

# 디지털 인문학 교과서

## Contents

<b>1. Part 1. 디지털 인문학의 지평</b>	<b>1</b>
1.1 디지털 인문학 실습 예제: 텍스트와 코드	1
1.2 방법론적 전환: 읽기에서 채굴로	3
1.3 비판적 디지털 인문학	4
<b>2. Part 2. 컴퓨팅의 원형과 역사</b>	<b>6</b>
2.1 튜링 머신과 폰 노이만	6
2.2 비트와 메모리	7
2.3 네트워크와 웹의 역사	8
<b>3. Part 3. 문자의 디지털 재현</b>	<b>10</b>
3.1 ASCII와 서구 중심주의	10
3.2 한글 인코딩의 역사	11
3.3 유니코드(UTF-8)의 원리	12
<b>4. Part 4. 비정형 텍스트 다루기</b>	<b>14</b>
4.1 플레이인 텍스트의 힘	14
4.2 정규표현식 기초	15
4.3 정규표현식 심화	16
<b>5. Part 5. 구조적 데이터와 데이터베이스</b>	<b>18</b>
5.1 표 형식 데이터(CSV)	18
5.2 관계형 데이터베이스의 이해	19
5.3 SQL 기초	20
5.4 XML과 TEI 마크업	21
<b>6. Part 6. 의미와 연결</b>	<b>23</b>
6.1 시맨틱 웹과 RDF	23
6.2 온톨로지 설계 기초	24
<b>7. Part 7. 데이터 분석 방법론</b>	<b>26</b>
7.1 텍스트 마이닝 프로세스	26
7.2 토픽 모델링	27
7.3 네트워크 분석 이론	28
7.4 공간 정보와 GIS	29
<b>8. Part 8. 인공지능과 인문학</b>	<b>31</b>
8.1 AI 패러다임의 변화	31
8.2 딥러닝 기초	32
8.3 생성형 AI와 LLM	33
<b>9. Part 9. 연구 인프라와 출판</b>	<b>35</b>
9.1 CLI와 리눅스 기초	35
9.2 웹 서버 구축	36
9.3 Git과 협업	37
9.4 Jupyter Book(GitHub Actions)	38
9.5 MyST Markdown 표준 문법 가이드	41

## 1. Part 1. 디지털 인문학의 지평

### 1.1 디지털 인문학 실습 예제: 텍스트와 코드

Colab에서 실행하기

이 장에서는 디지털 인문학 수업에서 자주 사용되는 **파이썬 코드와 실행 결과가 PDF 교재에서 어떻게 표현되는지를 간단한 예제로 살펴본다.**

#### 1.1.1 1. 파이썬 코드 예제

아래는 문자열 데이터를 간단히 처리하는 파이썬 코드이다.

```
text = "Digital Humanities connects computing and the humanities."  
  
words = text.lower().replace(".", "").split()  
word_count = len(words)  
  
word_count
```

실행 결과

7

#### 1.1.2 2. 빈도 계산 예제

이번에는 단어 빈도를 계산해 본다.

```
from collections import Counter  
  
counter = Counter(words)  
counter
```

실행 결과

```
Counter({  
    'digital': 1,  
    'humanities': 1,  
    'connects': 1,  
    'computing': 1,  
    'and': 1,  
    'the': 1,  
    'humanities': 1  
})
```

⚠️ PDF에서는 실행 결과가 “고정폭 텍스트 블록”으로 출력되며, 노트북과 달리 접하지 않고 그대로 노출된다.

#### 1.1.3 3. 표 형태의 출력 예제

```
import pandas as pd
```

```
df = pd.DataFrame(counter.items(), columns=["word", "count"])
df
```

실행 결과 (丑)

word	count
digital	1
humanities	2
connects	1
computing	1
and	1
the	1

#### 1.1.4 4. 시각화 예제

```
import matplotlib.pyplot as plt

plt.bar(df["word"], df["count"])
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

::{note} 이와 같이 PDF 교재에서는

- 코드 arrow.r 실행 결과 arrow.r 해설 구조를 명확히 분리하는 것이 가독성에 유리하다.
- 복잡한 인터랙션은 웹북/Colab에서, 정리된 결과는 PDF 교재에서 제공하는 것이 권장된다.

## 1.2 방법론적 전환: 읽기에서 채굴로

### 1.2.1 방법론적 전환: 읽기에서 채굴로

::{note} 이 챕터는 아직 작성되지 않았습니다.

## **1.3 비판적 디지털 인문학**

### **1.3.1 비판적 디지털 인문학**

::{note} 이 챕터는 아직 작성되지 않았습니다.



## **2. Part 2. 컴퓨팅의 원형과 역사**

### **2.1 투링 머신과 폰 노이만**

#### **2.1.1 투링 머신과 폰 노이만**

::{note} 이 챕터는 아직 작성되지 않았습니다.

## **2.2 비트와 메모리**

### **2.2.1 비트와 메모리**

::{note} 이 챕터는 아직 작성되지 않았습니다.

## **2.3 네트워크와 웹의 역사**

### **2.3.1 네트워크와 웹의 역사**

::{note} 이 챕터는 아직 작성되지 않았습니다.



### **3. Part 3. 문자의 디지털 재현**

#### **3.1 ASCII와 서구 중심주의**

##### **3.1.1 ASCII와 서구 중심주의**

::{note} 이 챕터는 아직 작성되지 않았습니다.

## 3.2 한글 인코딩의 역사

### 3.2.1 한글 인코딩의 역사

::{note} 이 챕터는 아직 작성되지 않았습니다.

### **3.3 유니코드(UTF-8)의 원리**

#### **3.3.1 유니코드(UTF-8)의 원리**

::{note} 이 챕터는 아직 작성되지 않았습니다.



## **4. Part 4. 비정형 텍스트 다루기**

### **4.1 플레인 텍스트의 힘**

#### **4.1.1 플레인 텍스트의 힘**

::{note} 이 챕터는 아직 작성되지 않았습니다.

## 4.2 정규표현식 기초

### 4.2.1 정규표현식 기초

::{note} 이 챕터는 아직 작성되지 않았습니다.

## **4.3 정규표현식 심화**

### **4.3.1 정규표현식 심화**

::{note} 이 챕터는 아직 작성되지 않았습니다.



## **5. Part 5. 구조적 데이터와 데이터베이스**

### **5.1 표 형식 데이터(CSV)**

#### **5.1.1 표 형식 데이터(CSV)**

::{note} 이 챕터는 아직 작성되지 않았습니다.

## 5.2 관계형 데이터베이스의 이해

### 5.2.1 관계형 데이터베이스의 이해

::{note} 이 챕터는 아직 작성되지 않았습니다.

## 5.3 SQL 기초

### 5.3.1 SQL 기초

::{note} 이 챕터는 아직 작성되지 않았습니다.

## **5.4 XML과 TEI 마크업**

### **5.4.1 XML과 TEI 마크업**

::{note} 이 챕터는 아직 작성되지 않았습니다.



## **6. Part 6. 의미와 연결**

### **6.1 시맨틱 웹과 RDF**

#### **6.1.1 시맨틱 웹과 RDF**

::{note} 이 챕터는 아직 작성되지 않았습니다.

## 6.2 온톨로지 설계 기초

### 6.2.1 온톨로지 설계 기초

::{note} 이 챕터는 아직 작성되지 않았습니다.



## **7. Part 7. 데이터 분석 방법론**

### **7.1 텍스트 마이닝 프로세스**

#### **7.1.1 텍스트 마이닝 프로세스**

::{note} 이 챕터는 아직 작성되지 않았습니다.

## 7.2 토픽 모델링

### 7.2.1 토픽 모델링

::{note} 이 챕터는 아직 작성되지 않았습니다.

## 7.3 네트워크 분석 이론

### 7.3.1 네트워크 분석 이론

::{note} 이 챕터는 아직 작성되지 않았습니다.

## 7.4 공간 정보와 GIS

### 7.4.1 공간 정보와 GIS

::{note} 이 챕터는 아직 작성되지 않았습니다.



## **8. Part 8. 인공지능과 인문학**

### **8.1 AI 패러다임의 변화**

#### **8.1.1 AI 패러다임의 변화**

::{note} 이 챕터는 아직 작성되지 않았습니다.

## 8.2 딥러닝 기초

### 8.2.1 딥러닝 기초

::{note} 이 챕터는 아직 작성되지 않았습니다.

## 8.3 생성형 AI와 LLM

### 8.3.1 생성형 AI와 LLM

::{note} 이 챕터는 아직 작성되지 않았습니다.



## **9. Part 9. 연구 인프라와 출판**

### **9.1 CLI와 리눅스 기초**

#### **9.1.1 CLI와 리눅스 기초**

::{note} 이 챕터는 아직 작성되지 않았습니다.

## **9.2 웹 서버 구축**

### **9.2.1 웹 서버 구축**

::{note} 이 챕터는 아직 작성되지 않았습니다.

## **9.3 Git과 협업**

### **9.3.1 Git과 협업**

::{note} 이 챕터는 아직 작성되지 않았습니다.

## 9.4 Jupyter Book(GitHub Actions)

이 챕터에서는 로컬 컴퓨터에서 매번 빌드할 필요 없이, GitHub에 소스를 올리기만 하면 자동으로 웹사이트를 만들어 배포해주는 **GitHub Actions** 설정 방법을 다룹니다.

특히 최신 \*\*MyST 엔진(v2 이상)\*\*을 사용할 때 자주 발생하는 경로 문제(BASE\_URL), 실행 파일 오류(command not found), 그리고 사이트가 빈 화면으로 나오는 문제(Jekyll 충돌)를 방지하는 **표준 설정**을 안내합니다.

### 9.4.1 1. 프로젝트 설정 파일 준비 (`myst.yml`)

최신 Jupyter Book은 `_config.yml` 대신 `myst.yml`을 사용하여 프로젝트를 설정합니다. 배포 시 오류를 막기 위해 **저자 정보**를 상세히 기록해야 합니다.

```
version: 1
project:
  id: dh-textbook
  title: "나의 디지털 인문학 교과서"
  authors:
    # 이름 오류 방지를 위해 given(이름)과 family(성)를 명확히 분리합니다.
    - name:
        given: "바로"
        family: "김"
  toc:
    - file: intro.md
      # 추가 챕터가 있다면 아래 형식으로 추가합니다.
      # - file: deployment.md

site:
  template: book-theme
```

### 9.4.2 2. GitHub 저장소 권한 설정

GitHub Actions가 배포를 수행할 수 있도록 권한을 열어주어야 합니다.

- 저장소 상단의 **Settings** 탭을 클릭합니다.
- 왼쪽 사이드바에서 **Pages** 메뉴를 선택합니다.
- Build and deployment** 섹션의 **Source** 항목을 `Deploy from a branch`에서 `GitHub Actions`로 변경합니다.

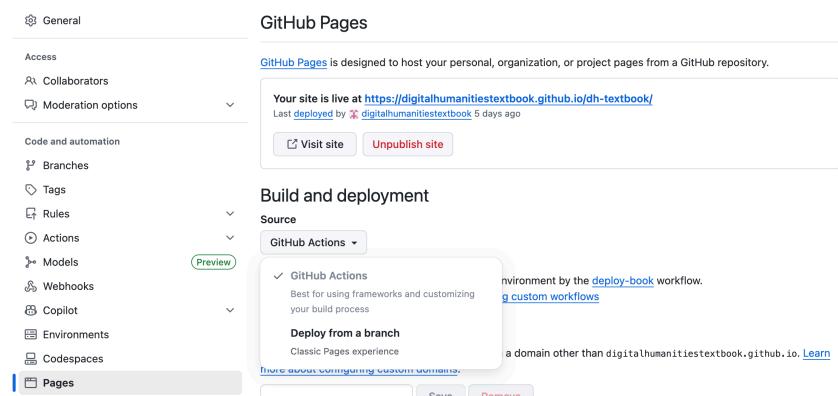


Figure 9.1: GitHub Pages에서 배포 소스를 GitHub Actions로 설정하는 화면

#### 9.4.3 3. 배포 스크립트 작성 (`deploy.yml`)

저장소의 `.github/workflows/` 폴더 안에 `deploy.yml` 파일을 만들고 아래 내용을 작성합니다.

이 스크립트는 \*\*MyST 엔진(mystmd)\*\*을 올바르게 설치하고, GitHub Pages의 특성(Jekyll)으로 인한 오류를 자동으로 해결합니다.

```
name: deploy-book

# main 브랜치에 푸시(Push)될 때마다 실행
on:
  push:
    branches:
      - main

# GitHub Pages 배포를 위한 권한 설정 (필수)
permissions:
  contents: read
  pages: write
  id-token: write

concurrency:
  group: "pages"
  cancel-in-progress: false

jobs:
  deploy:
    environment:
      name: github-pages
      url: ${{ steps.deployment.outputs.page_url }}
    runs-on: ubuntu-latest
    steps:
      - name: Checkout
        uses: actions/checkout@v4

      # 1. Python 설치 (Jupyter Book 의존성)
      - name: Set up Python
        uses: actions/setup-python@v5
        with:
          python-version: "3.11"

      # 2. Node.js 설치 (MyST 엔진 필수 런타임)
      - name: Set up Node.js
        uses: actions/setup-node@v4
        with:
          node-version: "20"

      # 3. 도구 설치
```

```

- name: Install dependencies
  run: |
    pip install jupyter-book
    # [중요] 'myst-cli' 대신 실행 파일이 포함된 'mystmd'를 설치해야 합니다.
    # -g 옵션으로 전역 설치하여 경로 문제를 방지합니다.
    npm install -g mystmd

# 4. 책 빌드 (HTML 생성)
- name: Build the book
  run: |
    # [중요] 저장소 이름(예: /DHTextBook)을 BASE_URL로 지정해야 CSS가 깨지지 않습니다.
    # 본인의 저장소 이름에 맞게 수정하세요.
    export BASE_URL="/DHTextBook"
    myst build --html

# 5. Jekyll 처리 방지 (.nojekyll 생성)
# GitHub Pages는 기본적으로 언더바(_)로 시작하는 폴더를 무시합니다.
# 이 파일이 없으면 _static 폴더가 누락되어 사이트 스타일이 깨집니다.
- name: Disable Jekyll
  run: |
    touch _build/html/.nojekyll

# 6. 결과물 업로드
- name: Upload artifact
  uses: actions/upload-pages-artifact@v3
  with:
    path: "_build/html"

# 7. 실제 배포
- name: Deploy to GitHub Pages
  id: deployment
  uses: actions/deploy-pages@v4

```

#### 9.4.4 4. 트러블슈팅 (자주 묻는 질문)

배포 과정에서 오류가 발생할 경우 다음 사항을 확인하세요.

## 9.5 MyST Markdown 표준 문법 가이드

이 문서는 Jupyter Book의 표준 문법인 \*\*MyST(Markedly Structured Text)\*\*의 핵심 사용법을 정리한 교육용 문서입니다. 일반적인 마크다운 문법에 학술적인 글쓰기를 위한 기능(각주, 인용, 수식, 상호 참조 등)이 추가된 형태입니다.

### 9.5.1 1. 텍스트 강조 (Text Formatting)

기본적인 텍스트 강조는 일반 마크다운과 동일합니다.

- **굵게:** \*\*굵은 텍스트\*\* arrow.r **굵은 텍스트**
- **기울임:** \*기울임 텍스트\* arrow.r **기울임 텍스트**
- **인라인 코드:** `코드` arrow.r 코드
- ~~취소선~~: ~~취소선~~ arrow.r ~~취소선~~

### 9.5.2 2. 블록 인용 및 경고창 (Admonitions)

교과서나 기술 문서에서 중요한 내용을 강조할 때 사용하는 박스 형태의 문법입니다. :::{type} 형식을 사용합니다.

#### 9.5.2.1 기본 노트

```
:::{note}
```

이곳에 노트 내용을 작성합니다.

보충 설명이나 참고 사항을 적을 때 유용합니다.

```
:::
```

#### 9.5.2.2 경고 (Warning)

```
:::{warning}
```

주의가 필요한 내용은 warning을 사용합니다.

```
:::
```

#### 9.5.2.3 팁 (Tip)

```
:::{tip}
```

유용한 팁이나 지름길을 안내할 때 사용합니다.

```
:::
```

#### 9.5.2.4 제목이 있는 박스 (Custom Admonition)

```
:::{admonition} 여기에는 제목이 들어갑니다
```

```
:class: seealso
```

본문 내용은 여기에 작성합니다. `seealso`, `tip`, `warning` 등 클래스를 지정하여 색상을 바꿀 수 있습니다.

```
:::
```

### 9.5.3 3. 이미지와 그림 (Figures)

단순 이미지 삽입을 넘어, 캡션(설명)을 달고 크기를 조절하려면 {figure} 지시어를 사용합니다.

```
:::{figure} [https://mystmd.org/images/logo.png](https://mystmd.org/images/logo.png)
```

```
:name: my-figure-label
```

```
:width: 200px
```

```
:align: center
```

여기에 이미지 캡션(설명)을 적습니다.

```
:::
```

- :name:: 나중에 본문에서 이 그림을 참조할 때 사용하는 ID입니다.
- :width:: 이미지의 너비를 지정합니다 (px 또는 %).
- :align:: 정렬 방식 (left, center, right).

#### 9.5.4 4. 코드 블록 (Code Blocks)

프로그래밍 코드를 작성할 때는 언어를 지정하고, 필요시 줄 번호나 제목을 붙일 수 있습니다.

```
def hello_world():
    print("Hello, Digital Humanities!")
```

옵션을 추가하려면 다음과 같이 작성합니다.

```
```python
:linenos:
:caption: 파이썬 예제 코드

def hello_world():
    print("Hello, Digital Humanities!")
```

```

#### 9.5.5 5. 수식 (Math & LaTeX)

학술적인 수식은 LaTeX 문법을 사용하여 표현합니다.

##### 9.5.5.1 인라인 수식 (문장 중간)

문장 중간에  $E = mc^2$  처럼 수식을 넣으려면 \$ 기호를 사용합니다.

- 입력: \$E = mc^2\$

##### 9.5.5.2 블록 수식 (별도 문단)

수식을 가운데 정렬하여 크게 보여주려면 \$\$ 또는 {math} 지시어를 사용합니다.

```
$$
\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}
$$
```

수식에 번호를 붙여서 참조하려면 다음과 같이 씁니다.

```
:::{math}
:label: my-equation

x = \frac{1}{2} a t^2 + v_0 t + x_0
:::
```

#### 9.5.6 6. 상호 참조 (Cross-Referencing)

문서 내의 다른 챕터, 그림, 수식 등을 링크할 때 사용합니다. 페이지 번호가 바뀌어도 링크가 유지되므로 매우 중요합니다.

### 9.5.6.1 섹션(제목) 참조하기

참조하고 싶은 제목 바로 위에 라벨을 붙입니다.

```
(section-label)=  
## 2. 데이터 분석 방법
```

다른 곳에서 위 섹션을 링크하려면:

- 입력: 2장 {ref}section-label 내용을 참고하세요.
- 출력: 2장 **2. 데이터 분석 방법** 내용을 참고하세요.

### 9.5.6.2 그림/수식 참조하기

그림이나 수식에 :name: 또는 :label:을 붙였다면 {ref} 대신 {numref}를 써서 “그림 1”, “식 2”처럼 번호로 참조할 수 있습니다.

- 입력: {numref}my-figure-label을 보세요.
- 출력: **그림 1**을 보세요.

### 9.5.7 7. 각주 (Footnotes)

본문 내용에 부가 설명을 달 때 사용합니다.

디지털 인문학[^dh]은 인문학 연구에 디지털 기술을 접목한 분야입니다.

[^dh]: Digital Humanities의 약자.

### 9.5.8 8. 탭 (Tab Set)

여러 언어의 코드나, 다양한 옵션을 탭으로 보여주고 싶을 때 사용합니다.

```
:::::{tab-set}  
  
:::{tab-item} Python  
print("Hello Python")  
:::  
  
:::{tab-item} JavaScript  
console.log("Hello JS");  
:::  
  
:::::
```

---

#### 더 자세한 내용 참고:

- MyST 공식 문서 (영문)
- Jupyter Book 갤러리