Data Sourcing - Exam Questions

Objective Questions (MCQ/MSQ) - 20 Questions

1. Which Excel function scrapes data from web tables?

A) `WEBSERVICE()`
B) `IMPORTHTML()`
C) `WEBQUERY()`
D) `IMPORTDATA()`

Answer: D

2. What does the `requests` library in Python primarily handle?

A) Data processing
B) HTTP requests
C) File operations
D) Database connections

Answer: B

3. Which CSS selector targets all elements with class “product”?

A) `#product`
B) `.product`
C) `product`
D) `*product`

Answer: B

4. In BeautifulSoup, which method finds the first matching element?

A) `find_all()`

B) `find()`

C) `select()`

D) `get()`

Answer: B

5. Which HTTP status code indicates successful data retrieval?

A) `200`

B) `201`

C) `404`

D) `500`

Answer: A

6. What does Playwright primarily enable?

A) Data analysis

B) Web automation and scraping

C) Database management

D) File processing

Answer: B

7. Which command-line tool is commonly used for web scraping?

A) `wget`

B) `curl`

C) `scrapy`

D) All of the above

Answer: D

8. In API requests, what does the `User-Agent` header specify?

A) User credentials

B) Browser/client identification

C) Request type
D) Data format

Answer: B

9. Which Python library handles PDF text extraction?

A) `PyPDF2`
B) `pdfplumber`
C) `tabula-py`
D) All of the above

Answer: D

10. What does rate limiting in web scraping prevent?

A) Data corruption
B) Server overload
C) Authentication errors
D) Parsing errors

Answer: B

11. Which format is commonly used for API responses?

A) XML
B) JSON
C) CSV
D) Both A and B

Answer: D

12. In web scraping, what does `robots.txt` specify?

A) Website structure
B) Scraping permissions
C) Data formats
D) Authentication methods

Answer: B

13. Which Python library handles JavaScript-rendered content?

A) `requests`
B) `selenium`
C) `beautifulsoup`
D) `scrapy`

Answer: B

14. What does the `time.sleep()` function do in scraping scripts?

A) Improves performance
B) Adds delays between requests
C) Handles errors
D) Parses data

Answer: B

15. Which Wikipedia API endpoint provides page content?

A) `/api.php`
B) `/wiki/`
C) `/content/`
D) `/pages/`

Answer: A

16. In Nominatim API, what does geocoding convert?

A) Coordinates to addresses
B) Addresses to coordinates
C) Both A and B
D) Neither A nor B

Answer: B

17. Which tool converts HTML to Markdown?

A) `pandoc`
B) `html2text`
C) `markdownify`
D) All of the above

Answer: D

18. What does `tabula-py` extract from PDFs?

A) Text only
B) Images only
C) Tables
D) Metadata

Answer: C

19. Which GitHub Actions event triggers on schedule?

A) `on: schedule`
B) `on: cron`
C) `on: timer`
D) `on: interval`

Answer: A

20. In web scraping, what does session management maintain?

A) Data consistency
B) Login state and cookies
C) Request speed
D) Error handling

Answer: B

Subjective/Scenario Questions - 20 Questions

1. Web Scraping Architecture

Design a scalable web scraping system for monitoring product prices across multiple e-commerce sites. Include rate limiting, error handling, data storage, and legal compliance considerations.

Answer: Architecture: **Scrapy framework** with distributed crawling, **rotating proxies** and user agents, **rate limiting** per domain, **retry mechanisms** with exponential backoff, **data validation** and cleaning, **database storage** (PostgreSQL/MongoDB), **monitoring** and alerting, **robots.txt** compliance, **terms of service** review, **data retention** policies, and **change detection** algorithms.

2. API Integration Strategy

Create a comprehensive strategy for integrating data from multiple APIs with different authentication methods, rate limits, and data formats. How would you handle failures and ensure data consistency?

Answer: Strategy: **API wrapper classes** for each service, **OAuth/API key** management, **rate limiting** with token buckets, **circuit breakers** for failures, **retry logic** with jitter, **data normalization** pipelines, **schema validation**, **caching** for expensive calls, **monitoring** and alerting, **fallback mechanisms**, **data quality** checks, and **audit logging**.

3. Real-time Data Pipeline

Design a real-time data sourcing pipeline that collects data from social media APIs, news feeds, and web scraping. How would you handle high-volume streams and ensure data quality?

Answer: Pipeline: **Apache Kafka** for streaming, **multiple producers** for different sources, **stream processing** with Apache Spark, **data validation** rules, **duplicate detection**, **schema evolution**, **backpressure handling**, **error queues**, **monitoring** dashboards, **data lineage** tracking, **quality metrics**, and **alerting** systems.

4. Legal and Ethical Compliance

Establish guidelines for ethical data sourcing that comply with legal requirements, respect website terms of service, and maintain data privacy. What framework would you implement?

Answer: Framework: **Legal review** process, **robots.txt** compliance checking, **rate limiting** policies, **data minimization** principles, **consent management**, **PII detection** and handling, **terms of service** monitoring, **copyright** considerations, **data retention** policies, **audit trails**, **stakeholder** training, and **regular** compliance reviews.

5. Multi-format Data Integration

Design a system to extract and integrate data from PDFs, HTML tables, APIs, and databases into a unified format. What challenges would you address?

Answer: System: **Format-specific parsers** (tabula, BeautifulSoup, requests), **schema mapping** and normalization, **data quality** validation, **error handling** for malformed data, **incremental** processing, **change detection**, **metadata** preservation, **unified** data model, **transformation** pipelines, **quality** metrics, **monitoring**, and **lineage** tracking.

6. Automated Content Monitoring

Create an automated system to monitor news websites, social media, and forums for specific topics or keywords. How would you handle content changes and ensure comprehensive coverage?

Answer: System: **RSS feed** monitoring, **web scraping** with change detection, **social media APIs**, **keyword** matching with NLP, **content** deduplication, **sentiment** analysis, **real-time** alerts, **historical** tracking, **source** reliability scoring, **false positive** reduction, **dashboard** visualization, and **export** capabilities.

7. Data Quality Framework

Implement a comprehensive data quality framework for sourced data including validation, cleansing, and monitoring. What metrics and processes would you establish?

Answer: Framework: **Completeness** checks (missing values), **accuracy** validation (format/range), **consistency** rules (cross-field), **timeliness** monitoring (freshness), **uniqueness** detection (duplicates), **validity** testing (business rules), **automated** cleansing, **quality** scoring, **trend** analysis, **alerting** thresholds, **remediation** workflows, and **reporting** dashboards.

8. Scalable Scraping Infrastructure

Design infrastructure to scrape thousands of websites daily while managing IP rotation, CAPTCHA solving, and browser automation. What architecture would you use?

Answer: Infrastructure: **Kubernetes** cluster for scaling, **headless browsers** (Playwright/Selenium), **proxy rotation** services, **CAPTCHA** solving APIs, **distributed** task queues (Celery), **load balancing**, **resource** monitoring, **auto-scaling** policies, **failure** recovery, **data** pipelines, **storage** optimization, and **cost** management.

9. API Rate Limit Management

Develop a sophisticated rate limiting system that optimizes API usage across multiple services with different limits and pricing tiers. How would you maximize efficiency while staying within limits?

Answer: System: **Token bucket** algorithms per API, **priority queues** for requests, **usage** forecasting, **cost** optimization, **burst** handling, **quota** monitoring, **automatic** throttling, **request** batching, **caching** strategies, **fallback** APIs, **performance** metrics, and **budget** controls.

10. Geographic Data Collection

Design a system to collect and validate geographic data from multiple sources including APIs, government databases, and crowdsourced platforms. How would

you ensure accuracy and completeness?

Answer: System: **Multi-source** integration (Nominatim, Google Maps, OpenStreetMap), **coordinate** validation, **address** standardization, **geocoding** accuracy scoring, **conflict** resolution, **data** fusion algorithms, **quality** metrics, **change** detection, **validation** workflows, **crowdsource** verification, and **continuous** improvement.

11. Social Media Data Mining

Create a comprehensive social media data collection system that respects platform policies and handles API limitations. What strategies would you implement?

Answer: Strategies: **Official APIs** (Twitter, Facebook, LinkedIn), **rate limit** management, **authentication** handling, **data** filtering and sampling, **privacy** compliance, **terms of service** adherence, **content** moderation, **sentiment** analysis, **trend** detection, **real-time** processing, **historical** data collection, and **ethical** guidelines.

12. Document Processing Pipeline

Design an automated pipeline for processing various document types (PDFs, Word, PowerPoint) to extract structured data. How would you handle different formats and layouts?

Answer: Pipeline: **Format detection**, **OCR** for scanned documents, **layout** analysis, **table** extraction (tabula), **text** extraction (PyPDF2), **image** processing, **metadata** extraction, **content** classification, **quality** assessment, **error** handling, **batch** processing, **progress** tracking, and **output** standardization.

13. Change Detection System

Implement a system that monitors websites for changes and triggers appropriate actions. How would you optimize for efficiency while ensuring comprehensive coverage?

Answer: System: **Content hashing** for change detection, **differential** crawling, **priority** scheduling, **change** classification, **notification** systems, **historical**

tracking, **false positive** reduction, **resource** optimization, **parallel** processing, **storage** efficiency, **monitoring** dashboards, and **alerting** rules.

14. Data Marketplace Integration

Design integration with multiple data marketplaces and vendors with different APIs, formats, and pricing models. How would you manage costs and ensure data quality?

Answer: Integration: **Vendor** abstraction layers, **cost** tracking and optimization, **quality** assessment frameworks, **SLA** monitoring, **contract** management, **usage** analytics, **vendor** comparison, **fallback** strategies, **data** lineage, **compliance** checking, **automated** procurement, and **ROI** analysis.

15. Streaming Data Ingestion

Create a system for ingesting high-volume streaming data from IoT devices, sensors, and real-time feeds. How would you ensure reliability and handle backpressure?

Answer: System: **Apache Kafka** for streaming, **schema** registry, **partitioning** strategies, **consumer** groups, **backpressure** handling, **dead letter** queues, **monitoring** and alerting, **auto-scaling**, **data** validation, **duplicate** handling, **ordering** guarantees, **fault** tolerance, and **performance** optimization.

16. Cross-border Data Collection

Design data collection strategies that work across different countries with varying regulations, languages, and technical infrastructure. What challenges would you address?

Answer: Strategies: **Regulatory** compliance (GDPR, local laws), **multi-language** support, **cultural** considerations, **local** infrastructure adaptation, **proxy** services, **currency** and format handling, **time zone** management, **legal** entity requirements, **data** localization, **translation** services, **local** partnerships, and **compliance** monitoring.

17. Competitive Intelligence System

Build a system for collecting competitive intelligence from public sources while maintaining ethical standards. What data sources and analysis methods would you use?

Answer: System: **Public** data sources (websites, SEC filings, patents), **news** monitoring, **social media** analysis, **job posting** tracking, **pricing** monitoring, **product** launches, **executive** movements, **financial** data, **market** research, **sentiment** analysis, **trend** identification, **competitive** benchmarking, and **ethical** guidelines.

18. Data Freshness Management

Implement a system that ensures data freshness across multiple sources with different update frequencies. How would you optimize collection schedules and handle stale data?

Answer: System: **Adaptive** scheduling based on change patterns, **freshness** scoring, **priority** queues, **incremental** updates, **change** detection, **staleness** alerts, **fallback** mechanisms, **cache** invalidation, **SLA** monitoring, **resource** optimization, **predictive** scheduling, and **quality** metrics.

19. Error Recovery and Resilience

Design error recovery mechanisms for data sourcing pipelines that handle network failures, API outages, and data corruption. What strategies would ensure system resilience?

Answer: Strategies: **Circuit breakers** for failing services, **exponential backoff** with jitter, **dead letter** queues, **retry** policies, **graceful** degradation, **health** checks, **failover** mechanisms, **data** validation, **corruption** detection, **rollback** capabilities, **monitoring** and alerting, **incident** response, and **disaster** recovery.

20. Cost Optimization Framework

Develop a framework for optimizing data sourcing costs across APIs, infrastructure, and human resources. How would you balance cost, quality, and timeliness?

Answer: Framework: **Cost** tracking per source, **usage** optimization, **caching** strategies, **batch** processing, **resource** scheduling, **vendor** negotiation, **alternative** sources evaluation, **quality** vs. cost tradeoffs, **ROI** analysis, **budget** controls, **forecasting**, **automated** cost alerts, and **continuous** optimization.