence.

Python Advantages

Introduction

Python is a versatile programming language known for its numerous advantages in various domains. This quick reference provides an overview of the key benefits of using Python.

Readability

Clear and Readable Syntax: Python's syntax emphasizes readability and reduces the cost of program maintenance.
Indentation-based Structure: Python enforces code indentation, making the code structure visually clear and consistent.

Versatility

General-Purpose Language: Python can be used for a wide range of applications, including web development, data analysis, machine learning, and more.
Integration Capabilities: Python easily integrates with other languages and tools, facilitating complex projects.

Community Support

Active Community: Python has a large and active community of developers who provide support, share knowledge, and create open-source libraries.
Documentation and Resources: Abundant documentation and online resources are available for Python users, making it easy to learn and troubleshoot.

Vast Library Ecosystem

Rich Standard Library: Python's standard library includes modules for various tasks, reducing the need for external dependencies.
Third-Party Libraries: Python boasts a vast ecosystem of third-party libraries and frameworks for specific domains, enhancing productivity.

Cross-Platform

Platform Independence: Python code can run on multiple platforms without modification, increasing portability.
Support for Major Operating Systems: Python supports Windows, macOS, Linux, and more.

Simplicity

Minimalistic Syntax: Python's clean and minimalistic syntax reduces the learning curve for beginners.
Expressive and Concise Code: Python allows developers to express complex ideas in fewer lines of code.

Table of Contents

Readability
Versatility
Community Support
Vast Library Ecosystem
Cross-Platform
Simplicity

Advantages of Python

Creator: Guido van Rossum
Year of Origin: 1991
Readability: Python's syntax promotes code readability.
Large Standard Library: Python has a comprehensive standard library.
Versatility: Python can be used for web development, data analysis, and more.
Official Documentation: docs.python.org/3

Table of Contents

Readability
Versatility
Community Support
Vast Library Ecosystem
Cross-Platform
Simplicity

Advantages of Python

Creator: Guido van Rossum
Year of Origin: 1991
Readability: Python's syntax promotes code readability.
Large Standard Library: Python has a comprehensive standard library.
Versatility: Python can be used for web development, data analysis, and more.
Official Documentation: docs.python.org/3

Python Advantages

Introduction

Python is a versatile programming language known for its numerous advantages in various domains. This quick reference provides an overview of the key benefits of using Python.

Readability

Clear and Readable Syntax: Python's syntax emphasizes readability and reduces the cost of program maintenance.
Indentation-based Structure: Python enforces code indentation, making the code structure visually clear and consistent.

Versatility

General-Purpose Language: Python can be used for a wide range of applications, including web development, data analysis, machine learning, and more.
Integration Capabilities: Python easily integrates with other languages and tools, facilitating complex projects.

Community Support

Active Community: Python has a large and active community of developers who provide support, share knowledge, and create open-source libraries.
Documentation and Resources: Abundant documentation and online resources are available for Python users, making it easy to learn and troubleshoot.

Vast Library Ecosystem

Rich Standard Library: Python's standard library includes modules for various tasks, reducing the need for external dependencies.
Third-Party Libraries: Python boasts a vast ecosystem of third-party libraries and frameworks for specific domains, enhancing productivity.

Cross-Platform

Platform Independence: Python code can run on multiple platforms without modification, increasing portability.
Support for Major Operating Systems: Python supports Windows, macOS, Linux, and more.

Simplicity

Minimalistic Syntax: Python's clean and minimalistic syntax reduces the learning curve for beginners.
Expressive and Concise Code: Python allows developers to express complex ideas in fewer lines of code.

Python Philosophy

Introduction

Python is not just a programming language; it also embodies a set of guiding principles and philosophies that shape its design and usage. This quick reference provides an overview of the key philosophies that define Python's ethos.

Zen of Python

The Zen of Python: A collection of guiding aphorisms that capture Python's design philosophy and principles.
Readability Counts: Python code should be easy to read and understand, even by those who didn't write it.

Readability Counts

Readability: Python's emphasis on clean and readable code makes it easier to maintain and collaborate on projects.
Code as Communication: Python code should communicate its purpose clearly, making it accessible to others.

Simple is Better Than Complex

Simplicity: Python values simplicity in design and implementation, favoring straightforward solutions over complex ones.
Minimize Cognitive Load: Simple code reduces cognitive load, making it easier for developers to understand and modify.

Explicit is Better Than Implicit

Clarity: Python encourages explicit code that leaves no room for ambiguity or hidden behavior.
Obviousness: Code should be obvious in its intent and actions, minimizing surprises.

Community and Collaboration

Community: Python's community-driven development fosters collaboration, knowledge sharing, and support.
Open Source: Python's open-source nature promotes transparency, allowing anyone to contribute and improve the language.

Practicality Beats Purity

Real-World Solutions: Python prioritizes practical solutions that address real-world problems over rigid purity.
Flexibility: Python's flexibility allows developers to choose the best tool for the job.

Table of Contents

Zen of Python
Readability Counts
Simple is Better Than Complex
Explicit is Better Than Implicit
Community and Collaboration
Practicality Beats Purity

Philosophy of Python

Creator: Guido van Rossum
Year of Origin: 1991
Design Philosophy: Readability, Simplicity, Explicit is better than implicit
Key Principles: Zen of Python (PEP 20)
Notable Quote: "There should be one-- and preferably only one --obvious way to do it."
Official Python Documentation: docs.python.org/3

Table of Contents

Zen of Python
Readability Counts
Simple is Better Than Complex
Explicit is Better Than Implicit
Community and Collaboration
Practicality Beats Purity

Philosophy of Python

Creator: Guido van Rossum
Year of Origin: 1991
Design Philosophy: Readability, Simplicity, Explicit is better than implicit
Key Principles: Zen of Python (PEP 20)
Notable Quote: "There should be one-- and preferably only one --obvious way to do it."
Official Python Documentation: docs.python.org/3

Python Philosophy

Introduction

Python is not just a programming language; it also embodies a set of guiding principles and philosophies that shape its design and usage. This quick reference provides an overview of the key philosophies that define Python's ethos.

Zen of Python

The Zen of Python: A collection of guiding aphorisms that capture Python's design philosophy and principles.
Readability Counts: Python code should be easy to read and understand, even by those who didn't write it.

Readability Counts

Readability: Python's emphasis on clean and readable code makes it easier to maintain and collaborate on projects.
Code as Communication: Python code should communicate its purpose clearly, making it accessible to others.

Simple is Better Than Complex

Simplicity: Python values simplicity in design and implementation, favoring straightforward solutions over complex ones.
Minimize Cognitive Load: Simple code reduces cognitive load, making it easier for developers to understand and modify.

Explicit is Better Than Implicit

Clarity: Python encourages explicit code that leaves no room for ambiguity or hidden behavior.
Obviousness: Code should be obvious in its intent and actions, minimizing surprises.

Community and Collaboration

Community: Python's community-driven development fosters collaboration, knowledge sharing, and support.
Open Source: Python's open-source nature promotes transparency, allowing anyone to contribute and improve the language.

Practicality Beats Purity

Real-World Solutions: Python prioritizes practical solutions that address real-world problems over rigid purity.
Flexibility: Python's flexibility allows developers to choose the best tool for the job.

Python Examples of Programs

Introduction

Python is a versatile programming language that can be used for a wide range of applications. This quick reference provides examples of Python programs across various domains to showcase its practicality and versatility.

Hello, World!

Hello, World!: The classic introductory program that displays "Hello, World!" on the screen.

```
print("Hello, World!")
```

Web Development

Simple Web Server: Create a basic web server using Python's built-in `http.server` module.

```
import http.server
import socketserver

PORT = 8000

with socketserver.TCPServer(
  ("", PORT),
  http.server.SimpleHTTPRequestHandler
) as httpd:
    print("Serving at port", PORT)
    httpd.serve_forever()
```

Data Analysis

Data Visualization: Use libraries like Matplotlib and Pandas to create interactive data visualizations.

```
import matplotlib.pyplot as plt
import pandas as pd

# Create a simple plot
data = {'x': [1, 2, 3, 4, 5],
  'y': [10, 12, 5, 8, 7]}
df = pd.DataFrame(data)
plt.plot(df['x'], df['y'])
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Simple Data Plot')
plt.show()
```

Machine Learning

Linear Regression: Implement a simple linear regression model using Scikit-Learn.

```
from sklearn import linear_model

# Sample data
X = [[1], [2], [3], [4], [5]]
y = [10, 12, 5, 8, 7]

# Create a linear regression model
model = linear_model.LinearRegression()
model.fit(X, y)

# Predict values
predicted = model.predict([[6]])
print("Predicted:", predicted[0])
```

Automation

File Backup Script: Create a Python script to automate file backups to a specified location.

```
import shutil

source_dir = '/path/to/source'
backup_dir = '/path/to/backup'

shutil.copytree(source_dir, backup_dir)
```

Game Development

Simple Game Using Pygame: Develop a basic game using the Pygame library.

```
import pygame
import sys

# Initialize Pygame
pygame.init()

# Create a game window
screen = pygame.display.set_mode(
  (800, 600)
)
pygame.display.set_caption(
  "Simple Game"
)

# Main game loop
while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()

    pygame.display.update()
```

Table of Contents

Hello, World!
Web Development
Data Analysis
Machine Learning
Automation
Game Development

Apps with Python

Popular Libraries: Django, Flask, NumPy, SciPy, TensorFlow
Common Use Cases: Web Development, Data Analysis, Machine Learning, AI
Notable Python Programs: YouTube, Instagram, Spotify, Reddit
Python Version: Python 3.x
Official Python Documentation: docs.python.org/3

Python Wiki

Table of Contents

Hello, World!
Web Development
Data Analysis
Machine Learning
Automation
Game Development

Apps with Python

Popular Libraries: Django, Flask, NumPy, SciPy, TensorFlow
Common Use Cases: Web Development, Data Analysis, Machine Learning, AI
Notable Python Programs: YouTube, Instagram, Spotify, Reddit
Python Version: Python 3.x
Official Python Documentation: docs.python.org/3

Python Examples of Programs

Introduction

Python is a versatile programming language that can be used for a wide range of applications. This quick reference provides examples of Python programs across various domains to showcase its practicality and versatility.

Hello, World!

Hello, World!: The classic introductory program that displays "Hello, World!" on the screen.

```
print("Hello, World!")
```

Web Development

Simple Web Server: Create a basic web server using Python's built-in `http.server` module.

```
import http.server
import socketserver

PORT = 8000

with socketserver.TCPServer(
  ("", PORT),
  http.server.SimpleHTTPRequestHandler
) as httpd:
    print("Serving at port", PORT)
    httpd.serve_forever()
```

Data Analysis

Data Visualization: Use libraries like Matplotlib and Pandas to create interactive data visualizations.

```
import matplotlib.pyplot as plt
import pandas as pd

# Create a simple plot
data = {'x': [1, 2, 3, 4, 5],
  'y': [10, 12, 5, 8, 7]}
df = pd.DataFrame(data)
plt.plot(df['x'], df['y'])
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Simple Data Plot')
plt.show()
```

Machine Learning

Linear Regression: Implement a simple linear regression model using Scikit-Learn.

```
from sklearn import linear_model

# Sample data
X = [[1], [2], [3], [4], [5]]
y = [10, 12, 5, 8, 7]

# Create a linear regression model
model = linear_model.LinearRegression()
model.fit(X, y)

# Predict values
predicted = model.predict([[6]])
print("Predicted:", predicted[0])
```

Automation

File Backup Script: Create a Python script to automate file backups to a specified location.

```
import shutil

source_dir = '/path/to/source'
backup_dir = '/path/to/backup'

shutil.copytree(source_dir, backup_dir)
```

Game Development

Simple Game Using Pygame: Develop a basic game using the Pygame library.

```
import pygame
import sys

# Initialize Pygame
pygame.init()

# Create a game window
screen = pygame.display.set_mode(
  (800, 600)
)
pygame.display.set_caption(
  "Simple Game"
)

# Main game loop
while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()

    pygame.display.update()
```

Код

```
<!-- advantages.html -->

<!DOCTYPE html>
<html lang="en" class="scroll-smooth">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"
/>
    <title>Advantages - Python Wiki</title>
    <link rel="stylesheet" href="./css/base.css" />
    <link rel="stylesheet" href="./css/classes.css" />
    <link rel="stylesheet" href="./css/custom.css" />
    <link rel="shortcut icon" href="./img/snake.png" type="image/x-icon"
/>
  </head>
  <body>
    <div class="min-h-screen w-full bg-slate-100 text-slate-900">
      <div class="mx-auto grid max-w-6xl gap-4 p-4">
        <header
          class="flex items-center justify-between rounded-md bg-white p-
4"
        >
          <a href="./index.html" class="text-lg font-medium">Python
Wiki</a>
          <a href="./lab3/index.html" title="View resume">
            <img
              src="./img/cv.png"
              alt="resume logo image link"
              class="h-6 w-6"
            />
          </a>
        </header>

        <div class="bg-white rounded-md p-4 flex items-center gap-2
md:gap-4">
          <a href="./index.html">Overview</a>
          <a href="./history.html">History</a>
          <a href="./advantages.html" class="font-medium">Advantages</a>
          <a href="./philosophy.html">Philosophy</a>
          <a href="./examples.html">Apps</a>
        </div>

        <section class="grid gap-4 md:flex md:flex-row-reverse">
          <div class="grid gap-4 md:flex md:flex-col md:w-[20rem]">
            <nav class="grid rounded-md bg-white p-4">
```

```
            <h2 class="pb-2">Table of Contents</h2>
            <ol>
              <li><a href="#readability">Readability</a></li>
              <li><a href="#versatility">Versatility</a></li>
              <li><a href="#community-support">Community
Support</a></li>
              <li>
                <a href="#vast-library-ecosystem">Vast Library
Ecosystem</a>
              </li>
              <li><a href="#cross-platform">Cross-Platform</a></li>
              <li><a href="#simplicity">Simplicity</a></li>
            </ol>
          </nav>
          <aside class="grid rounded-md bg-white p-4">
            <div class="flex items-center gap-2 pb-3">
              <img
                src="./img/snake.png"
                alt="python logo image"
                class="h-6 w-6"
              />
              <h2 class="text-xl font-medium">Advantages of Python</h2>
            </div>

            <ul>
              <li><strong>Creator:</strong> Guido van Rossum</li>
              <li><strong>Year of Origin:</strong> 1991</li>
              <li>
                <strong>Readability:</strong> Python's syntax promotes
code
                readability.
              </li>
              <li>
                <strong>Large Standard Library:</strong> Python has a
                comprehensive standard library.
              </li>
              <li>
                <strong>Versatility:</strong> Python can be used for
web
                development, data analysis, and more.
              </li>
              <li>
                <strong>Official Documentation:</strong>
                <a
href="https://docs.python.org/3/">docs.python.org/3</a>
              </li>
            </ul>
          </aside>
```

```
          </div>

          <main class="grid rounded-md bg-white p-4 flex-1">
            <h1>Python Advantages</h1>

            <h2 class="first-heading">Introduction</h2>
            <p>
              Python is a versatile programming language known for its
numerous
              advantages in various domains. This quick reference
provides an
              overview of the key benefits of using Python.
            </p>

            <h2 id="readability">Readability</h2>
            <p>
              <strong>Clear and Readable Syntax</strong>: Python's syntax
              emphasizes readability and reduces the cost of program
              maintenance.
            </p>
            <p>
              <strong>Indentation-based Structure</strong>: Python
enforces code
              indentation, making the code structure visually clear and
              consistent.
            </p>

            <h2 id="versatility">Versatility</h2>
            <p>
              <strong>General-Purpose Language</strong>: Python can be
used for
              a wide range of applications, including web development,
data
              analysis, machine learning, and more.
            </p>
            <p>
              <strong>Integration Capabilities</strong>: Python easily
              integrates with other languages and tools, facilitating
complex
              projects.
            </p>

            <h2 id="community-support">Community Support</h2>
            <p>
              <strong>Active Community</strong>: Python has a large and
active
              community of developers who provide support, share
knowledge, and
```

```
              create open-source libraries.
            </p>
            <p>
              <strong>Documentation and Resources</strong>: Abundant
              documentation and online resources are available for Python
users,
              making it easy to learn and troubleshoot.
            </p>

            <h2 id="vast-library-ecosystem">Vast Library Ecosystem</h2>
            <p>
              <strong>Rich Standard Library</strong>: Python's standard
library
              includes modules for various tasks, reducing the need for
external
              dependencies.
            </p>
            <p>
              <strong>Third-Party Libraries</strong>: Python boasts a
vast
              ecosystem of third-party libraries and frameworks for
specific
              domains, enhancing productivity.
            </p>

            <h2 id="cross-platform">Cross-Platform</h2>
            <p>
              <strong>Platform Independence</strong>: Python code can run
on
              multiple platforms without modification, increasing
portability.
            </p>
            <p>
              <strong>Support for Major Operating Systems</strong>:
Python
              supports Windows, macOS, Linux, and more.
            </p>

            <h2 id="simplicity">Simplicity</h2>
            <p>
              <strong>Minimalistic Syntax</strong>: Python's clean and
              minimalistic syntax reduces the learning curve for
beginners.
            </p>
            <p>
              <strong>Expressive and Concise Code</strong>: Python allows
              developers to express complex ideas in fewer lines of code.
            </p>
```

```
          </main>
        </section>

        <footer class="grid rounded-md bg-white p-4">
          <a href="./lab3/index.html">Onyshchenko - Resume Link</a>
        </footer>
      </div>
    </div>
  </body>
</html>
```

```
<!-- examples.html -->

<!DOCTYPE html>
<html lang="en" class="scroll-smooth">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"
/>
    <title>App Examples - Python Wiki</title>
    <link rel="stylesheet" href="./css/base.css" />
    <link rel="stylesheet" href="./css/classes.css" />
    <link rel="stylesheet" href="./css/custom.css" />
    <link rel="shortcut icon" href="./img/snake.png" type="image/x-icon"
/>
  </head>
  <body>
    <div class="min-h-screen w-full bg-slate-100 text-slate-900">
      <div class="mx-auto grid max-w-6xl gap-4 p-4">
        <header
          class="flex items-center justify-between rounded-md bg-white p-
4"
        >
          <a href="./index.html" class="text-lg font-medium">Python
Wiki</a>
          <a href="./lab3/index.html" title="View resume">
            <img
              src="./img/cv.png"
              alt="resume logo image link"
              class="h-6 w-6"
            />
          </a>
        </header>

        <div class="bg-white rounded-md p-4 flex items-center gap-2
md:gap-4">
          <a href="./index.html">Overview</a>
```

```
          <a href="./history.html">History</a>
          <a href="./advantages.html">Advantages</a>
          <a href="./philosophy.html">Philosophy</a>
          <a href="./examples.html" class="font-medium">Apps</a>
        </div>

        <section class="grid gap-4 md:flex md:flex-row-reverse">
          <div class="grid gap-4 md:flex md:flex-col md:w-[20rem]">
            <nav class="grid rounded-md bg-white p-4">
              <h2 class="pb-2">Table of Contents</h2>
              <ol>
                <li><a href="#hello-world">Hello, World!</a></li>
                <li><a href="#web-development">Web Development</a></li>
                <li><a href="#data-analysis">Data Analysis</a></li>
                <li><a href="#machine-learning">Machine Learning</a></li>
                <li><a href="#automation">Automation</a></li>
                <li><a href="#game-development">Game Development</a></li>
              </ol>
            </nav>

            <aside class="grid rounded-md bg-white p-4">
              <div class="flex items-center gap-2 pb-3">
                <img
                  src="./img/snake.png"
                  alt="python logo image"
                  class="h-6 w-6"
                />
                <h2 class="text-xl font-medium">Apps with Python</h2>
              </div>

              <ul>
                <li>
                  <strong>Popular Libraries:</strong> Django, Flask,
NumPy,
                  SciPy, TensorFlow
                </li>
                <li>
                  <strong>Common Use Cases:</strong> Web Development,
Data
                  Analysis, Machine Learning, AI
                </li>
                <li>
                  <strong>Notable Python Programs:</strong> YouTube,
Instagram,
                  Spotify, Reddit
                </li>
                <li><strong>Python Version:</strong> Python 3.x</li>
                <li>
```

```
                <strong>Official Python Documentation:</strong>
                <a
href="https://docs.python.org/3/">docs.python.org/3</a>
              </li>
            </ul>
          </aside>
        </div>

        <main class="grid rounded-md bg-white p-4 flex-1">
          <h1>Python Examples of Programs</h1>

          <h2 class="first-heading">Introduction</h2>
          <p>
            Python is a versatile programming language that can be used
for a
            wide range of applications. This quick reference provides
examples
            of Python programs across various domains to showcase its
            practicality and versatility.
          </p>

          <h2 id="hello-world">Hello, World!</h2>
          <p>
            <strong>Hello, World!</strong>: The classic introductory
program
            that displays "Hello, World!" on the screen.
          </p>
          <code>
            <pre>
print("Hello, World!")
        </pre
            >
          </code>

          <h2 id="web-development">Web Development</h2>
          <p>
            <strong>Simple Web Server</strong>: Create a basic web
server
            using Python's built-in <code>http.server</code> module.
          </p>
          <code>
            <pre>
import http.server
import socketserver

PORT = 8000

with socketserver.TCPServer(
```

```
  ("", PORT),
  http.server.SimpleHTTPRequestHandler
) as httpd:
    print("Serving at port", PORT)
    httpd.serve_forever()
        </pre
              >
            </code>

            <h2 id="data-analysis">Data Analysis</h2>
            <p>
              <strong>Data Visualization</strong>: Use libraries like
Matplotlib
              and Pandas to create interactive data visualizations.
            </p>
            <code>
              <pre>
import matplotlib.pyplot as plt
import pandas as pd

# Create a simple plot
data = {'x': [1, 2, 3, 4, 5],
  'y': [10, 12, 5, 8, 7]}
df = pd.DataFrame(data)
plt.plot(df['x'], df['y'])
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Simple Data Plot')
plt.show()
        </pre
              >
            </code>

            <h2 id="machine-learning">Machine Learning</h2>
            <p>
              <strong>Linear Regression</strong>: Implement a simple
linear
              regression model using Scikit-Learn.
            </p>
            <code>
              <pre>
from sklearn import linear_model

# Sample data
X = [[1], [2], [3], [4], [5]]
y = [10, 12, 5, 8, 7]

# Create a linear regression model
```

```
model = linear_model.LinearRegression()
model.fit(X, y)

# Predict values
predicted = model.predict([[6]])
print("Predicted:", predicted[0])
        </pre
              >
            </code>

            <h2 id="automation">Automation</h2>
            <p>
              <strong>File Backup Script</strong>: Create a Python script
to
              automate file backups to a specified location.
            </p>
            <code>
              <pre>
import shutil

source_dir = '/path/to/source'
backup_dir = '/path/to/backup'

shutil.copytree(source_dir, backup_dir)
        </pre
              >
            </code>

            <h2 id="game-development">Game Development</h2>
            <p>
              <strong>Simple Game Using Pygame</strong>: Develop a basic
game
              using the Pygame library.
            </p>
            <code>
              <pre>
import pygame
import sys

# Initialize Pygame
pygame.init()

# Create a game window
screen = pygame.display.set_mode(
  (800, 600)
)
pygame.display.set_caption(
  "Simple Game"
```

```
)

# Main game loop
while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()

    pygame.display.update()
        </pre
              >
            </code>
          </main>
        </section>

        <footer class="grid rounded-md bg-white p-4">
          <a href="./lab3/index.html">Onyshchenko - Resume Link</a>
        </footer>
      </div>
    </div>
  </body>
</html>
```

```
<!-- history.html -->

<!DOCTYPE html>
<html lang="en" class="scroll-smooth">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"
/>
    <title>History - Python Wiki</title>
    <link rel="stylesheet" href="./css/base.css" />
    <link rel="stylesheet" href="./css/classes.css" />
    <link rel="stylesheet" href="./css/custom.css" />
    <link rel="shortcut icon" href="./img/snake.png" type="image/x-icon"
/>
  </head>
  <body>
    <div class="min-h-screen w-full bg-slate-100 text-slate-900">
      <div class="mx-auto grid max-w-6xl gap-4 p-4">
        <header
          class="flex items-center justify-between rounded-md bg-white p-
4"
        >
```

```
          <a href="./index.html" class="text-lg font-medium">Python
Wiki</a>
          <a href="./lab3/index.html" title="View resume">
            <img
              src="./img/cv.png"
              alt="resume logo image link"
              class="h-6 w-6"
            />
          </a>
        </header>

        <div class="bg-white rounded-md p-4 flex items-center gap-2
md:gap-4">
          <a href="./index.html">Overview</a>
          <a href="./history.html" class="font-medium">History</a>
          <a href="./advantages.html">Advantages</a>
          <a href="./philosophy.html">Philosophy</a>
          <a href="./examples.html">Apps</a>
        </div>

        <section
          class="grid gap-4 md:flex md:flex-row-reverse flex-row-reverse"
        >
          <div class="grid gap-4 md:flex md:flex-col md:w-[20rem]">
            <nav class="grid rounded-md bg-white p-4">
              <h2 class="pb-2">Table of Contents</h2>
              <ol>
                <li><a href="#inception">Inception</a></li>
                <li><a href="#major-versions">Major Versions</a></li>
                <li>
                  <a href="#python-enhancement-proposals-pep"
                    >Python Enhancement Proposals (PEP)</a
                  >
                </li>
                <li>
                  <a href="#notable-contributors">Notable
Contributors</a>
                </li>
                <li>
                  <a href="#community-and-growth">Community and
Growth</a>
                </li>
              </ol>
            </nav>

            <aside class="grid rounded-md bg-white p-4">
              <div class="flex items-center gap-2 pb-3">
                <img
```

```
                src="./img/snake.png"
                alt="python logo image"
                class="h-6 w-6"
              />
              <h2 class="text-xl font-medium">The Journey of
Python</h2>
            </div>

            <ul>
              <li><strong>Creator:</strong> Guido van Rossum</li>
              <li><strong>Year of Origin:</strong> 1991</li>
              <li>
                <strong>Programming Paradigms:</strong> Object-
oriented,
                Procedural, Functional
              </li>
              <li>
                <strong>Type System:</strong> Duck-typing, Dynamic,
Strong
                typing
              </li>
              <li>
                <strong>Official Website:</strong>
                <a href="https://www.python.org/">www.python.org</a>
              </li>
            </ul>
          </aside>
        </div>

        <main class="grid rounded-md bg-white p-4 flex-1">
          <h1 class="text-2xl font-semibold pb-4">Python History</h1>

          <h2 class="first-heading">Introduction</h2>
          <p>
            Python is a widely-used high-level programming language
with a
            rich history. This quick reference provides an overview of
key
            milestones and events in the development of Python.
          </p>

          <h2 id="inception">Inception</h2>
          <p>
            <strong>Creation of Python</strong>: Python was created by
Guido
            van Rossum and first released in 1991.
          </p>
          <p>
```

```
              <strong>Motivation</strong>: Guido aimed to create a
language that
              emphasized code readability and allowed programmers to
express
              concepts in fewer lines of code.
            </p>

            <h2 id="major-versions">Major Versions</h2>
            <p>
              <strong>Python 2.x</strong>: The Python 2 series (e.g.,
2.7) was
              widely used for many years but officially reached end-of-
life in
              2020.
            </p>
            <p>
              <strong>Python 3.x</strong>: Python 3 introduced
significant
              changes to improve consistency and eliminate some of the
              inconsistencies present in Python 2.
            </p>

            <h2 id="python-enhancement-proposals-pep">
              Python Enhancement Proposals (PEP)
            </h2>
            <p>
              <strong>PEP 8</strong>: This style guide for Python code
has been
              influential in shaping Python's code formatting
conventions.
            </p>
            <p>
              <strong>PEP 20 (Zen of Python)</strong>: A collection of
guiding
              principles for writing computer programs in Python,
providing
              insights into Python's design philosophy.
            </p>

            <h2 id="notable-contributors">Notable Contributors</h2>
            <p>
              <strong>Guido van Rossum</strong>: Python's creator and
              "Benevolent Dictator For Life" (BDFL) until he stepped down
in
              2018.
            </p>
            <p>
```

```
              <strong>Python Software Foundation (PSF)</strong>:
Established in
              2001 to promote, protect, and advance Python and its
community.
            </p>

            <h2 id="community-and-growth">Community and Growth</h2>
            <p>
              <strong>Python Community</strong>: Python has a vibrant and
              welcoming community of developers, with conferences and
user
              groups worldwide.
            </p>
            <p>
              <strong>Python's Popularity</strong>: Python's simplicity
and
              versatility have contributed to its popularity in various
fields,
              including web development, data science, and artificial
              intelligence.
            </p>
          </main>
        </section>

        <footer class="grid rounded-md bg-white p-4">
          <a href="./lab3/index.html">Onyshchenko - Resume Link</a>
        </footer>
      </div>
    </div>
  </body>
</html>
```

```
<!-- index.html -->

<!DOCTYPE html>
<html lang="en" class="scroll-smooth">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"
/>
    <title>Overview - Python Wiki</title>
    <link rel="stylesheet" href="./css/base.css" />
    <link rel="stylesheet" href="./css/classes.css" />
    <link rel="stylesheet" href="./css/custom.css" />
    <link rel="shortcut icon" href="./img/snake.png" type="image/x-icon"
/>
  </head>
```

```
  <body>
    <div class="min-h-screen w-full bg-slate-100 text-slate-900">
      <div class="mx-auto grid max-w-6xl gap-4 p-4">
        <header
          class="flex items-center justify-between rounded-md bg-white p-
4"
        >
          <a href="./index.html" class="text-lg font-medium">Python
Wiki</a>
          <a href="./lab3/index.html" title="View resume">
            <img
              src="./img/cv.png"
              alt="resume logo image link"
              class="h-6 w-6"
            />
          </a>
        </header>

        <div class="bg-white rounded-md p-4 flex items-center gap-2
md:gap-4">
          <a href="./index.html" class="font-medium">Overview</a>
          <a href="./history.html">History</a>
          <a href="./advantages.html">Advantages</a>
          <a href="./philosophy.html">Philosophy</a>
          <a href="./examples.html">Apps</a>
        </div>

        <section class="grid gap-4 md:flex md:flex-row-reverse">
          <div class="grid gap-4 md:flex md:flex-col md:w-[20rem]">
            <nav class="grid rounded-md bg-white p-4">
              <h2 class="pb-2">Table of Contents</h2>

              <ol>
                <li>
                  <a href="#variables-and-data-types"
                    >Variables and Data Types</a
                  >
                </li>
                <li>
                  <a href="#conditional-statements">Conditional
Statements</a>
                </li>
                <li><a href="#loops">Loops</a></li>
                <li><a href="#functions">Functions</a></li>
                <li><a href="#lists">Lists</a></li>
                <li><a href="#dictionaries">Dictionaries</a></li>
                <li><a href="#strings">Strings</a></li>
                <li><a href="#file-handling">File Handling</a></li>
```

```
            </ol>
          </nav>

          <aside class="grid rounded-md bg-white p-4">
            <div class="flex items-center gap-2 pb-3">
              <img
                src="./img/snake.png"
                alt="python logo image"
                class="h-6 w-6"
              />
              <h2 class="text-xl font-medium">Python Lanugage</h2>
            </div>

            <ul>
              <li><strong>Designed by:</strong> Guido van Rossum</li>
              <li><strong>First appeared:</strong> 1991</li>
              <li>
                <strong>Paradigms:</strong> Object-oriented,
procedural,
                functional
              </li>
              <li>
                <strong>Typing discipline:</strong> Duck, dynamic,
strong
                typing
              </li>
              <li>
                <strong>Website:</strong>
                <a href="https://www.python.org/">www.python.org</a>
              </li>
            </ul>
          </aside>
        </div>

        <main class="grid rounded-md bg-white p-4 flex-1">
          <h1 class="pb-4 text-2xl font-semibold">Python Overview</h1>

          <h2 class="first-heading">Introduction</h2>

          <p>
            Python is a high-level, versatile programming language
known for
            its simplicity and readability. This quick reference
provides an
            overview of essential Python concepts and syntax.
          </p>
```

```
            <h2 id="variables-and-data-types">Variables and Data
Types</h2>

            <ul>
              <li>
                <strong>Variables:</strong> In Python, you can assign
values to
                variables using the <code>=</code> operator. Example:
                <code>x = 10</code>.
              </li>
              <li>
                <strong>Data Types:</strong> Python supports various data
types,
                including integers, floats, strings, booleans, and more.
              </li>
              <li>
                <strong>Type Conversion:</strong> You can convert between
data
                types using functions like <code>int()</code>,
                <code>float()</code>, and <code>str()</code>.
              </li>
            </ul>

            <h2 id="conditional-statements">Conditional Statements</h2>

            <ul>
              <li>
                <strong>if Statement:</strong> Use <code>if</code> to
execute
                code conditionally based on a Boolean expression.
              </li>
              <li>
                <strong>elif and else:</strong> Extend
                <code>if</code> statements with <code>elif</code> (else
if) and
                <code>else</code> for multiple branching.
              </li>
            </ul>

            <h2 id="loops">Loops</h2>

            <ul>
              <li>
                <strong>for Loop:</strong> Iterate over a sequence (e.g.,
list,
                string) using a <code>for</code> loop.
              </li>
              <li>
```

```
                <strong>while Loop:</strong> Execute code repeatedly as
long as
                a condition is true with a <code>while</code> loop.
              </li>
            </ul>

            <h2 id="functions">Functions</h2>

            <ul>
              <li>
                <strong>Defining Functions:</strong> Create reusable code
blocks
                using the <code>def</code> keyword.
              </li>
              <li>
                <strong>Parameters:</strong> Pass arguments to functions,
which
                can be optional or required.
              </li>
              <li>
                <strong>Return Values:</strong> Functions can return
values
                using the <code>return</code> statement.
              </li>
            </ul>

            <h2 id="lists">Lists</h2>

            <ul>
              <li>
                <strong>Creating Lists:</strong> Define lists using
square
                brackets, e.g., <code>my_list = [1, 2, 3]</code>.
              </li>
              <li>
                <strong>Indexing:</strong> Access list elements using
their
                index (zero-based).
              </li>
              <li>
                <strong>Slicing:</strong> Extract portions of a list
using
                slicing, e.g., <code>my_list[1:3]</code>.
              </li>
            </ul>

            <h2 id="dictionaries">Dictionaries</h2>
```

```
            <ul>
              <li>
                <strong>Creating Dictionaries:</strong> Define key-value
pairs
                using curly braces, e.g.,
                <code>my_dict = {'name': 'John', 'age': 30}</code>.
              </li>
              <li>
                <strong>Accessing Values:</strong> Retrieve values by
specifying
                the key.
              </li>
            </ul>

            <h2 id="strings">Strings</h2>

            <ul>
              <li>
                <strong>Creating Strings:</strong> Define strings with
single or
                double quotes, e.g., <code>"Hello, World!"</code>.
              </li>
              <li>
                <strong>String Methods:</strong> Use built-in string
methods for
                operations like concatenation and splitting.
              </li>
            </ul>

            <h2 id="file-handling">File Handling</h2>

            <ul>
              <li>
                <strong>Opening Files:</strong> Use <code>open()</code>
to open
                files for reading or writing.
              </li>
              <li>
                <strong>Reading and Writing:</strong> Read file contents
with
                <code>read()</code>, write data with
<code>write()</code>.
              </li>
            </ul>

            <h3 id="read-more" class="pb-2 font-medium border-t-2 pt-2
mt-6">
              Read more
```

```
            </h3>

            <ul>
              <li><a href="./history.html">History</a></li>
              <li><a href="./advantages.html">Advantages</a></li>
              <li><a href="./philosophy.html">Philosophy</a></li>
              <li><a href="./examples.html">Examples of programs</a></li>
            </ul>
          </main>
        </section>

        <footer class="grid rounded-md bg-white p-4">
          <a href="./lab3/index.html">Onyshchenko - Resume Link</a>
        </footer>
      </div>
    </div>
  </body>
</html>
```

```
<!-- philosophy.html -->

<!DOCTYPE html>
<html lang="en" class="scroll-smooth">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"
/>
    <title>Philosophy - Python Wiki</title>
    <link rel="stylesheet" href="./css/base.css" />
    <link rel="stylesheet" href="./css/classes.css" />
    <link rel="stylesheet" href="./css/custom.css" />
    <link rel="shortcut icon" href="./img/snake.png" type="image/x-icon"
/>
  </head>
  <body>
    <div class="min-h-screen w-full bg-slate-100 text-slate-900">
      <div class="mx-auto grid max-w-6xl gap-4 p-4">
        <header
          class="flex items-center justify-between rounded-md bg-white p-
4"
        >
          <a href="./index.html" class="text-lg font-medium">Python
Wiki</a>
          <a href="./lab3/index.html" title="View resume">
            <img
              src="./img/cv.png"
              alt="resume logo image link"
```

```
              class="h-6 w-6"
            />
          </a>
        </header>

        <div class="bg-white rounded-md p-4 flex items-center gap-2
md:gap-4">
          <a href="./index.html">Overview</a>
          <a href="./history.html">History</a>
          <a href="./advantages.html">Advantages</a>
          <a href="./philosophy.html" class="font-medium">Philosophy</a>
          <a href="./examples.html">Apps</a>
        </div>

        <section class="grid gap-4 md:flex md:flex-row-reverse">
          <div class="grid gap-4 md:flex md:flex-col md:w-[20rem]">
            <nav class="grid rounded-md bg-white p-4">
              <h2 class="pb-2">Table of Contents</h2>
              <ol>
                <li><a href="#zen-of-python">Zen of Python</a></li>
                <li><a href="#readability-counts">Readability
Counts</a></li>
                <li>
                  <a href="#simple-is-better-than-complex"
                    >Simple is Better Than Complex</a
                  >
                </li>
                <li>
                  <a href="#explicit-is-better-than-implicit"
                    >Explicit is Better Than Implicit</a
                  >
                </li>
                <li>
                  <a href="#community-and-collaboration"
                    >Community and Collaboration</a
                  >
                </li>
                <li>
                  <a href="#practicality-beats-purity"
                    >Practicality Beats Purity</a
                  >
                </li>
              </ol>
            </nav>

            <aside class="grid rounded-md bg-white p-4">
              <div class="flex items-center gap-2 pb-3">
                <img
```

```
                src="./img/snake.png"
                alt="python logo image"
                class="h-6 w-6"
              />
              <h2 class="text-xl font-medium">Philosophy of Python</h2>
            </div>

            <ul>
              <li><strong>Creator:</strong> Guido van Rossum</li>
              <li><strong>Year of Origin:</strong> 1991</li>
              <li>
                <strong>Design Philosophy:</strong> Readability,
Simplicity,
                Explicit is better than implicit
              </li>
              <li><strong>Key Principles:</strong> Zen of Python (PEP
20)</li>
              <li>
                <strong>Notable Quote:</strong> "There should be one--
and
                preferably only one --obvious way to do it."
              </li>
              <li>
                <strong>Official Python Documentation:</strong>
                <a
href="https://docs.python.org/3/">docs.python.org/3</a>
              </li>
            </ul>
          </aside>
        </div>

        <main class="grid rounded-md bg-white p-4 flex-1">
          <h1>Python Philosophy</h1>

          <h2 class="first-heading">Introduction</h2>
          <p>
            Python is not just a programming language; it also embodies
a set
            of guiding principles and philosophies that shape its
design and
            usage. This quick reference provides an overview of the key
            philosophies that define Python's ethos.
          </p>

          <h2 id="zen-of-python">Zen of Python</h2>
          <p>
            <strong>The Zen of Python</strong>: A collection of guiding
```

```
              aphorisms that capture Python's design philosophy and
principles.
            </p>
            <p>
              <strong>Readability Counts</strong>: Python code should be
easy to
              read and understand, even by those who didn't write it.
            </p>

            <h2 id="readability-counts">Readability Counts</h2>
            <p>
              <strong>Readability</strong>: Python's emphasis on clean
and
              readable code makes it easier to maintain and collaborate
on
              projects.
            </p>
            <p>
              <strong>Code as Communication</strong>: Python code should
              communicate its purpose clearly, making it accessible to
others.
            </p>

            <h2 id="simple-is-better-than-complex">
              Simple is Better Than Complex
            </h2>
            <p>
              <strong>Simplicity</strong>: Python values simplicity in
design
              and implementation, favoring straightforward solutions over
              complex ones.
            </p>
            <p>
              <strong>Minimize Cognitive Load</strong>: Simple code
reduces
              cognitive load, making it easier for developers to
understand and
              modify.
            </p>

            <h2 id="explicit-is-better-than-implicit">
              Explicit is Better Than Implicit
            </h2>
            <p>
              <strong>Clarity</strong>: Python encourages explicit code
that
              leaves no room for ambiguity or hidden behavior.
            </p>
```

```
            <p>
              <strong>Obviousness</strong>: Code should be obvious in its
intent
              and actions, minimizing surprises.
            </p>

            <h2 id="community-and-collaboration">
              Community and Collaboration
            </h2>
            <p>
              <strong>Community</strong>: Python's community-driven
development
              fosters collaboration, knowledge sharing, and support.
            </p>
            <p>
              <strong>Open Source</strong>: Python's open-source nature
promotes
              transparency, allowing anyone to contribute and improve the
              language.
            </p>

            <h2 id="practicality-beats-purity">Practicality Beats
Purity</h2>
            <p>
              <strong>Real-World Solutions</strong>: Python prioritizes
              practical solutions that address real-world problems over
rigid
              purity.
            </p>
            <p>
              <strong>Flexibility</strong>: Python's flexibility allows
              developers to choose the best tool for the job.
            </p>
          </main>
        </section>

        <footer class="grid rounded-md bg-white p-4">
          <a href="./lab3/index.html">Onyshchenko - Resume Link</a>
        </footer>
      </div>
    </div>
  </body>
</html>
```

```
/* base.css */

*,
```

```css
::before,
::after {
  box-sizing: border-box;
  border-width: 0;
  border-style: solid;
  border-color: #e5e7eb;
}

::before,
::after {
  --tw-content: "";
}

html {
  line-height: 1.5;
  -webkit-text-size-adjust: 100%;
  -moz-tab-size: 4;
  tab-size: 4;
  font-family: ui-sans-serif, system-ui, -apple-system,
BlinkMacSystemFont,
    "Segoe UI", Roboto, "Helvetica Neue", Arial, "Noto Sans", sans-serif,
    "Apple Color Emoji", "Segoe UI Emoji", "Segoe UI Symbol", "Noto Color
Emoji";
  font-feature-settings: normal;
  font-variation-settings: normal;
}

body {
  margin: 0;
  line-height: inherit;
}

hr {
  height: 0;
  color: inherit;
  border-top-width: 1px;
}

abbr:where([title]) {
  -webkit-text-decoration: underline dotted;
  text-decoration: underline dotted;
}

h1,
h2,
h3,
h4,
h5,
```

```css
h6 {
  font-size: inherit;
  font-weight: inherit;
}

a {
  color: inherit;
  text-decoration: inherit;
}

b,
strong {
  font-weight: bolder;
}

code,
kbd,
samp,
pre {
  font-family: ui-monospace, SFMono-Regular, Menlo, Monaco, Consolas,
    "Liberation Mono", "Courier New", monospace;
  font-size: 1em;
}

small {
  font-size: 80%;
}

sub,
sup {
  font-size: 75%;
  line-height: 0;
  position: relative;
  vertical-align: baseline;
}

sub {
  bottom: -0.25em;
}

sup {
  top: -0.5em;
}

table {
  text-indent: 0;
  border-color: inherit;
  border-collapse: collapse;
```

```
}

button,
input,
optgroup,
select,
textarea {
  font-family: inherit;
  font-feature-settings: inherit;
  font-variation-settings: inherit;
  font-size: 100%;
  font-weight: inherit;
  line-height: inherit;
  color: inherit;
  margin: 0;
  padding: 0;
}

button,
select {
  text-transform: none;
}

button,
[type="button"],
[type="reset"],
[type="submit"] {
  -webkit-appearance: button;
  background-color: transparent;
  background-image: none;
}

:-moz-focusring {
  outline: auto;
}

:-moz-ui-invalid {
  box-shadow: none;
}

progress {
  vertical-align: baseline;
}

::-webkit-inner-spin-button,
::-webkit-outer-spin-button {
  height: auto;
}
```

```css
[type="search"] {
  -webkit-appearance: textfield;
  outline-offset: -2px;
}

::-webkit-search-decoration {
  -webkit-appearance: none;
}

::-webkit-file-upload-button {
  -webkit-appearance: button;
  font: inherit;
}

summary {
  display: list-item;
}

blockquote,
dl,
dd,
h1,
h2,
h3,
h4,
h5,
h6,
hr,
figure,
p,
pre {
  margin: 0;
}

fieldset {
  margin: 0;
  padding: 0;
}

legend {
  padding: 0;
}

ol,
ul,
menu {
  list-style: none;
```

```
  margin: 0;
  padding: 0;
}

dialog {
  padding: 0;
}

textarea {
  resize: vertical;
}

input::placeholder,
textarea::placeholder {
  opacity: 1;
  color: #9ca3af;
}

button,
[role="button"] {
  cursor: pointer;
}

:disabled {
  cursor: default;
}

img,
svg,
video,
canvas,
audio,
iframe,
embed,
object {
  display: block;
  vertical-align: middle;
}

img,
video {
  max-width: 100%;
  height: auto;
}

[hidden] {
  display: none;
}
```

```
*,
::before,
::after {
  --tw-border-spacing-x: 0;
  --tw-border-spacing-y: 0;
  --tw-translate-x: 0;
  --tw-translate-y: 0;
  --tw-rotate: 0;
  --tw-skew-x: 0;
  --tw-skew-y: 0;
  --tw-scale-x: 1;
  --tw-scale-y: 1;
  --tw-pan-x: ;
  --tw-pan-y: ;
  --tw-pinch-zoom: ;
  --tw-scroll-snap-strictness: proximity;
  --tw-gradient-from-position: ;
  --tw-gradient-via-position: ;
  --tw-gradient-to-position: ;
  --tw-ordinal: ;
  --tw-slashed-zero: ;
  --tw-numeric-figure: ;
  --tw-numeric-spacing: ;
  --tw-numeric-fraction: ;
  --tw-ring-inset: ;
  --tw-ring-offset-width: 0px;
  --tw-ring-offset-color: #fff;
  --tw-ring-color: rgb(59 130 246 / 0.5);
  --tw-ring-offset-shadow: 0 0 #0000;
  --tw-ring-shadow: 0 0 #0000;
  --tw-shadow: 0 0 #0000;
  --tw-shadow-colored: 0 0 #0000;
  --tw-blur: ;
  --tw-brightness: ;
  --tw-contrast: ;
  --tw-grayscale: ;
  --tw-hue-rotate: ;
  --tw-invert: ;
  --tw-saturate: ;
  --tw-sepia: ;
  --tw-drop-shadow: ;
  --tw-backdrop-blur: ;
  --tw-backdrop-brightness: ;
  --tw-backdrop-contrast: ;
  --tw-backdrop-grayscale: ;
  --tw-backdrop-hue-rotate: ;
  --tw-backdrop-invert: ;
```

```
  --tw-backdrop-opacity: ;
  --tw-backdrop-saturate: ;
  --tw-backdrop-sepia: ;
}

::backdrop {
  --tw-border-spacing-x: 0;
  --tw-border-spacing-y: 0;
  --tw-translate-x: 0;
  --tw-translate-y: 0;
  --tw-rotate: 0;
  --tw-skew-x: 0;
  --tw-skew-y: 0;
  --tw-scale-x: 1;
  --tw-scale-y: 1;
  --tw-pan-x: ;
  --tw-pan-y: ;
  --tw-pinch-zoom: ;
  --tw-scroll-snap-strictness: proximity;
  --tw-gradient-from-position: ;
  --tw-gradient-via-position: ;
  --tw-gradient-to-position: ;
  --tw-ordinal: ;
  --tw-slashed-zero: ;
  --tw-numeric-figure: ;
  --tw-numeric-spacing: ;
  --tw-numeric-fraction: ;
  --tw-ring-inset: ;
  --tw-ring-offset-width: 0px;
  --tw-ring-offset-color: #fff;
  --tw-ring-color: rgb(59 130 246 / 0.5);
  --tw-ring-offset-shadow: 0 0 #0000;
  --tw-ring-shadow: 0 0 #0000;
  --tw-shadow: 0 0 #0000;
  --tw-shadow-colored: 0 0 #0000;
  --tw-blur: ;
  --tw-brightness: ;
  --tw-contrast: ;
  --tw-grayscale: ;
  --tw-hue-rotate: ;
  --tw-invert: ;
  --tw-saturate: ;
  --tw-sepia: ;
  --tw-drop-shadow: ;
  --tw-backdrop-blur: ;
  --tw-backdrop-brightness: ;
  --tw-backdrop-contrast: ;
  --tw-backdrop-grayscale: ;
```

```
  --tw-backdrop-hue-rotate: ;
  --tw-backdrop-invert: ;
  --tw-backdrop-opacity: ;
  --tw-backdrop-saturate: ;
  --tw-backdrop-sepia: ;
}
```

```
/* classes.css */

.mx-auto {
  margin-left: auto;
  margin-right: auto;
}

.mt-6 {
  margin-top: 1.5rem;
}

.flex {
  display: flex;
}

.grid {
  display: grid;
}

.contents {
  display: contents;
}

.h-6 {
  height: 1.5rem;
}

.min-h-screen {
  min-height: 100vh;
}

.w-6 {
  width: 1.5rem;
}

.w-full {
  width: 100%;
}

.max-w-6xl {
```

```
  max-width: 72rem;
}

.flex-1 {
  flex: 1 1 0%;
}

.items-center {
  align-items: center;
}

.justify-between {
  justify-content: space-between;
}

.gap-2 {
  gap: 0.5rem;
}

.gap-4 {
  gap: 1rem;
}

.scroll-smooth {
  scroll-behavior: smooth;
}

.rounded-md {
  border-radius: 0.375rem;
}

.border-t-2 {
  border-top-width: 2px;
}

.bg-slate-100 {
  --tw-bg-opacity: 1;
  background-color: rgb(241 245 249 / var(--tw-bg-opacity));
}

.bg-white {
  --tw-bg-opacity: 1;
  background-color: rgb(255 255 255 / var(--tw-bg-opacity));
}

.p-4 {
  padding: 1rem;
}
```

```css
.pb-2 {
  padding-bottom: 0.5rem;
}

.pb-3 {
  padding-bottom: 0.75rem;
}

.pb-4 {
  padding-bottom: 1rem;
}

.pt-2 {
  padding-top: 0.5rem;
}

.pt-4 {
  padding-top: 1rem;
}

.text-2xl {
  font-size: 1.5rem;
  line-height: 2rem;
}

.text-lg {
  font-size: 1.125rem;
  line-height: 1.75rem;
}

.text-xl {
  font-size: 1.25rem;
  line-height: 1.75rem;
}

.font-medium {
  font-weight: 500;
}

.font-semibold {
  font-weight: 600;
}

.text-slate-900 {
  --tw-text-opacity: 1;
  color: rgb(15 23 42 / var(--tw-text-opacity));
}
```

```css
.underline {
  -webkit-text-decoration-line: underline;
  text-decoration-line: underline;
}

@media (min-width: 768px) {
  .md\:flex {
    display: flex;
  }

  .md\:w-\[20rem\] {
    width: 20rem;
  }

  .md\:flex-row-reverse {
    flex-direction: row-reverse;
  }

  .md\:flex-col {
    flex-direction: column;
  }

  .md\:gap-4 {
    gap: 1rem;
  }
}
```

```css
/* custom.css */

h2 {
  font-size: 1.125rem /* 18px */;
  line-height: 1.75rem /* 28px */;
  font-weight: 500;
}
a:hover {
  text-underline-offset: 2px;
  text-decoration-line: underline;
}
strong {
  font-weight: 500;
}
h1 {
  font-weight: 600;
  font-size: 1.5rem /* 24px */;
  line-height: 2rem /* 32px */;
  padding-bottom: 1rem /* 16px */;
```

```
}
main h2:not(.first-heading) {
  padding-bottom: 0.5rem /* 8px */;
  padding-top: 1rem /* 16px */;
}
main .first-heading {
  padding-bottom: 0.5rem /* 8px */;
}
code {
  overflow: hidden;
}
code pre {
  margin-bottom: -1.25rem /* -20px */;
  --tw-bg-opacity: 1;
  background-color: rgb(30 41 59 / var(--tw-bg-opacity));
  padding: 1rem /* 16px */;
  --tw-text-opacity: 1;
  color: rgb(248 250 252 / var(--tw-text-opacity));
  margin-top: 0.5rem /* 8px */;
}
```

Висновки

Таким чином, ми вивчили способи групування та організації елементів та навчились керувати розташуванням елементів на сторінці.

Контрольні питання

Чим відрізняються блочні та рядкові елементи

Блочні елементи зазвичай займають весь доступний горизонтальний простір та починаються з нового рядка, тоді як рядкові елементи займають лише необхідний горизонтальний простір та не переносяться на новий рядок.

Перерахуйте три блочних та три рядкових тега

Три блочні теги: , <p>, <h1>. Три рядкові теги: , <a>, .

Як значення block, inline, inline-block властивості display впливає на відображення елемента?

display: block; робить елемент блочним, відповідно він займає весь горизонтальний простір і починається з нового рядка.

display: inline; робить елемент рядковим, відповідно займає лише необхідний горизонтальний простір та не переносяться на новий рядок.

display: inline-block; комбінує рядкову та блочну поведінку, дозволяючи елементу займати необхідний горизонтальний простір і залишатися на тому ж рядку з іншими inline-елементами.