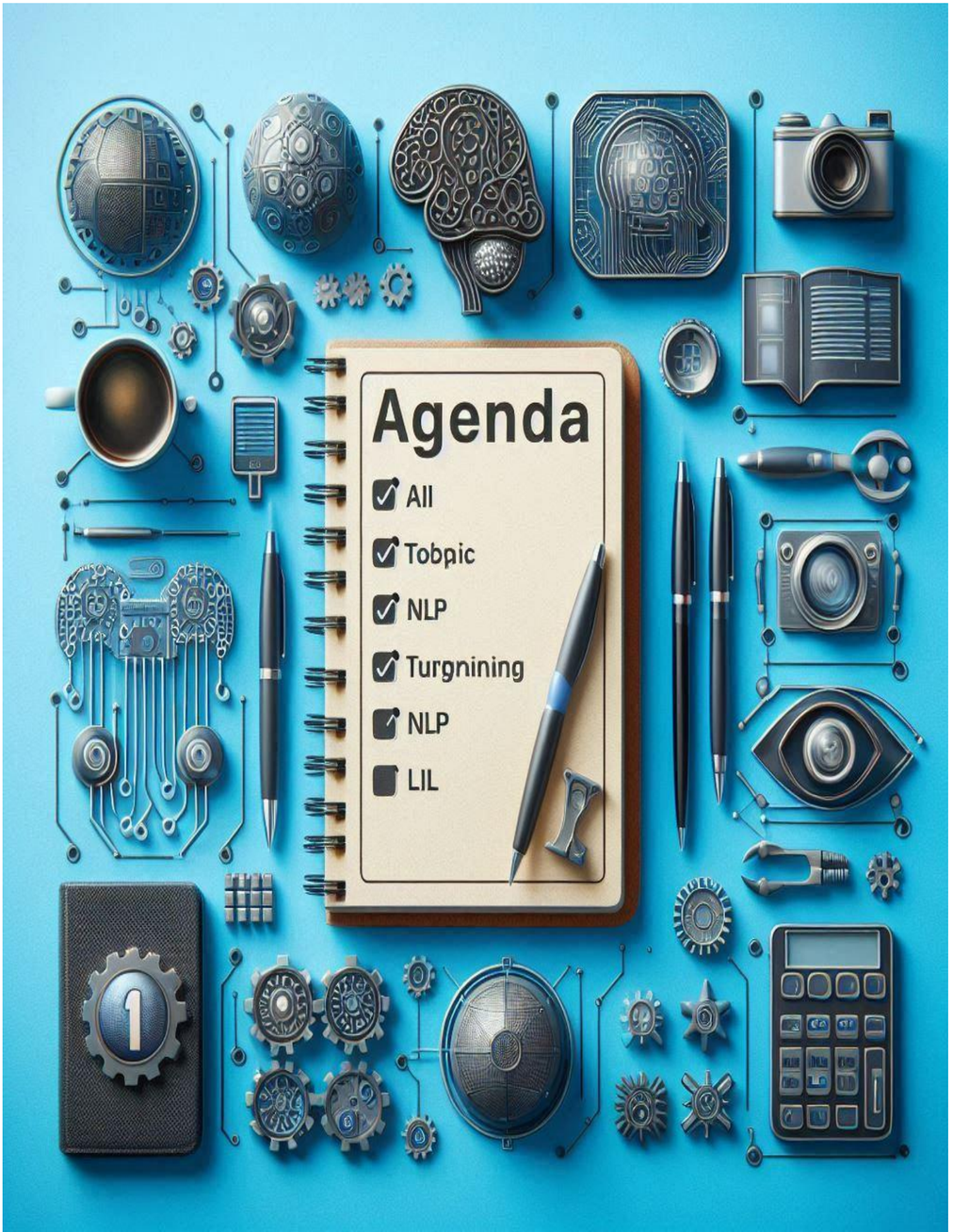
National Telecommunication Institute

What are your expectations for the course?

National Telecommunication Institute

Agenda



Part 1: OCR – Optical Character Recognition

- Introduction to OCR
- OCR Applications
- Technical Pipeline
- Tools & Algorithms
- Challenges and Trends

Part 2: Web Scraping

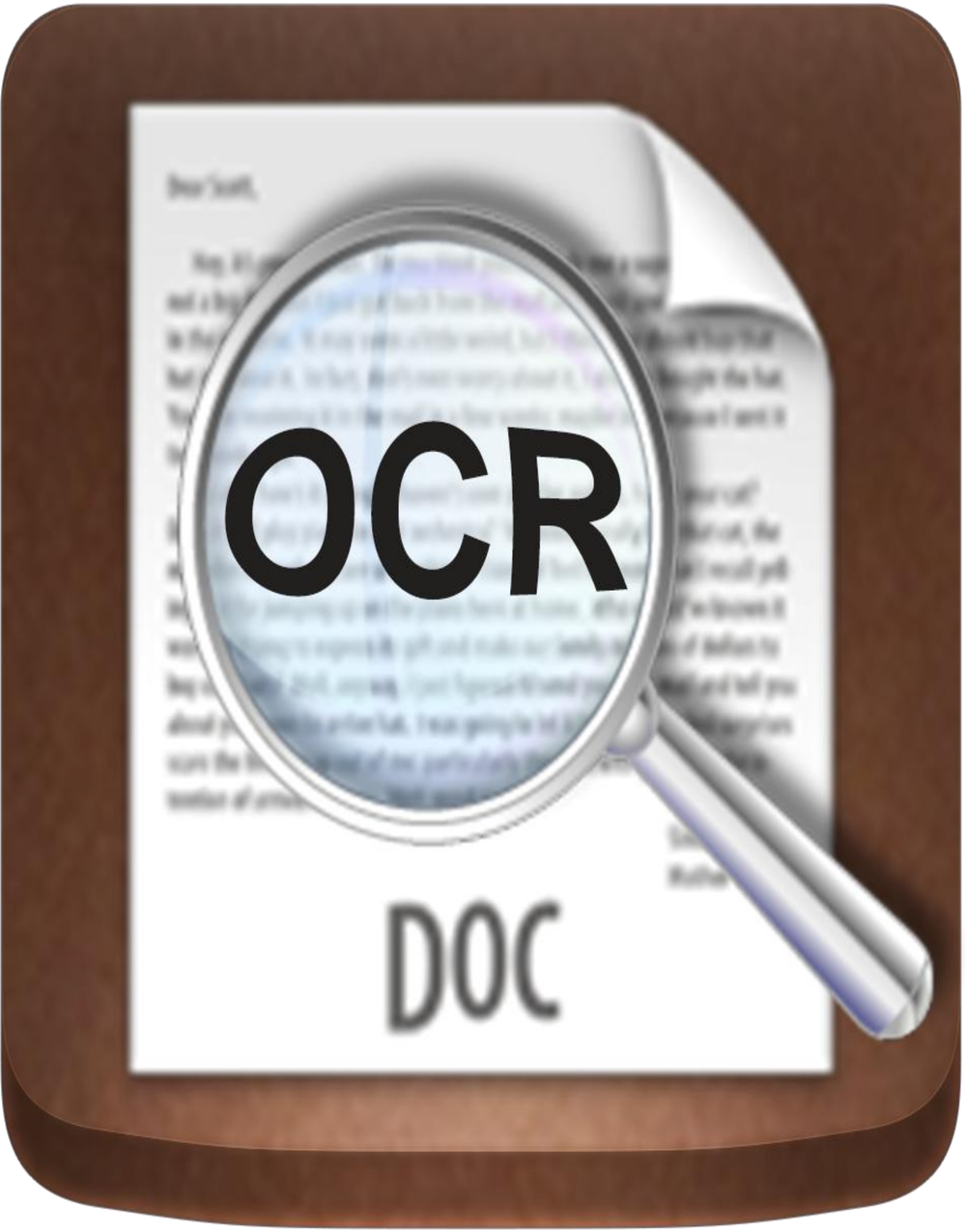
- Introduction to Web Scraping
- Key Concepts (HTML, DOM, XPath)
- Libraries & Tools
- Legal/Ethical Considerations
- Common Challenges & Best Practices

Part 3: Voice Recognition

- Voice vs Speech Recognition
- Audio Processing Concepts
- Applications & APIs
- Deep Learning in Voice Tech
- Future Directions and Challenges

National Telecommunication Institute

What is Optical Character Recognition (OCR)?



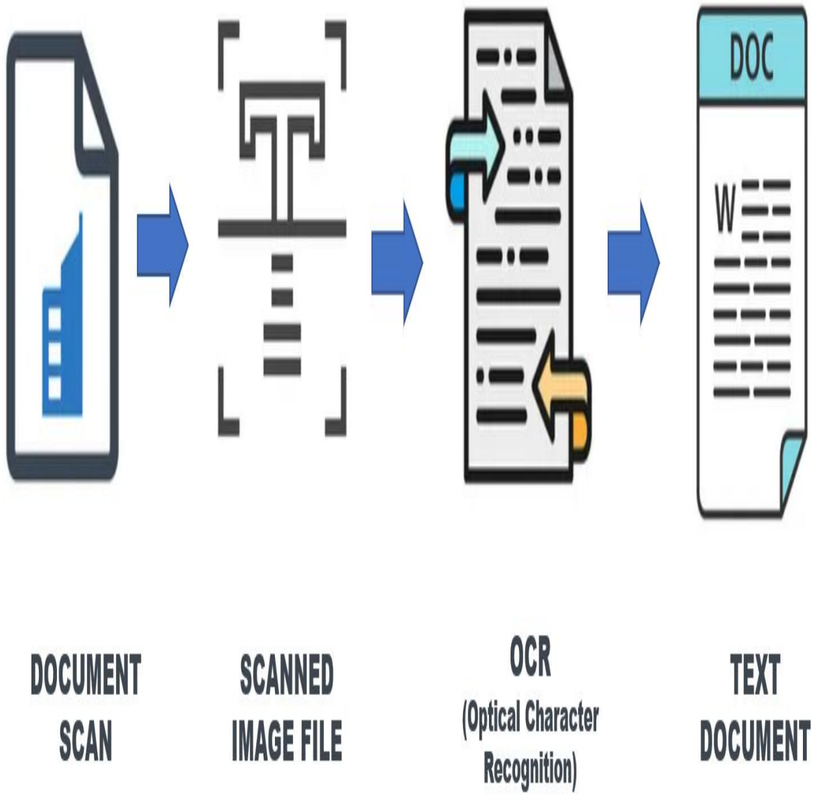
- **OCR** stands for **Optical Character Recognition**
- It is a technology used to **convert different types of documents**—such as scanned paper documents, PDFs, or images—into **editable and searchable data**
- Recognizes **text characters** in printed or handwritten form
- Used in **digital archiving**, **automation**, **data extraction**, and more

National Telecommunication Institute

History and Evolution of OCR

- **1929**: First OCR-like machine patented for reading characters (Gustav Tauschek – Germany)
- **1950s–60s**: OCR developed for banking and postal systems
- **1965**: IBM's **“Scan-Optical Reader”** introduced
- **1970s–80s**: OCR commercialized for office use (reading typed or printed text)
- **1990s**: Introduction of OCR software for PCs (like ABBYY, OmniPage)
- **2000s–Today**:
- AI-based OCR with **machine learning and deep learning**
- Open-source tools like **Tesseract (by Google)**
- Supports **handwriting**, multiple languages, and **real-time OCR** on smartphones

How OCR Works



- **Image Acquisition**
  - Input can be scanned documents, photos, or PDFs
- **Preprocessing**
  - Noise reduction, binarization, resizing, deskewing
  - Improves image quality for better recognition
- **Text Detection (Segmentation)**
  - Locates regions in the image that contain text
  - Segments lines, words, and individual characters
- **Character Recognition**
  - Uses pattern matching or ML models to identify characters
  - Converts visual symbols into digital characters (A–Z, 0–9, etc.)
- **Postprocessing**
  - Spell check, error correction, language modeling
  - Formats the recognized text into structured output
- **Output Generation**
  - Text is saved in formats like TXT, DOCX, or searchable PDFs

Preprocessing Techniques in OCR

- **Grayscale Conversion**
  Converts color images to shades of gray to simplify processing
- **Binarization**
  Converts grayscale to black & white
  Common method: **Otsu's Thresholding**
- **Noise Removal**
  Removes specks, distortions, or smudges using filters (e.g., median filter)
- **Deskewing**
  Corrects tilted or rotated images to align text properly
- **Cropping & Border Removal**
  Focuses on text areas and removes irrelevant parts
- **Text Line Segmentation**
  Breaks down the image into lines, words, or characters
- **Normalization**
  Rescales and aligns characters for consistent shape recognition

National Telecommunication Institute

Text Detection vs Text Recognition

Text Detection	Text Recognition
Finds **where** text is located in image	Identifies **what** the text actually says
Outputs bounding boxes around text areas	Converts image regions into actual characters
Doesn’t care about content	Focuses on content inside text areas
Techniques: MSER, EAST, CTPN	Techniques: CNNs, RNNs, CRNNs, Tesseract OCR
Step 1 in OCR pipeline	Step 2 in OCR pipeline

Tools of OCR

🛠️ Tool / Algorithm	🧠 Type	📄 Use Case	🧩 Special Strengths
Tesseract OCR	Deep Learning (LSTM)	Document OCR	Open-source, multi-language, CLI & Python support
CRNN (CNN + RNN + CTC)	Deep Learning	Handwriting, Scene Text OCR	Sequence modeling, real-time performance
EAST (Text Detector)	Deep Learning	Scene Text Detection	Fast and accurate rotated box detection
MSER + HOG + SVM	Traditional (Classical CV)	Basic OCR, Simple Layouts	Lightweight, no training needed
Google Vision API	Cloud + Deep Learning	Documents, Images, Mobile	Powerful cloud-based OCR, real-time text reading
AWS Textract	Cloud + Deep Learning	Structured Docs (Forms, Tables)	Text + Form parsing, table structure extraction
ABBYY FineReader	Hybrid	Professional Document OCR	High-accuracy, PDF editing, OCR SDK available
OpenCV OCR	Traditional / Hybrid	Custom OCR Tasks	Flexible, can integrate Tesseract or custom tools

Challenges in OCR

- **Handwriting Recognition**
  - Variability in writing styles and spacing
  - Cursive and connected letters are hard to segment
- **Image Noise & Artifacts**
  - Blurred scans, shadows, low resolution, or compression artifacts
  - Can confuse detection and recognition stages
- **Skewed or Rotated Text**
  - Scanned documents or camera-captured images may be tilted
- **Fonts & Text Styles**
  - Decorative or distorted fonts can hinder recognition accuracy
- **Multilingual and Mixed Scripts**
  - OCR engines may struggle with documents that mix languages/scripts
  - E.g., Arabic + English or Latin + Cyrillic
- **Scene Text in Complex Backgrounds**
  - Real-world images (e.g., street signs) have cluttered backgrounds, angles, lighting variations
- **Structured Documents**
  - Tables, forms, and multi-column layouts are hard to parse correctly

Future of OCR and AI Integration

Deep Learning & Transformers for OCR

- Shift from rule-based systems to AI-driven models
- Transformer-based OCR (e.g., TrOCR by Microsoft) improves accuracy

Better Handwriting Recognition

- AI models are learning cursive, mixed-language, and personalized handwriting
- Useful for digitizing historical documents and handwritten notes

Document Understanding & Layout Analysis

- Beyond just reading text — AI models extract structure, fields, tables
- Foundation for **Intelligent Document Processing (IDP)**

Multilingual & Cross-Script OCR

- Modern OCR engines now support real-time multi-language recognition
- Integration with translation systems

On-Device & Edge AI OCR

- OCR on mobile without internet
- Faster, more private, and usable in remote areas

Combining OCR with NLP & RPA

- End-to-end automation: Scan → Understand → Take action
- Used in finance, legal, logistics, HR, and healthcare

National Telecommunication Institute

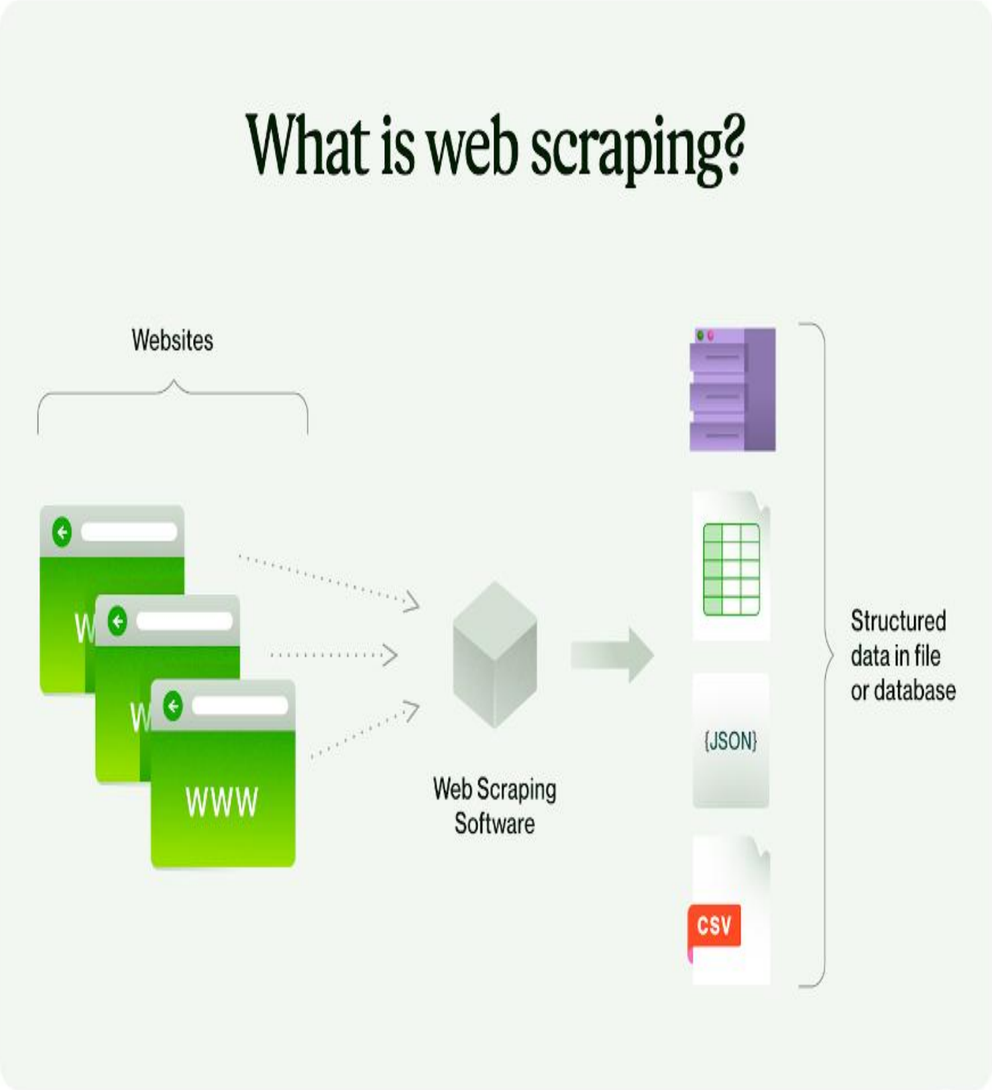
WEB SCRAPING

HTML WEBSITES

WEB SCRAPING

DATA

What is Web Scraping?

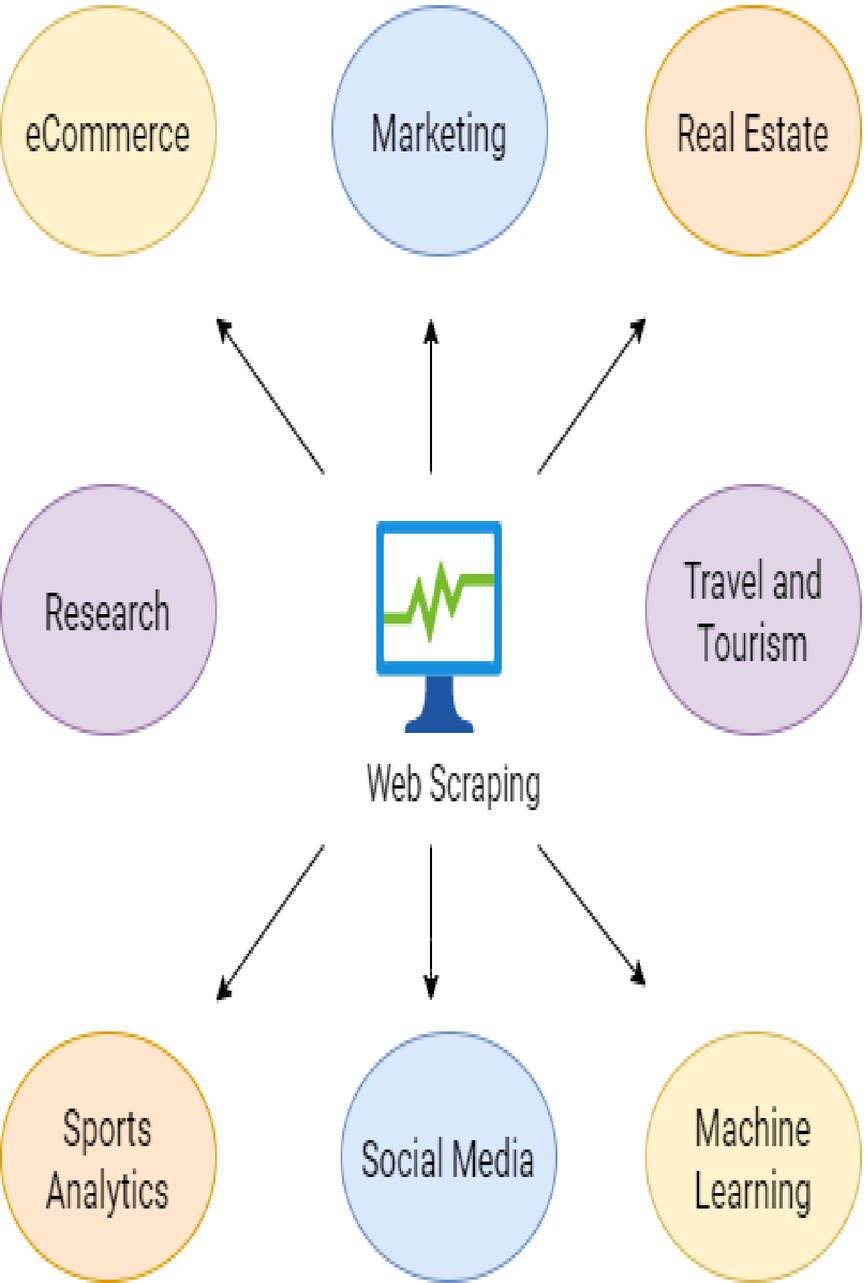


- **Web Scraping** is the process of **automatically extracting data** from websites
- It involves sending requests to web pages and **parsing the HTML content** to extract structured data
- Often used to collect **text, tables, prices, images, links**, and more
- A key technique for **data mining**, **competitive analysis**, **market research**, and **AI dataset creation**
- Tools used include **BeautifulSoup**, **Scrapy**, **Selenium**, and browser automation libraries

Difference Between Scraping and Crawling

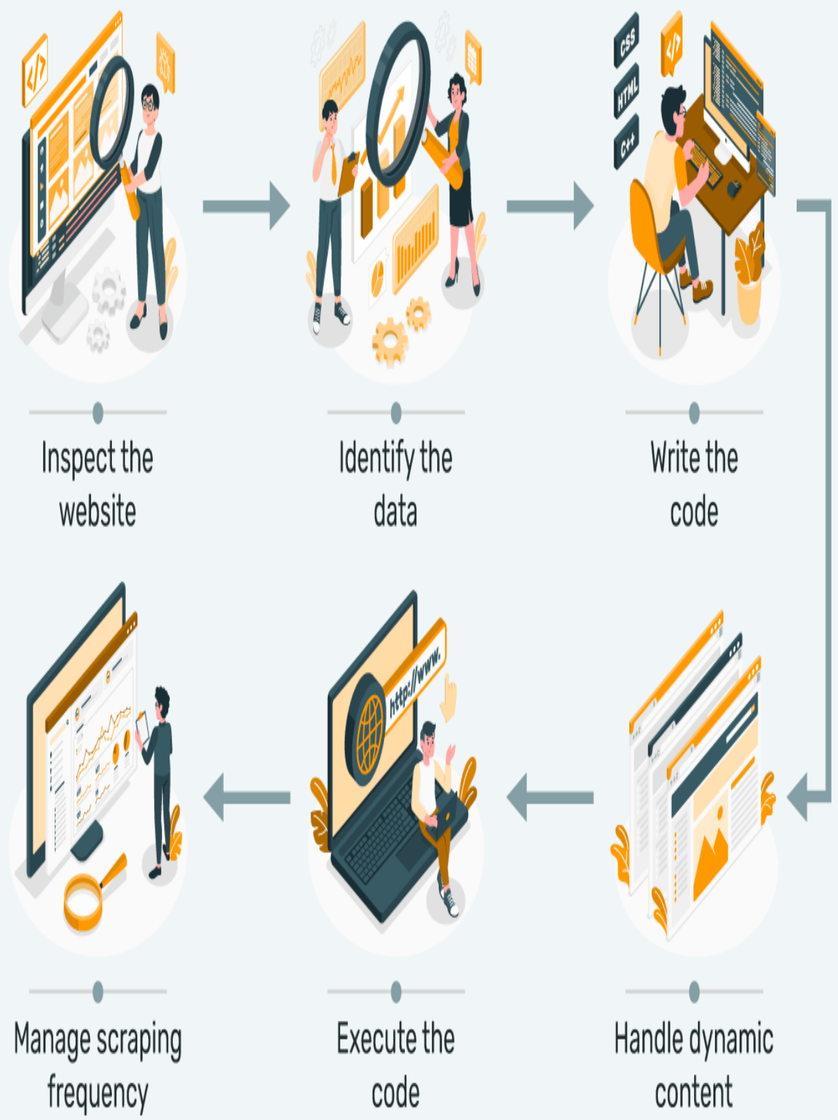
Web Scraping	**Web Crawling**
Extracts **specific data** from web pages	Discovers and indexes **multiple web pages**
Focuses on **content** (text, images, etc.)	Focuses on **structure** and links between pages
Works on **known pages or targets**	Automatically **explores** new pages via links
Used for **data collection**	Used for **site indexing**, **data discovery**
Example: Extracting product prices	Example: Googlebot indexing websites
Tools: BeautifulSoup, Selenium, Scrapy	Tools: Scrapy, Apache Nutch, custom crawlers

Web Scraping Use Cases



- **E-commerce Price Monitoring**
  Track competitor pricing, discounts, and stock availability
- **News Aggregation**
  Collect headlines, articles, and summaries from news sites
- **Review & Sentiment Analysis**
  Scrape user reviews from sites like TripAdvisor, Amazon, etc.
  Train sentiment analysis or rating prediction models
- **Market & Financial Data**
  Extract stock prices, economic indicators, crypto values, etc.
- **Academic Research & Open Data**
  Gather data from government, health, or education portals
- **Job Listings & Resume Mining**
  Collect postings from LinkedIn, Indeed, or Glassdoor for trends or job matching
- **AI & ML Dataset Creation**
  Build custom datasets for NLP, computer vision, recommendation systems
- **Real Estate & Classifieds**
  Extract details about properties, cars, or listings for analysis or resale

How Web Scraping Works



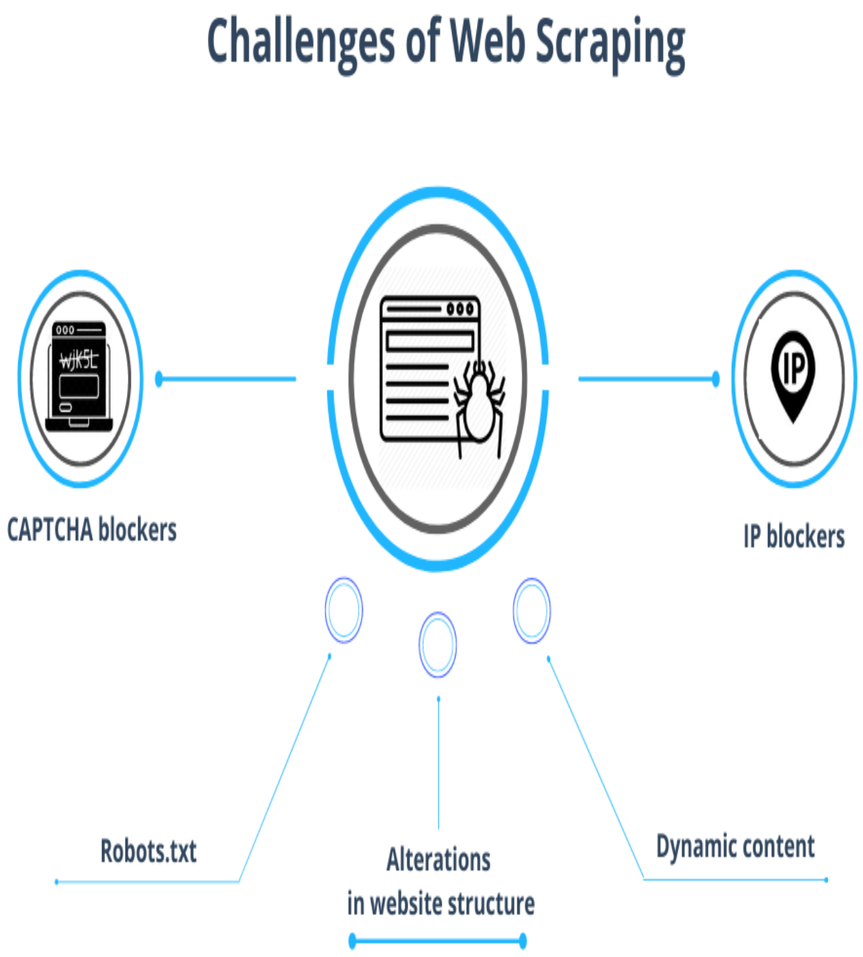
- **Send HTTP Request**
  Use tools like requests or url lib to access a web page (GET method)
- **Receive HTML Response**
  The server returns raw HTML content of the page
- **Parse the HTML**
  Use parsers (e.g., **BeautifulSoup**, **lxml**) to navigate the page structure
  Locate elements by tags, classes, IDs, or XPath
- **Extract the Data**
  Pull specific data (text, images, links, tables) from HTML tags
- **Clean & Format Data**
  Remove unwanted tags, whitespace, or symbols
  Organize into structured formats (CSV, JSON, database)
- **Save or Use the Data**
  Store locally or feed into dashboards, ML pipelines, or databases

Common Python Libraries

- **BeautifulSoup**
  Lightweight HTML/XML parser
  Great for small projects and static pages
  Easy navigation using tags, classes, and IDs
- **Scrapy**
  Full-featured **web crawling and scraping framework**
  Supports asynchronous scraping, pipelines, and item storage
  Ideal for large-scale or production scrapers
- **Selenium**
  Automates web browsers (Chrome, Firefox, etc.)
  Handles **JavaScript-heavy websites** and dynamic content
  Useful for interacting with buttons, forms, scroll, etc.
- **Requests**
  Simplifies HTTP requests (GET, POST, headers)
  Often used with BeautifulSoup for static pages

- **Respect Terms of Service (ToS)**
  Many websites explicitly forbid scraping in their ToS
  Violating this can lead to legal action or IP bans
- **Follow robots.txt**
  A file that specifies which parts of a site can/can’t be accessed by bots
  Ethical scrapers obey these rules
- **Avoid Overloading Servers**
  Use rate limiting and polite scraping (e.g., sleep between requests)
  Prevents denial of service or server crashes
- **Don’t Scrape Personal or Sensitive Data**
  Avoid scraping names, emails, or private information without consent
  Can violate privacy laws like **GDPR** or **CCPA**
- **Legal Risks**
  Some countries treat scraping of protected content as illegal (copyright, data theft)
  Notable cases: LinkedIn vs. hiQ Labs (US), Ryanair vs. screen-scrapers (EU)
- **Use APIs When Available**
  Many platforms offer official APIs — more stable, legal, and structured

Challenges in Web Scraping



Dynamic Content & JavaScript Rendering

- Many modern websites load data via JavaScript
- Requires tools like **Selenium**, **Playwright**, or **headless browsers**

Anti-Scraping Measures

- IP blocking, CAPTCHA, honeypots, user-agent filtering
- Solutions: proxies, user-agent rotation, CAPTCHA solvers

Changing Website Structure

- Frequent updates to HTML layout can break scrapers
- Needs regular maintenance and robust selectors

Rate Limiting & Bans

- Sending too many requests too fast may lead to temporary/permanent bans
- Use throttling, delays, and respect

Pagination & Infinite Scroll

- Data may be split across multiple pages or loaded on scroll
- Requires logic to follow links or simulate scrolling

Complex Page Layouts & Nested Elements

- Data inside tables, accordions, tabs, or deeply nested divs
- Requires careful parsing with XPath or CSS selectors

National Telecommunication Institute