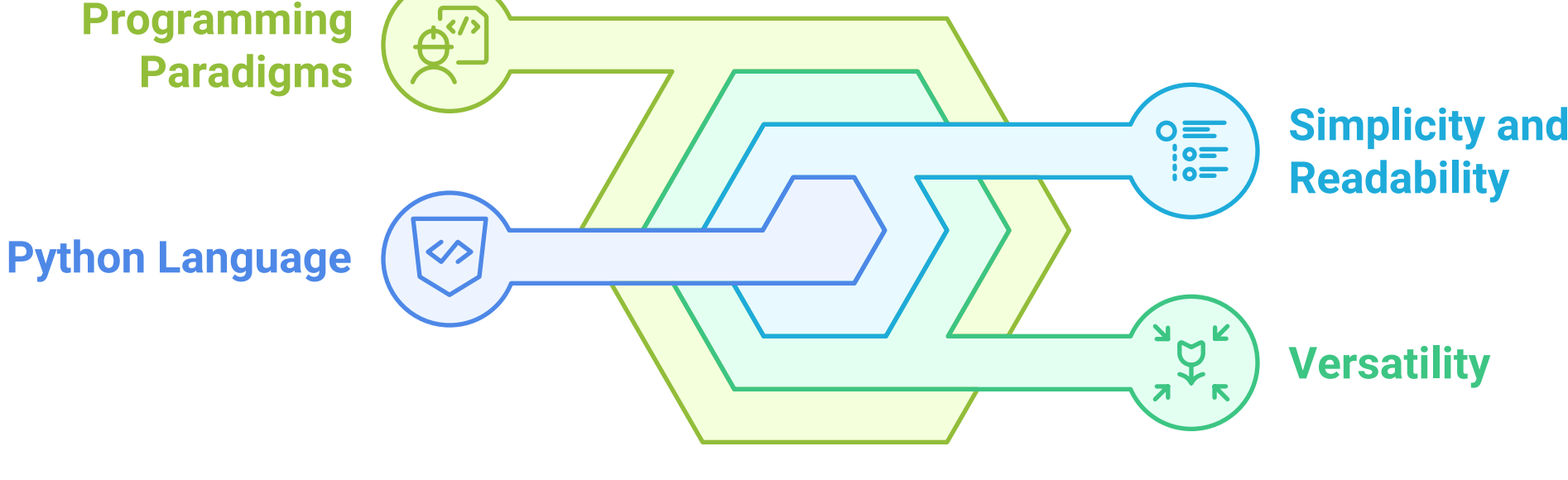


Python Notes

1. What is Python?

Python is a high-level, interpreted, and general-purpose programming language. It is known for its simplicity, readability, and versatility. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

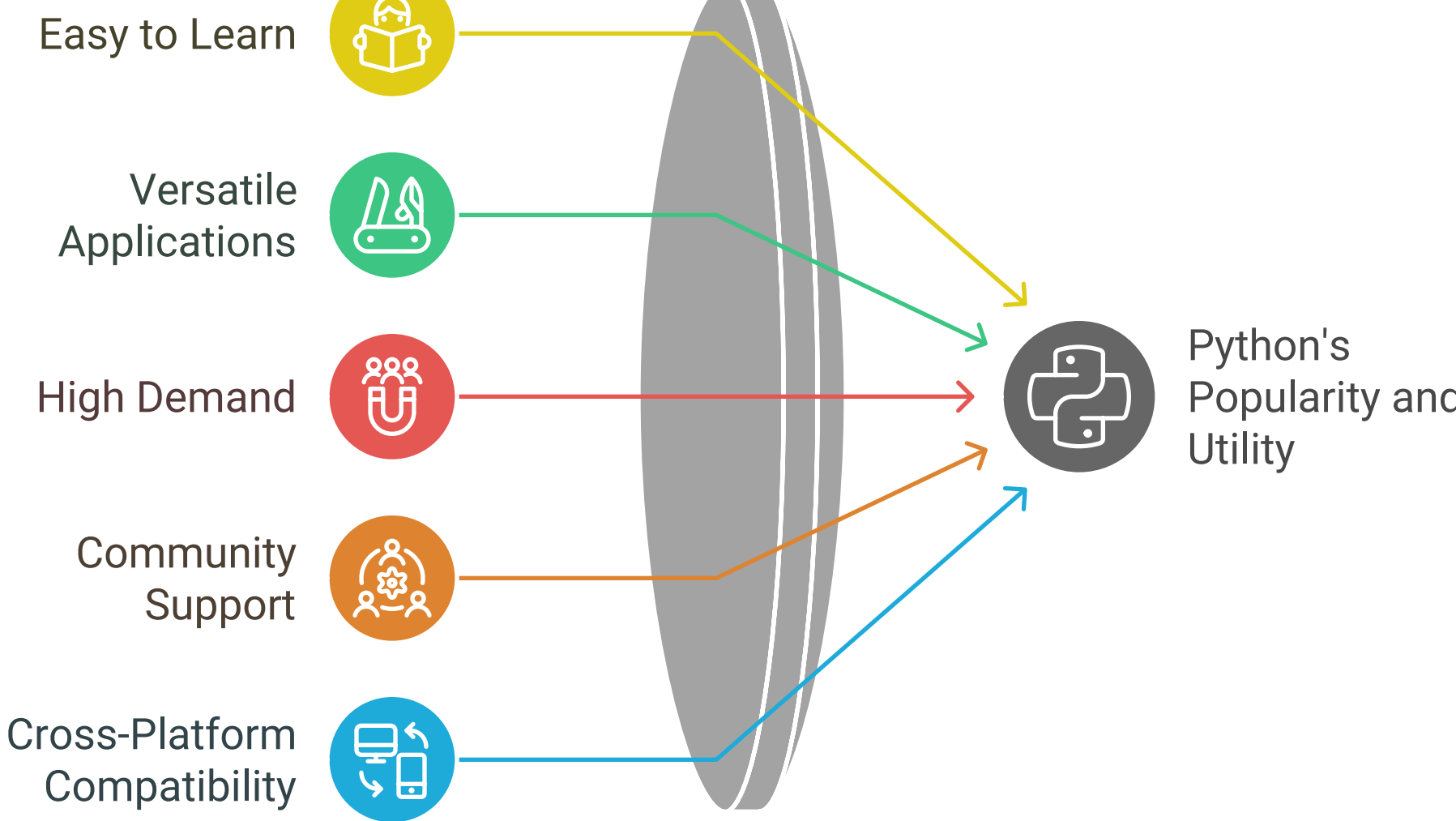
Python Programming Language Structure



2. Why Learn Python?

- **Easy to Learn:** Its simple syntax mimics natural language, making it beginner-friendly.
- **Versatile:** Used in web development, data analysis, machine learning, automation, and more.
- **High Demand:** Python developers are in demand in the tech industry.
- **Community Support:** A large community ensures extensive libraries, frameworks, and learning resources.
- **Cross-Platform:** Runs seamlessly on Windows, macOS, and Linux.

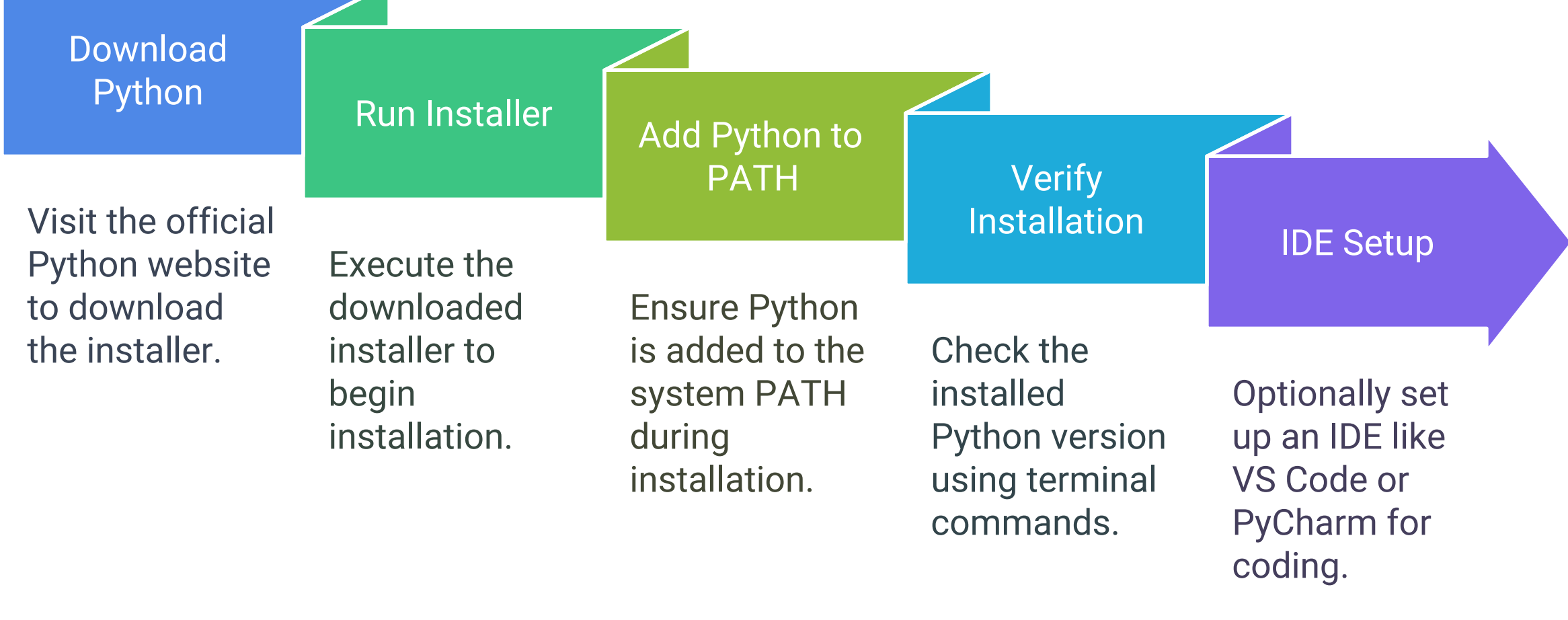
Factors Contributing to Python's Success



3. Installation of Python

1. **Download:**
 - Visit the official website: <https://www.python.org/downloads/>
 - Download the version suitable for your operating system.
2. **Install:**
 - Run the downloaded installer.
 - Check the box "Add Python to PATH" during installation.
3. **Verify Installation:**
 - Open a terminal or command prompt.
 - Type **python --version** or **python3 --version**.
 - It should display the installed Python version.
4. **IDE Setup** (Optional):
 - Use an Integrated Development Environment (IDE) like VS Code, PyCharm, or Jupyter Notebook for coding.

Python Installation and Setup Process



4. Python Syntax Basics

- **Hello World Example:**

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

- **Variables:**

```
name = "Alice"
age = 25
```

- **Loops:**

```
for i in range(5):
    print(i)
```

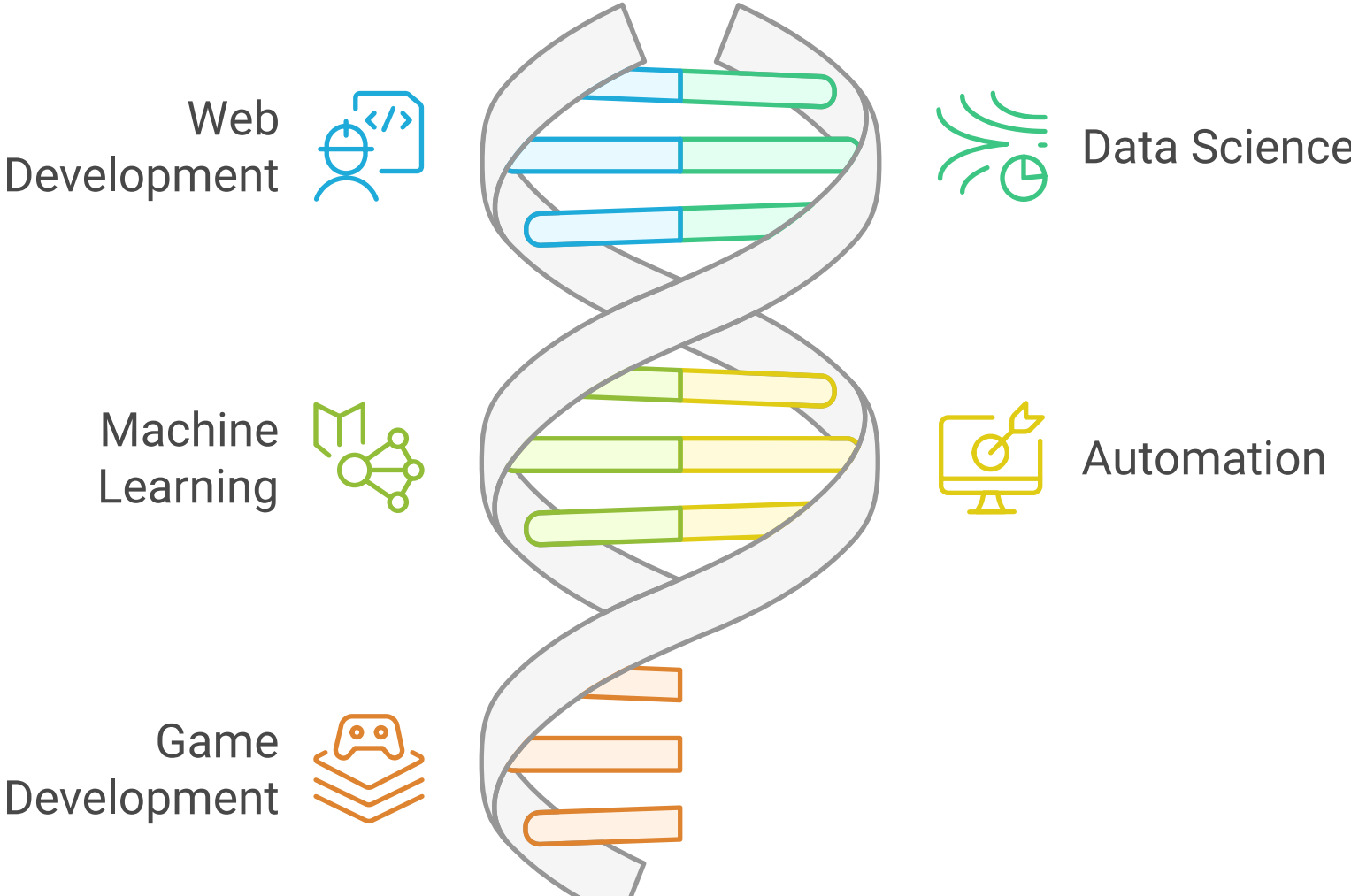
- **Conditionals:**

```
if age > 18:
    print("Adult")
else:
    print("Minor")
```

5. Applications of Python

1. **Web Development:** Frameworks like Django, Flask, and FastAPI.
2. **Data Science:** Libraries like Pandas, NumPy, Matplotlib, and Seaborn for data analysis and visualization.
3. **Machine Learning and AI:** Tools like TensorFlow, PyTorch, and Scikit-learn.
4. **Automation:** Scripts for automating repetitive tasks using libraries like Selenium and BeautifulSoup.
5. **Game Development:** Libraries like Pygame for building 2D games.
6. **Cybersecurity:** Ethical hacking tools and security scripts.
7. **Embedded Systems:** Python on devices like Raspberry Pi for IoT projects.

Exploring Diverse Python Applications in Modern Technology



6. Real-World Examples

- **Web Application:**

```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def home():
    return "Welcome to Flask!"

if __name__ == "__main__":
    app.run(debug=True)
```

- **Data Analysis:**

```
import pandas as pd
data = pd.read_csv("data.csv")
print(data.describe())
```

- **Automation:**

```
import os
files = os.listdir()
for file in files:
    print(file)
```

7. Advantages of Python

- Extensive library support.
- Simple and readable syntax.
- Scalable for large applications.
- Strong community and ecosystem.

Key Advantages of Python

