

인문·사회과학을 위한 파이썬 프로그래밍

제2강 파이썬이란 무엇이고 왜 인문·사회과학에 필요한가



Prof. Byong-Rae Ryu, Dr.phil.
ryu@cnu.ac.kr
<https://ling.cnu.ac.kr/>

Department of Linguistics
Chungnam National University
Daejeon, South Korea

lecture02-week01-part02

학습 목표

- 1 파이썬 프로그래밍언어에 대한 개요를 안다.
- 2 파이썬 프로그래밍언어의 특징을 안다.
- 3 프로그래밍언어로서 파이썬의 위상에 대하여 안다.
- 4 파이썬 프로그래밍언어의 간략한 역사에 대하여 안다.

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

간단한 역사

How to Learn
Python

Outline

- 1 왜 프로그래밍을 배워야 하는가?
- 2 디지털 인문학(Digital Humanities)이란?
- 3 Python이란?
- 4 파이썬의 특징
- 5 프로그래밍 언어로서의 파이썬 현황
 - 파이썬 패키지 인덱스(Python Packages Index)
 - 파이썬으로 무엇을 할 수 있는가?
 - 파이썬 사용 현황
- 6 간단한 역사
- 7 How to Learn Python

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

간단한 역사

How to Learn
Python

왜 프로그래밍을 배워야 하는가? [1/4]

- Steve Jobs: 프로그래밍 교육의 중요성
 - "Everybody in this country should learn how to program a computer... because it teaches you how to think."
 - "이 나라의 모든 사람들은 컴퓨터 프로그래밍을 배워야 합니다. 왜냐하면 그것이 사고하는 방법을 가르쳐주기 때문입니다."
 - 프로그래밍은 단순한 기술 습득을 넘어, 논리적 사고와 문제 해결 능력을 향상시키는 데 큰 도움

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

간단한 역사

How to Learn
Python

왜 프로그래밍을 배워야 하는가? [2/4]

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

간단한 역사

How to Learn
Python

- Bill Gates
 - "Learning to write programs stretches your mind, and helps you think better, creates a way of thinking about things about things that I think is helpful in all domains"
 - "프로그래밍은 사고의 범위를 넓혀주고 더 나은 생각을 할 수 있게 만들며 분야에 상관없이 모든 문제에 대해 새로운 해결책을 생각할 수 있는 힘을 길러줍니다."
- Mark Zuckerberg
 - "If you can code, you can sit down and create something completely new, and that's really powerful."
 - "코딩을 할 수 있다면, 앉은 자리에서 완전히 새로운 것을 만들어낼 수 있습니다. 이는 정말로 강력한 능력입니다."

왜 프로그래밍을 배워야 하는가? [3/4]

- 문과생도 프로그래밍을 해야 하는 이유 (1/2)
 - 게이츠는 잡스와 달리 컴퓨터 코딩을 이해했다. 그의 사고 방식은 잡스보다 더 실용적이고 질서 정연했으며, 분석적 처리 능력이 풍부했다. 반대로 잡스는 직관적이고 낭만적이었다. 그(잡스)는 기술을 유용하게 만들고 디자인에 매력을 불어넣었으며 인터페이스를 친화적으로 만드는 데 소질이 있었다.
 - 월터 아이작슨, <스티브 잡스> 중에서
 - 스티브 워즈니악을 설득해 애플을 공동 창업할 수 있었던 이유는 스티브 잡스가 IT 개발자였던 스티브 워즈니악과 IT 분야에 대해 얘기할 수 있는 IT 지식이 있었기 때문
 - 사용자 관점에서 보다 유용하게 만들고 디자인에 매력을 불어넣는 소질과 안목, 그리고 상품화시키는 능력

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

간단한 역사

How to Learn
Python

왜 프로그래밍을 배워야 하는가? [4/4]

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

간단한 역사

How to Learn
Python

- 문과생도 프로그래밍을 해야 하는 이유 (2/2)
- 코딩을 할 수 있는 개발자가 IT 창업에 유리한 것은 사실
- 코딩 자체를 배우는 것보다 IT의 기본 지식과 기술의 흐름을 익히고, IT 분야와 사업적 아이디어를 접목시켜 비전을 세우며 개발자와 대화하고 설득할 수 있는 IT 역량을 갖추는 것이 더 중요
- 현재는 코딩에 능숙하지 않아도 코딩을 대신해주는 AI를 다룰 수 있는 능력을 갖추는 것이 더 중요해짐

Outline

- 1 왜 프로그래밍을 배워야 하는가?
- 2 디지털 인문학(Digital Humanities)이란?
- 3 Python이란?
- 4 파이썬의 특징
- 5 프로그래밍 언어로서의 파이썬 현황
 - 파이썬 패키지 인덱스(Python Packages Index)
 - 파이썬으로 무엇을 할 수 있는가?
 - 파이썬 사용 현황
- 6 간단한 역사
- 7 How to Learn Python

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

간단한 역사

How to Learn
Python

디지털 인문학(Digital Humanities)이란? [1/4]



- 디지털 인문학(Digital Humanities)
 - 전통적인 인문학 연구와 디지털 기술을 결합하여 새로운 방식으로 인문학을 연구, 분석, 교육, 및 응용하는 학문적 분야
 - 인문학의 전통적인 주제인 역사, 문학, 철학, 언어학, 예술 등을 다루면서, 디지털 기술을 활용해 데이터를 처리하고 새로운 관점을 제시

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

간단한 역사

How to Learn
Python

디지털 인문학(Digital Humanities)이란? [2/4]



- 주요 특징

- 디지털 기술의 활용

- 데이터 마이닝, 텍스트 분석, 네트워크 분석, 데이터 시각화 같은 기술을 이용.
 - 빅데이터, 인공지능(AI), 자연어 처리(NLP) 등의 기술을 연구에 적용.

- 인터디서플리너리(학제 간 융합):

- 컴퓨터 과학, 정보학, 데이터 과학, 인문학이 상호작용.
 - 기술적 도구를 통해 인문학 연구를 더욱 깊이 있게 수행.

- 공유와 협업

- 디지털화된 자료를 온라인으로 공유하여 협업 연구 가능.
 - 연구 결과를 대중과 소통하거나 교육 자료로 활용.

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

간단한 역사

How to Learn
Python

디지털 인문학(Digital Humanities)이란? [3/4]



- 응용 분야 및 예시
 - 텍스트 분석
 - 고전 문학 작품의 단어 빈도, 텍스트 구조 분석.
 - 예: 셰익스피어 작품에서 특정 주제의 변화 추적.
 - 역사 연구
 - 지리정보시스템(GIS)을 활용해 과거 사건을 지도 상에서 시각화.
 - 예: 특정 시기 동안 무역 네트워크의 변화 분석.
 - 언어 연구
 - 자연어 처리 기술로 방대한 문서 데이터 분석.
 - 예: 디지털 아카이브에서 언어 사용 패턴 추적.
 - 예술 및 문화
 - 예술 작품의 디지털 복원, 3D 모델링.
 - 예: 박물관 컬렉션의 온라인 공개 및 가상현실(VR) 투어.
 - 교육
 - 디지털 도구를 활용한 강의 및 학습 자료 제작.
 - 예: 대규모 개방 온라인 강좌(MOOC)에서 역사적 사건의 인터랙티브 지도 활용.

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

간단한 역사

How to Learn
Python

디지털 인문학(Digital Humanities)이란? [4/4]



- 디지털 인문학의 중요성

- 디지털 인문학은 전통적인 인문학을 현대적이고 기술적으로 확장시켜, 더 넓은 관점에서 인간과 사회를 이해할 수 있도록 함.
- 기술과 인문학의 융합은 현대 사회의 복잡한 문제를 해결하고, 과거와 현재를 연결하며, 미래를 준비하는 데 중요한 역할을 수행

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

간단한 역사

How to Learn
Python

Outline

- 1 왜 프로그래밍을 배워야 하는가?
- 2 디지털 인문학(Digital Humanities)이란?
- 3 Python이란?**
- 4 파이썬의 특징
- 5 프로그래밍 언어로서의 파이썬 현황

파이썬 패키지 인덱스(Python Packages Index)

파이썬으로 무엇을 할 수 있는가?

파이썬 사용 현황

- 6 간단한 역사
- 7 How to Learn Python

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

간단한 역사

How to Learn
Python

개요 [1/3]

- Created by Guido van Rossum(귀도 반 로섬, 1956년 1월 31일, Universiteit van Amsterdam)

- Stichting Mathematisch Centrum (CWI, <https://www.cwi.nl>)
- Corporation for National Research Initiatives (CNRI, <https://www.cnri.reston.va.us>)
- BeOpen PythonLabs team at BeOpen.com
- Google
- Dropbox

- Python Software Foundation (PSF, <https://www.python.org/psf/>, 'Benevolent Dictator For Life (BDFL)'(자선 종신 이사))



Figure 1: Guido van Rossum (2014)

개요 [2/3]

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

간단한 역사

How to Learn
Python

```
PS C:\Users\brryu> python
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit
      (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> copyright
Copyright (c) 2001-2023 Python Software Foundation.
All Rights Reserved.

Copyright (c) 2000 BeOpen.com.
All Rights Reserved.

Copyright (c) 1995-2001 Corporation for National Research Initiatives.
All Rights Reserved.

Copyright (c) 1991-1995 Stichting Mathematisch Centrum, Amsterdam.
All Rights Reserved.
>>>
```

개요 [3/3]

- First released on Feb. 20, 1991 (Stable release 3.12.2 (Feb. 2024))
- 파이썬이라는 이름은 Guido van Rossum이 이 프로그램을 개발하면서 즐겨 보았던 영국 코미디 “Monty Python’s Flying Circus”에서 따온 것이라고 스스로 밝히고 있으나, python 그 자체는 “비단뱀”을 의미
- Filename extensions: **.py**, .pyc, .pyd, .pyo (prior to 3.5), .pyw, .pyz (since 3.5)

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

간단한 역사

How to Learn
Python

Outline

- ① 왜 프로그래밍을 배워야 하는가?
- ② 디지털 인문학(Digital Humanities)이란?
- ③ Python이란?
- ④ 파이썬의 특징
- ⑤ 프로그래밍 언어로서의 파이썬 현황

파이썬 패키지 인덱스(Python Packages Index)

파이썬으로 무엇을 할 수 있는가?

파이썬 사용 현황

- ⑥ 간단한 역사
- ⑦ How to Learn Python

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

간단한 역사

How to Learn
Python

프로그래밍언어로서 파이썬의 특징 [1/1]

- A platform-independent, interpreted, high-level, object-oriented programming language for general-purpose programming
 - ① **Platform-independent** (플랫폼 독립적) vs. Platform-dependent: 특정 운영체제에 국한되지 않은 언어
 - ② **Interpreted** (인터프리터식 프로그래밍언어) vs. Compiled: 프로그래머가 작성한 소스코드가 해석기에 의해 직접 실행되는 언어
 - ③ **High-level** (고급 프로그래밍언어) vs. Low-level: 사람이 이해하기 쉽게 작성된 프로그래밍 언어로서, 저급 프로그래밍 언어보다 가독성이 높고 명령어가 직관적이어서 다루기 쉽다.
 - ④ **General purpose** (범용 프로그래밍언어) vs. Targeted purpose: 다양한 도메인의 소프트웨어를 개발하기 위해 설계된 프로그래밍 언어
 - ⑤ **Object-Oriented Programming language(OOP)**(객체 지향 프로그래밍 언어) vs. Procedural Language: 컴퓨터 프로그래밍의 패러다임 중 하나로서, 컴퓨터 프로그램을 명령어의 목록(순차적인 절차)으로 보는 시각에서 벗어나 여러 개의 독립된 단위, 즉 ‘객체’들의 모임으로 파악하고자 하는 패러다임을 의미한다. 각각의 객체는 메시지를 주고받고, 데이터를 처리할 수 있다.

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

간단한 역사

How to Learn
Python

왜 파이썬으로 전산언어학을? [1/2]

- ① 어느 OS에나 설치가 가능하다(모바일 플랫폼인 안드로이드/iOS에서는 (아직) 동작하지 않는다.).
- ② 기술이 공개되어 있고, 누구나 무료로 사용이 가능하다.
- ③ 문법이 간결하고 우아하다.
- ④ 초보자도 바로 배워 사용할 수 있을 정도로 배우기 쉽다.
- ⑤ 강력한 스트링 연산 기능을 가지고 있어서 자연언어의 정보처리에 적합하다.
- ⑥ 자연언어처리를 위한 수 많은 라이브러리들이 구비되어 있어 기능이 강력하고 다양하다.
- ⑦ 웹 크롤링, 웹 프로그래밍, 데이터 분석 등을 수행할 수 있는 다양한 패키지를 사용할 수 있어 확장성이 매우 뛰어나다.
- ⑧ 다양하고 실용적인 프로그램을 새로운 언어를 다시 공부하지 않아도 바로 만들 수 있다.
- ⑨ NASA, Youtube, CERN에서도 사용하는 매우 강력한 언어이고, Deep Learning 이나 Google App을 만들 때에도 사용되는 언어이다.

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

간단한 역사

How to Learn
Python

왜 파이썬으로 전산언어학을? [2/2]

- ⑩ 도움말 문서도 정리가 잘 되어 있으며, 유니코드 문자열을 지원해서 다양한 언어의 문자 처리에도 능하다. 파이썬에서는 프로그램의 문서화가 매우 중시되고 있어 언어의 기본 기능에 포함되어 있다.

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

간단한 역사

How to Learn
Python

Outline

- 1 왜 프로그래밍을 배워야 하는가?
- 2 디지털 인문학(Digital Humanities)이란?
- 3 Python이란?
- 4 파이썬의 특징
- 5 **프로그래밍 언어로서의 파이썬 현황**

파이썬 패키지 인덱스(Python Packages Index)

파이썬으로 무엇을 할 수 있는가?

파이썬 사용 현황

- 6 간단한 역사
- 7 **How to Learn Python**

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

파이썬 패키지 인덱스
(Python Packages
Index)

파이썬으로 무엇을 할 수
있는가?

파이썬 사용 현황

간단한 역사

How to Learn
Python

프로그래밍 언어로서의 파이썬 현황 [1/3]

- Since 2003, Python has consistently ranked in the top 10 most popular programming languages in the TIOBE Programming Community Index
- As of January 2018, it is the fourth most popular language (behind Java, C, and C++), and since 2019 it has consistently ranked in the top 3 most popular programming languages.
- Language of the Year: 2007, 2010, 2018, 2020, 2021¹

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

파이썬 패키지 인덱스
(Python Packages
Index)

파이썬으로 무엇을 할 수
있는가?

파이썬 사용 현황

간단한 역사

How to Learn
Python

프로그래밍 언어로서의 파이썬 현황 [2/3]











Mar 2024	Mar 2023	Change	Programming Language	Ratings	Change
1	1		 Python	15.63%	+0.80%
2	2		 C	11.17%	-3.56%
3	4	▲	 C++	10.70%	-2.59%
4	3	▼	 Java	8.95%	-4.61%
5	5		 C#	7.54%	+0.37%
6	7	▲	 JavaScript	3.38%	+1.21%
7	8	▲	 SQL	1.92%	-0.04%
8	10	▲	 Go	1.56%	+0.32%
9	14	▲	 Scratch	1.46%	+0.45%
10	6	▼	 Visual Basic	1.42%	-3.33%

Figure 2: TIOBE Index for March 2024 (Source: <https://www.tiobe.com/tiobe-index/>)

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

파이썬 패키지 인덱스
(Python Packages
Index)

파이썬으로 무엇을 할 수
있는가?

파이썬 사용 현황

간단한 역사

How to Learn
Python

프로그래밍 언어로서의 파이썬 현황 [3/3]

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

파이썬 패키지 인덱스
(Python Packages
Index)

파이썬으로 무엇을 할 수
있는가?

파이썬 사용 현황

간단한 역사

How to Learn
Python

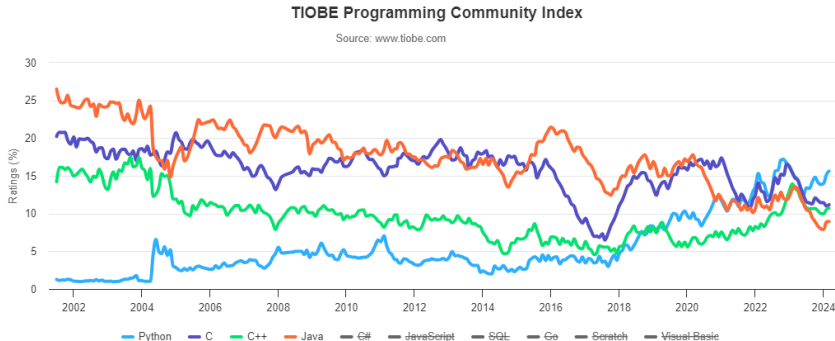


Figure 3: TIOBE Programming Community Index (Source: <http://www.tiobe.com>)

파이썬 패키지 인덱스(Python Packages Index)

- 파이썬에는 프로그래머가 바로 사용할 수 있는 라이브러리와 통합 환경이 이미 배포판과 함께 제공
- “건전지가 포함(Battery Included)”된 전자제품에 비유
- 파이썬 패키지 인덱스(Python Packages Index, PyPI)를 통해 체계적으로 관리: a repository of software for the Python programming language.
- 2024년 3월 기준: 522,592 projects 5,475,619 releases 10,541,385 files 794,131 users

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

파이썬 패키지 인덱스
(Python Packages
Index)

파이썬으로 무엇을 할 수
있는가?

파이썬 사용 현황

간단한 역사

How to Learn
Python

파이썬으로 무엇을 할 수 있는가? [1/6]

- AI and Machine Learning
 - scikit-learn
 - TensorFlow
 - Keras
 - PyTorch
 - XGBoost

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

파이썬 패키지 인덱스
(Python Packages
Index)

파이썬으로 무엇을 할 수
있는가?

파이썬 사용 현황

간단한 역사

How to Learn
Python

파이썬으로 무엇을 할 수 있는가? [2/6]

- Natural Language Processing
 - NLTK
 - Gensim
 - Transformers
 - PyTorch

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

파이썬 패키지 인덱스
(Python Packages
Index)

파이썬으로 무엇을 할 수
있는가?

파이썬 사용 현황

간단한 역사

How to Learn
Python

파이썬으로 무엇을 할 수 있는가? [3/6]

- Data Exploration and Transformation
 - pandas
 - Intake
 - NumPy
 - Dask
 - Apache Airflow

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

파이썬 패키지 인덱스
(Python Packages
Index)

파이썬으로 무엇을 할 수
있는가?

파이썬 사용 현황

간단한 역사

How to Learn
Python

파이썬으로 무엇을 할 수 있는가? [4/6]

- Visualization
 - Matplotlib
 - seaborn
 - Plotly
 - Bokeh
 - HoloViews

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

파이썬 패키지 인덱스
(Python Packages
Index)

파이썬으로 무엇을 할 수
있는가?

파이썬 사용 현황

간단한 역사

How to Learn
Python

파이썬으로 무엇을 할 수 있는가? [5/6]

- GUI and Front-End Development
 - Flask
 - Django
 - Pyramid
 - Tornado
 - CherryPy

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

파이썬 패키지 인덱스
(Python Packages
Index)

파이썬으로 무엇을 할 수
있는가?

파이썬 사용 현황

간단한 역사

How to Learn
Python

파이썬으로 무엇을 할 수 있는가? [6/6]

- R Packages
 - ggplot2
 - dplyr
 - tidyr
 - readr

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

파이썬 패키지 인덱스
(Python Packages
Index)

파이썬으로 무엇을 할 수
있는가?

파이썬 사용 현황

간단한 역사

How to Learn
Python

왜 Python? [1/1]

- ① 어느 OS에나 설치가 가능하다.
- ② 기술이 공개되어 있고, 누구나 무료로 사용이 가능하다.
- ③ 문법이 간결하고 우아하다.
- ④ 컴맹도 배울 수 있다고 해도 과언이 아닐 정도로 배우기 쉽다.
- ⑤ 강력한 스트링 연산 기능을 가지고 있어서 자연언어의 정보처리에 적합하다.
- ⑥ 자연언어처리를 위한 수 많은 라이브러리들이 구비되어 있어 기능이 강력하고 다양하다.
- ⑦ 웹 크롤링, 웹 프로그래밍, 데이터 분석 등을 수행할 수 있는 다양한 패키지를 사용할 수 있어 확장성이 매우 뛰어나다.
- ⑧ 다양하고 실용적인 프로그램을 새로운 언어를 다시 공부하지 않아도 바로 만들 수 있다.
- ⑨ NASA, Youtube, CERN에서도 사용하는 매우 강력한 언어이고, Deep Learning 이나 Google App을 만들 때에도 사용되는 언어이다.

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

파이썬 패키지 인덱스
(Python Packages
Index)

파이썬으로 무엇을 할 수
있는가?

파이썬 사용 현황

간단한 역사

How to Learn
Python

Outline

- ① 왜 프로그래밍을 배워야 하는가?
- ② 디지털 인문학(Digital Humanities)이란?
- ③ Python이란?
- ④ 파이썬의 특징
- ⑤ 프로그래밍 언어로서의 파이썬 현황

파이썬 패키지 인덱스(Python Packages Index)

파이썬으로 무엇을 할 수 있는가?

파이썬 사용 현황

- ⑥ 간단한 역사
- ⑦ How to Learn Python

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

간단한 역사

How to Learn
Python

A. HISTORY OF THE SOFTWARE [1/3]

- `license()` A. HISTORY OF THE SOFTWARE

코드 1: 파이썬 실행 후 `license()` 실행 화면

```
PS C:\Users\brryu> python
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64
    bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> license()
A. HISTORY OF THE SOFTWARE
=====

Python was created in the early 1990s by Guido van Rossum at Stichting
Mathematisch Centrum (CWI, see https://www.cwi.nl) in the Netherlands
as a successor of a language called ABC. Guido remains Python's principal
author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for
```

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

간단한 역사

How to Learn
Python

A. HISTORY OF THE SOFTWARE [2/3]

National Research Initiatives (CNRI, see <https://www.cnri.reston.va.us>)
in Reston, Virginia where he released several versions of the
software.

In May 2000, Guido and the Python core development team moved to
BeOpen.com to form the BeOpen PythonLabs team. In October of the same
year, the PythonLabs team moved to Digital Creations, which became
Zope Corporation. In 2001, the Python Software Foundation (PSF, see
<https://www.python.org/psf/>) was formed, a non-profit organization
created specifically to own Python-related Intellectual Property.
Zope Corporation was a sponsoring member of the PSF.

All Python releases are Open Source (see <https://opensource.org> for
the Open Source Definition). Historically, most, but not all, Python
Hit Return for more, or q (and Return) to quit:

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

간단한 역사

How to Learn
Python

A. HISTORY OF THE SOFTWARE [3/3]



왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

간단한 역사

How to Learn
Python

- Release

- 네덜란드 개발자 귀도 반 로섬(Guido van Rossum)이 1989년에 개발 시작
- 1994년 1.0 Release
- 2000년에 2.0 Release
- 2010년에 2.7 Release
- 2008년에 3.0 Release
- Python 2.7 2020년 1월 공식 지원 종료
- 2022년 3.10 Release
- 2023년 3.11 Release
- 2024년 3.12 Release

간단한 역사 [1/1]

- 프로그래밍언어 계보도

How to Learn Python

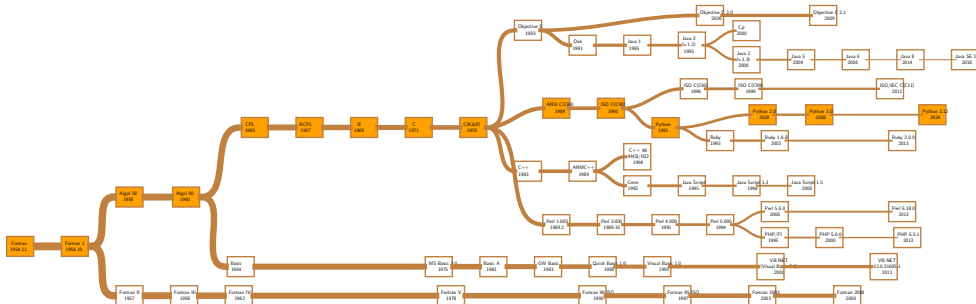


Figure 4: 프로그래밍언어 계보도

Outline

- ① 왜 프로그래밍을 배워야 하는가?
- ② 디지털 인문학(Digital Humanities)이란?
- ③ Python이란?
- ④ 파이썬의 특징
- ⑤ 프로그래밍 언어로서의 파이썬 현황

파이썬 패키지 인덱스(Python Packages Index)

파이썬으로 무엇을 할 수 있는가?

파이썬 사용 현황

- ⑥ 간단한 역사
- ⑦ How to Learn Python

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

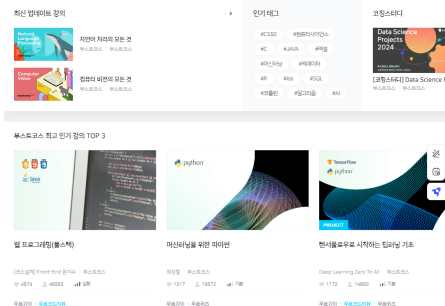
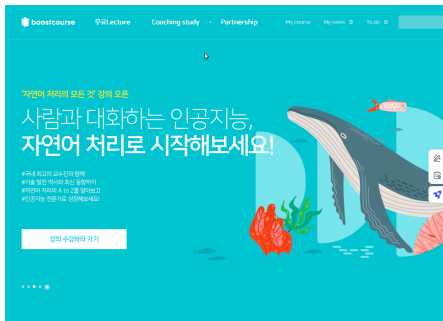
간단한 역사

How to Learn
Python

How to Learn Python [1/2]



- Boostcourse: <https://www.boostcourse.org/>



왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

간단한 역사

How to Learn
Python

How to Learn Python [2/2]

- Python 입문
 - 점프 투 파이썬: <https://wikidocs.net/book/1>
 - 파이썬 - 기본을 갖고 닦자!: <https://wikidocs.net/book/1553>
 - 왕초보를 위한 Python: <https://wikidocs.net/book/2>
- Python 및 프로그래밍에 대하여 조금 익숙할 경우
 - 공학자를 위한 Python: <https://wikidocs.net/book/1704>

왜 프로그래밍을
배워야 하는가?

디지털 인문학
(Digital
Humanities)
이란?

Python이란?

파이썬의 특징

프로그래밍
언어로서의
파이썬 현황

간단한 역사

How to Learn
Python