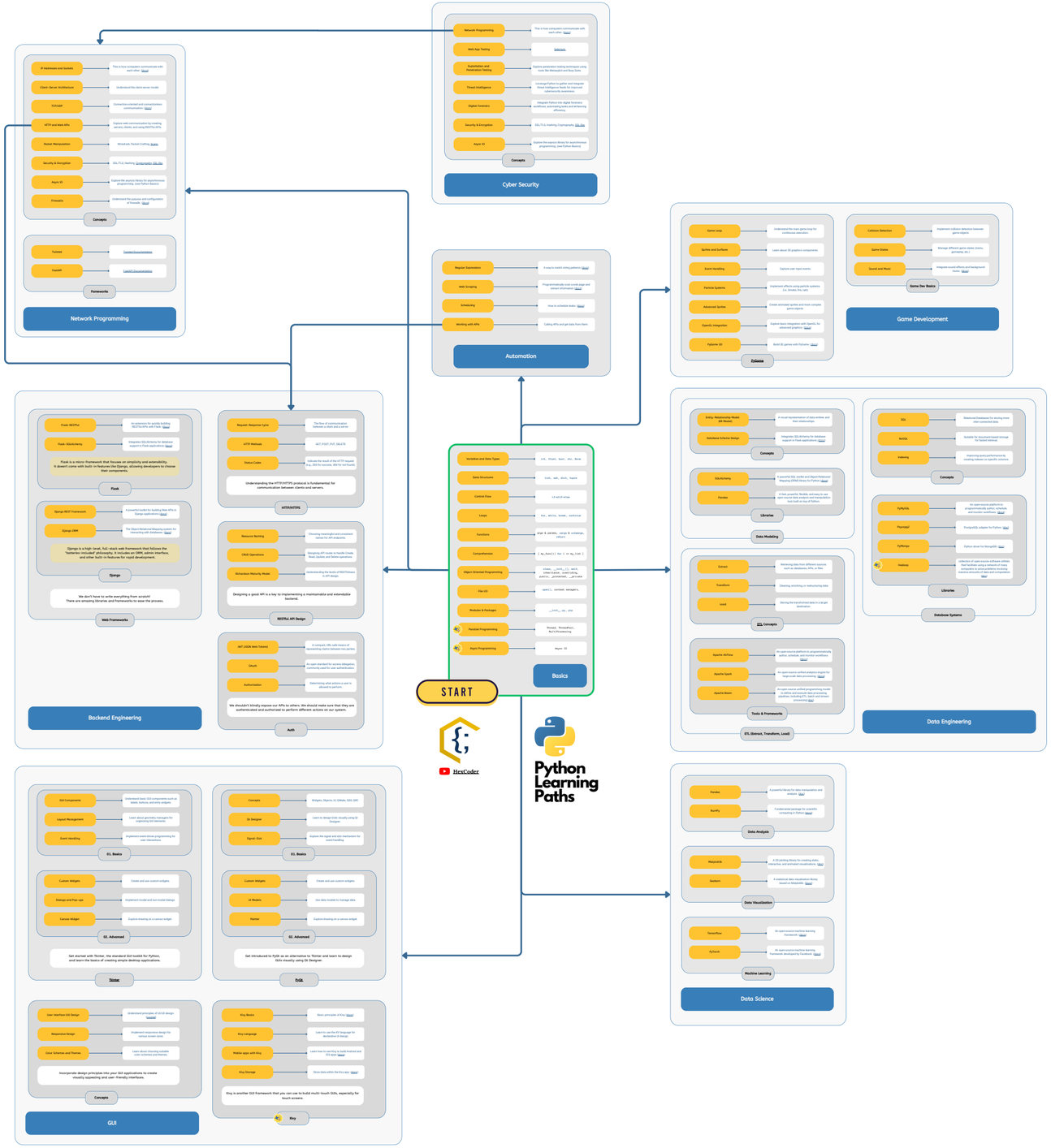
Network Programming
IP Addresses and Sockets
This is how computers communicate with each other. (Wow)
Client-Server Architecture
Understand the client-server model.
TCP/UDP
Connection-oriented and connectionless communication. (Wow)
HTTP and Web APIs
Explore web communication by creating servers, clients, and using RESTful APIs.
Packet Manipulation
Wireshark, Packet Crafting, Scapy.
Security & Encryption
SSL/TLS, Hashing, Cryptography, SSL Dos
Async IO
Explore the asyncio library for asynchronous programming. (see Python Basics)
Firewalls
Understand the purpose and configuration of firewalls. (Wow)
Concepts
Twisted
Twisted Documentation
FastAPI
FastAPI Documentation
Frameworks
Network Programming
Cyber Security
Network Programming
This is how computers communicate with each other. (Wow)
Web App Testing
Selenium.
Exploitation and Penetration Testing
Explore penetration testing techniques using tools like Metasploit and Burp Suite.
Threat Intelligence
Leverage Python to gather and integrate threat intelligence feeds, enhancing cybersecurity awareness.
Digital Forensics
Integrate Python into digital forensics workflows, automating tasks and enhancing efficiency.
Security & Encryption
SSL/TLS, Hashing, Cryptography, SSL Dos
Async IO
Explore the asyncio library for asynchronous programming. (see Python Basics)
Concepts
Cyber Security
Automation
Regular Expressions
A way to match string patterns. (Wow)
Web Scraping
Programmatically scrape a web page and extract information. (Wow)
Scheduling
How to schedule tasks. (Wow)
Working with APIs
Calling APIs and get data from them.
Automation
Game Loop
Understand the main game loop for continuous execution.
Sprites and Surfaces
Learn about 2D graphics components.
Event Handling
Capture user input events.
Particle Systems
Implement effects using particle systems. (i.e. Smoke, fire, rain)
Advanced Sprites
Create animated sprites and more complex game objects.
Opencv Integration
Explore basic integration with OpenCV for advanced graphics. (Wow)
Pygame 2D
Build 2D games with PyGame. (Wow)
PyGame
Collision Detection
Implement collision detection between game objects.
Game States
Manage different game states (menu, gameplay, etc)
Sound and Music
Integrate sound effects and background music. (Wow)
Game Dev Basics
Game Development
Flask: RESTful
An extension for quickly building RESTful APIs with Flask. (Wow)
Flask-SQLAlchemy
Integrates SQLAlchemy for database support in Flask applications. (Wow)
Flask is a micro-framework that focuses on simplicity and extensibility. It doesn't come with built-in features like Django, allowing developers to choose their components.
Flask
Request-Response Cycle
The flow of communication between a client and a server
HTTP Methods
GET,POST,PUT,DELETE
Status Codes
Indicate the result of the HTTP request (e.g., 200 for success, 404 for not found)
Understanding the HTTP/HTTPS protocol is fundamental for communication between clients and servers.
HTTP/HTTPS
Django REST Framework
A powerful toolkit for building Web APIs in Django applications. (Wow)
Django ORM
The Object-Relational Mapping system for interacting with databases. (Wow)
Django is a high-level, full-stack web framework that follows the "batteries included" philosophy. It includes an ORM, admin interface, and other built-in features for rapid development.
Django
Resource Naming
Choosing meaningful and consistent names for API endpoints.
CRUD Operations
Designing API routes to handle Create, Read, Update, and Delete operations.
Richardson Maturity Model
Understanding the levels of RESTfulness in API design.
Designing a good API is a key to implementing a maintainable and extendable backend.
RESTful API Design
We don't have to write everything from scratch!
There are amazing libraries and frameworks to ease the process.
Web Frameworks
JWT (JSON Web Token)
A compact, URL-safe means of representing claims between two parties
OAuth
An open standard for access delegation, commonly used for user authentication
Authorization
Determining what actions a user is allowed to perform
We shouldn't blindly expose our APIs to others. We should make sure that they are authenticated and authorized to perform different actions on our system.
Auth
Backend Engineering
Variables and Data Types
int, float, bool, str, None
Data Structures
list, set, dict, tuple
Control Flow
if-elif-else
Loops
for, while, break, continue
Functions
def & parameters, range & enumerate, return
Comprehensions
[x for x in lst if ...]
Object Oriented Programming
class, __init__(), self, inheritance, super(), dunder methods, public, protected, __private
File IO
open(), context managers,
Modules & Packages
import, pip, pip
Parallel Programming
Thread, ThreadPool, MultiProcessing
Async Programming
Async IO
Basics
START
Entity-Relationship Model (ER Model)
A visual representation of data entities and their relationships.
Database Schema Design
Integrates SQLAlchemy for database support in Flask applications. (Wow)
Concepts
SQLAlchemy
A powerful SQL toolkit and Object-Relational Mapping (ORM) library for Python. (Wow)
Pandas
A fast, powerful, flexible, and easy to use open-source data analysis and manipulation tool, built on top of Python.
Libraries
Data Modeling
Extract
Retrieving data from different sources such as databases, APIs, or files.
Transform
Cleaning, enriching or restructuring data.
Load
Storing the transformed data in a target destination.
ETL Concepts
Apache Airflow
An open-source platform to programmatically author, schedule, and monitor workflows. (Wow)
Apache Spark
An open-source unified analytics engine for large-scale data processing. (Wow)
Apache Beam
An open-source unified programming model to define and execute data processing pipelines, including ETL, batch and stream (streaming) data.
Tools & Frameworks
ETL (Extract, Transform, Load)
SQL
Relational Databases for storing more interconnected data.
NoSQL
Suitable for document-based storage for faster retrieval.
Indexing
Improving query performance by creating indexes on specific columns.
Concepts
PyMySQL
An open-source platform to programmatically author, schedule, and monitor workflows. (Wow)
Psycopg2
PostgreSQL adapter for Python. (Wow)
PyMongo
Python driver for MongoDB. (Wow)
Peewee
collection of open-source software utilities that facilitate using a relational database to manage object persistence in applications handling large volumes of data and complexities. (Wow)
Libraries
Database Systems
Data Engineering
Pandas
A powerful library for data manipulation and analysis. (Wow)
NumPy
Fundamental package for scientific computing in Python. (Wow)
Data Analysis
Matplotlib
A 2D plotting library for creating static, interactive, and animated visualizations. (Wow)
Seaborn
A statistical data visualization library based on Matplotlib. (Wow)
Data Visualization
TensorFlow
An open-source machine learning framework. (Wow)
PyTorch
An open-source machine learning framework developed by Facebook. (Wow)
Machine Learning
Data Science
GUI Components
Understand basic GUI components such as labels, buttons, and entry widgets.
Layout Management
Learn about geometry managers for organizing GUI elements.
Event Handling
Implement event-driven programming for user interactions.
01. Basics
Custom Widgets
Create and use custom widgets.
Dialogs and Pop-ups
Implement modal and non-modal dialogs.
Canvas Widget
Explore drawing on a canvas widget.
02. Advanced
Get started with Tkinter, the standard GUI toolkit for Python, and learn the basics of creating simple desktop applications.
Tkinter
Concepts
Widgets, Objects, UI, Dialogs, GUI, GRC
Qt Designer
Learn to design GUIs visually using Qt Designer.
Signal-Slot
Explore the signal and slot mechanism for event handling.
01. Basics
Custom Widgets
Create and use custom widgets.
UI Models
Use data models to manage data.
Painter
Explore drawing on a canvas widget.
02. Advanced
Get introduced to PyQt as an alternative to Tkinter and learn to design GUIs visually using Qt Designer.
PyQt
User Interface GUI Design
Understand principles of UI/UX design. (wikipedia)
Responsive Design
Implement responsive design for various screen sizes.
Color Schemes and Themes
Learn about choosing suitable color schemes and themes.
Incorporate design principles into your GUI applications to create visually appealing and user-friendly interfaces.
Concepts
Kivy Basics
Basic principles of Kivy. (Wow)
Kivy Language
Learn to use the KV language for declarative UI design.
Mobile apps with Kivy
Learn how to use Kivy to build Android and iOS apps. (Wow)
Kivy Storage
Store data within the Kivy app. (Wow)
Kivy is another GUI framework that you can use to build multi-touch GUIs, especially for touch screens.
Kivy
GUI
HexCoder
Python Learning Paths