

## 강의 목차

01 처음 배우는 파이썬

02 왜 파이썬을 배워야 할까요?

03 PYPL 지수란?

04 2025년 전 세계 1위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

# 처음 배우는 파이썬

2026년 중앙대학교

디지털 직무 아카데미

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 처음 배우는 파이썬

04 왜 파이썬을 배워야 할까요?

05 파이썬 라이브러리 파일

06 인터프리터와 디자인 원칙 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

## 강의 목차

01 처음 배우는 파이썬

02 왜 파이썬을 배워야 할까요?

03 PYPL 지수란?

04 2025년 전 세계 1위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

# 왜 파이썬을 배워야 할까요?

수백 개의 프로그래밍 언어 중…

- Java, JavaScript, C++, C#, Go, Rust…
- 왜 하필 파이썬일까요?

데이터가 답을 알려줍니다

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것

02 왜 파이썬을 배워야 할까요?

05 Python 표준 라이브러리

06 인터프리터와 디자인 원칙 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

## 강의 목차

01 처음 배우는 파이썬

02 왜 파이썬을 배워야 할까요?

03 PYPL 지수란?

04 2025년 전 세계 1위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

# PYPL 지수란?

Popularity of Programming Language

## 구글 검색 트렌드 기반 분석

- “Python tutorial” 검색량 측정
- 전 세계 개발자들의 학습 수요 반영
- 매월 업데이트되는 실시간 지표



11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 필요한 만큼은 아니어야 할까요?

05 PEP 8 지수란? 파일

06 인터프리터 장면점 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

## 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 2025년 전 세계 1위 언어

Python = 전 세계 1위 

순위	언어	점유율
1	Python	29.6%
2	Java	15.8%
3	JavaScript	8.9%

→ 2위와 약 2배 차이!

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것

04 왜 대화를 안 해야 할까요?

05 코드를 짠 후에 파일

06 2025년 전 세계를 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

## 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

## 언어별 트렌드 비교

5년간 변화 (2020 → 2025)

언어	변화
Python	+6.1% 
Java	-2.5% 
PHP	-2.1% 

### Python 성장 이유

- AI/머신러닝 붐
- 데이터 분석 필수 도구
- 교육 시장 표준



11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 파이썬은 배워야 할까요?

05 손으로 직접 짜는 컴퓨터 파일

06 인터프리터와 디자인 원칙 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

## 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 프로그래밍 언어 Top 10

순위	언어	점유율	주요 용도
1	Python	29.6%	AI, 데이터, 웹
2	Java	15.8%	엔터프라이즈
3	JavaScript	8.9%	웹 프론트엔드
4	C#	6.7%	게임 (Unity)

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 필요한 것들은 알아야 할까요?

05 코드를 짠 후에 파일

06 인터프리터와 디버거 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

순위	언어	점유율	주요 용도
5	C++	6.3%	시스템

**Python의 강점:** 범용성 (한 언어로 모든 분야)

## 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 파이썬의 탄생

## Guido van Rossum (구도 반 로섬)

- 네덜란드 프로그래머
- 1990년 크리스마스 휴가 중 개발
- 목표: “**읽기 쉽고, 배우기 쉬운 언어**”

## 역사

- 1991년: 첫 공개
- 2000년: Python 2.0
- 2008년: Python 3.0 (대대적 개선)

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터를 짜면 더 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

# 강의 목차

01 처음 배우는 파이썬

02 왜 파이썬을 배워야 할까요?

03 PYPL 지수란?

04 2025년 전 세계 1위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

## 이름의 유래

**영국 BBC 코미디 쇼 (1969-1974)**

Monty Python's Flying Circus

- 귀도가 가장 좋아하던 프로그램
- 재미있고 친근한 이미지

“짧고, 독특하며, 약간 신비로운 이름이 필요했다.”

**뱀(Snake)이 아니라 코미디 쇼가 원조!**



11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터를 찾는 법 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 로고의 의미

## 두 마리 뱀

 파란색 뱀 +  노란색 뱀

## 의미

- 파란색: Python 2.x
- 노란색: Python 3.x
- 서로 감싸는 형태 = 커뮤니티 협력

## 그리스 신화

파르나소스 산의 큰 뱀 Python (강력함, 지혜)

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 필요한 것들은 알아야 할까요?

05 코드를 짠 후에 파일

06 인터프리터와 디자인 위언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

## 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# The Zen of Python

```
>>> import this
```

## 핵심 격언

**Beautiful is better than ugly.**  
아름다운 것이 추한 것보다 낫다.

**Simple is better than complex.**  
단순한 것이 복잡한 것보다 낫다.

**Readability counts.**  
가독성은 중요하다.

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터가 어떤 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

## 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 아름다운 것이 추한 것보다 낫다

## 아름다운 코드

```
def calculate_total_price(items):  
    """상품 목록의 총 가격 계산"""  
    return sum(item.price for  
    item in items)
```

## 추한 코드

```
def calc(x):  
    t=0  
    for i in x:
```



11

아름다운 것이 추한 것보다 낫다

01-01

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓고 빠진

04 왜 대화만 하면 안 돼야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터에 대한 이해 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

```
t=t+i.price  
return t
```

**차이:** 명확한 의도, 가독성

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 단순함이 복잡함보다 낫다

## 단순함

```
squared = [n ** 2 for n in  
numbers]
```

## 불필요한 복잡함

```
class NumberProcessor:  
    def apply_operation(self,  
operation):  
        return [operation(n) for  
n in self.numbers]  
  
processor = NumberProcessor([1,
```

11 아름다운 것이 추한 것보다 낫다  
12 간이 모자

12 단순함이 복잡함보다 낫다

03 차를 살 때는 둘째 짜증

04 왜 이런 문제를 안 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터 장면점 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

```
2, 3])  
squared =  
processor.apply_operation(lambda  
x: x ** 2)
```

**YAGNI**: You Aren't Gonna Need It

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 가독성은 중요하다

## 가독성 낮음

```
def f(x):return sum([i**2 for i  
in x if i%2==0])
```

## 가독성 높음

```
def  
sum_of_even_squares(numbers):  
    """짝수의 제곱 합을 계산"""  
    even_numbers = [n for n in  
numbers if n % 2 == 0]  
    squared = [n ** 2 for n in  
even_numbers]  
    return sum(squared)
```



11

아름다운 것이 추한 것보다 낫다

## 강의 목차

12

단순함이 복잡함보다 낫다

03

처음 배우는 파이썬

04

왜 파이썬을 배워야 할까요?

05

파이썬 초기 스크립트 파일

06

언어학적 면밀한 면접 위 언어

05

언어별 트렌드 비교

06

프로그래밍 언어 Top 10

07

파이썬의 탄생

08

이름의 유래

09

로고의 의미

10

The Zen of Python

## 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 인터프리터 언어

코드를 한 줄씩 즉시 실행

```
>>> 2 + 2  
4  
>>> name = "Python"  
>>> print(f"Hello, {name}!")  
Hello, Python!
```

**REPL**: Read-Eval-Print Loop  
→ 즉각적인 피드백, 학습과 디버깅에 유리

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것

04 왜 API 허용을 배워야 할까요?

05 API 허용 스크립트 파일

06 인터프리터 장면 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 스크립트 vs 컴파일

## Python (스크립트)

코드 작성 → 즉시 실행

```
print("Hello, World!")
```

```
$ python hello.py  
Hello, World!
```

## C++ (컴파일)

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 필요한 것들은 알아야 할까요?

05 Python 속도 파일

06 인터프리터 장단점 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

코드 작성 → 컴파일 → 실행 파일 → 실행

```
$ g++ hello.cpp -o hello  
$ ./hello  
Hello, World!
```

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 인터프리터 장단점

항목	Python	C++
개발 속도	⚡ 빠름	🐢 느림
실행 속도	🐢 느림	⚡ 빠름 (10~100배)
디버깅	✓ 쉬움	✗ 어려움
메모리 관리	✓ 자동 (GC)	✗ 수동

## 결론

80%의 경우: 개발 속도 > 실행 속도

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것

04 왜 필요한 것에 대해서 알아야 할까요?

05 코드를 짠 후에는 파일

06 2025년 전망과 위언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 기계어의 난해함

## 기계어 (0과 1)

```
10110000 01100001  
10110011 00000100  
00000011 00011000
```

**의미**: “숫자 97을 저장하고, 4를 저장하고,  
두 값을 더하라”

## 추상화 계층

```
고급 언어 (Python) ← 사람  
↓  
중급 언어 (C)  
↓  
저급 언어 (어셈블리)
```



11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것

04 왜 필요한 것에 대해서 알아야 할까요?

05 코드를 짠 후에 파일

06 인터프리터와 디버거 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

↓  
기계어 (0101) ← 컴퓨터

## 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 어셈블리어의 복잡성

피보나치 수열 (10번째)

어셈블리 (17줄)

```
mov ecx, [n]
mov eax, 0
mov ebx, 1
fib_loop:
    cmp ecx, 0
    je done
    mov edx, eax
    add eax, ebx
    ...
    ...
```

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 샀을 때 놓고 왔다

04 왜 디버깅을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터를 정답점 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

```
def fibonacci(n):
    a, b = 0, 1
    for _ in range(n):
        a, b = b, a + b
    return a
```

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 파이썬의 간결함

## 리스트 필터링

### Java

```
List<Integer> evens = new  
ArrayList<>();  
for (Integer num : numbers) {  
    if (num % 2 == 0) {  
        evens.add(num);  
    }  
}
```

### Python

```
evens = [n for n in numbers if n  
% 2 == 0]
```



11 아름다운 것이 추한 것보다 낫다

→ 1줄로 끝!

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것

04 왜 대화를 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터 장면 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 고수준 vs 저수준

수준	언어	특징
고수준	Python	생산성 ↑, 속도 ↓
중수준	C	균형
저수준	Assembly	생산성 ↓, 속도 ↑

## 메모리 관리 예시

### C (수동)

```
int* arr = (int*)malloc(5 *  
sizeof(int));  
free(arr); // 필수!
```

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓고 빠진 것

04 왜 대화만 하면 안 돼야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터에 대한 이해

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

## Python (자동)

```
arr = [1, 2, 3, 4, 5]  
# 자동으로 메모리 관리
```

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 변수 다루기

## 언어별 선언 방식

### Python

```
age = 25
```

### JavaScript

```
let age = 25;
```

### Java

```
int age = 25;
```

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓고 빠진

04 왜かい면을 안 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터를 짜면 점 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

- **정적 타이핑**: Java, C++ (타입 미리 선언)
- **동적 타이핑**: Python, JS (타입 자동 추론)

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# Python의 변수

## 동적 타이핑

```
x = 10          # int  
x = "hello"    # str  
x = [1, 2, 3]  # list
```

**장점:** 빠른 프로토타이핑

**단점:** 런타임 에러 가능

## 다양한 데이터 타입

```
name = "개발자"          # 문자열  
(str)  
age = 25                 # 정수  
(int)  
height = 175.5            # 실수  
(float)
```

← 22 / 68 →

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓고 빠진 것

04 왜 대화만을 안 해야 할까요?

05 코드를 짜는 걸 좋아합니다

06 인터프리터 장면 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

```
is_student = True          # 불린  
(bool)  
fruits = ["사과", "바나나"]  # 리스트  
                           # list  
person = {"name": "김파이",  
          "age": 30}           # 딕셔너리 (dict)
```

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 변수 명명 규칙

## snake\_case (Python)

```
# Good
user_name = "Alice"
total_price = 1000
is_valid_email = True
```

```
# Bad
userName = "Alice"      #
camelCase
UserName = "Alice"      #
PascalCase
```

## 상수는 대문자

```
MAX_CONNECTIONS = 100
API_KEY = "abc123"
```

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓고 봐야 하는 것

04 왜 필요한 것들은 알아야 할까요?

05 코드를 짠 후에 파일

06 인터프리터가 어떤 점 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

PI = 3.14159

## 타입 확인

```
age = 25  
print(type(age)) # <class  
'int'>
```

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 문자열 포매팅

## f-string (Python 3.6+)

```
name = "개발자"  
age = 25  
  
# 기본 사용  
print(f"이름: {name}, 나이: {age}세")  
  
# 표현식  
print(f"내년: {age + 1}살")  
  
# 포맷 지정  
price = 1234.5  
print(f"가격: {price:.2f}원") # 1234.50원
```

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓고 빠진 것

04 왜 대화만 하면 안 돼야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터에 대한 이해

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

## 실무 활용

```
for student in students:  
    print(f"{student.name}의 점  
수: {student.score}")
```

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 언어별 비교

## 문자열 보간

### Python

```
print(f"이름: {name}, 나이:  
{age}")
```

### JavaScript

```
console.log(`이름: ${name}, 나이:  
${age}`);
```

### Bash

```
echo "이름: $name, 나이: $age"
```

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터를 찾는 법 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 데이터 타입 상세

## 기본 데이터 타입

```
# 숫자  
integer = 42  
floating = 3.14
```

```
# 문자열  
text = "Hello"  
multiline = """  
여러 줄  
문자열  
"""
```

```
# 불린  
is_active = True  
is_done = False
```

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 샀을 때 놓고 왔다

04 왜 대화만 하면 안 돼야 할까요?

05 코드를 짠 후에는 파일

06 인터프리터에 대한 이해 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

## 컬렉션

```
# 리스트 (순서 0, 변경 가능)
fruits = ["사과", "바나나", "체리"]
```

```
# 튜플 (순서 0, 변경 불가)
coordinates = (10, 20)
```

```
# 딕셔너리 (키-값 쌍)
user = {"name": "김파이", "age": 20}
```

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 제어문: 조건문

## if-elif-else

```
score = 85

if score >= 90:
    print("A학점")
elif score >= 80:
    print("B학점")
elif score >= 70:
    print("C학점")
else:
    print("재수강")
```

## 비교 연산자

```
age = 20
```

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것

04 왜 필요한 것에 대해서 알아야 할까요?

05 코드를 짜는 기본 파일

06 인터프리터와 디버거 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

```
if 18 <= age < 65:  
    print("성인입니다")
```

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

## 제어문: 반복문

### for 반복문

```
# 리스트 순회  
fruits = ["사과", "바나나", "체리"]  
for fruit in fruits:  
    print(fruit)  
  
# 범위 반복  
for i in range(5):  
    print(i) # 0, 1, 2, 3, 4
```

### while 반복문

```
count = 0  
while count < 5:
```



11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터에 대한 이해

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

```
print(count)  
count += 1
```

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 함수 정의

## 함수 기본

```
def greet(name):  
    """인사말을 출력하는 함수"""  
    return f"안녕하세요, {name}님!"  
  
message = greet("파이썬")  
print(message) # 안녕하세요, 파이썬  
님!
```

## 기본값 매개변수

```
def calculate_price(price,  
tax_rate=0.1):  
    """세금 포함 가격 계산"""  
    return price * (1 +  
tax_rate)
```

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것

04 왜 필요한 것에 대해서 알아야 할까요?

05 코드를 짠 후에 파일

06 인터프리터에 대한 이해

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

```
print(calculate_price(1000))  
# 1100.0  
print(calculate_price(1000,  
0.2)) # 1200.0
```

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 리스트 컴프리헨션

Python의 강력한 기능

## 기본 for 반복문

```
squares = []
for i in range(10):
    squares.append(i ** 2)
```

## 리스트 컴프리헨션

```
squares = [i ** 2 for i in
range(10)]
```

## 조건 필터링

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 샀을 때 놓고 왔다

04 왜 필요한지를 알아야 할까요?

05 코드를 짠 후에 파일

06 인터프리터 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

```
# 짝수만  
evens = [i for i in range(10) if  
i % 2 == 0]  
# [0, 2, 4, 6, 8]
```

```
# 문자열 변환  
names = ["alice", "bob",  
"charlie"]  
upper_names = [name.upper() for  
name in names]
```

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 딕셔너리 활용

## 딕셔너리 기본

```
student = {  
    "name": "김파이",  
    "age": 20,  
    "major": "컴퓨터공학"  
}
```

```
print(student["name"]) # 김파이  
print(student.get("grade", "미등록")) # 미등록
```

## 딕셔너리 순회

```
scores = {"국어": 90, "영어": 85, "수학": 95}
```



11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 샀을 때 놓고 왔다

04 왜 이런 문제를 애매해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터에 대한 이해

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

```
for subject, score in scores.items():
    print(f'{subject}: {score}점')
```

## 딕셔너리 컴프리헨션

```
squared = {x: x**2 for x in range(5)}
# {0: 0, 1: 1, 2: 4, 3: 9, 4:}
```

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 파일 입출력

## 파일 읽기

```
# with 문 사용 (권장)
with open('data.txt', 'r',
encoding='utf-8') as f:
    content = f.read()
    print(content)
# 자동으로 파일 닫힘
```

## 파일 쓰기

```
with open('output.txt', 'w',
encoding='utf-8') as f:
    f.write("첫 번째 줄\n")
    f.write("두 번째 줄\n")
```

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓고 빠졌다

04 왜 대화만을 안 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터에 대한 이해

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

## 줄 단위 읽기

```
with open('data.txt', 'r') as f:  
    for line in f:  
        print(line.strip())
```

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 예외 처리

## try-except

```
try:  
    number = int(input("숫자 입력:"))  
    result = 10 / number  
    print(f"결과: {result}")  
except ValueError:  
    print("숫자를 입력해주세요!")  
except ZeroDivisionError:  
    print("0으로 나눌 수 없습니다!")
```

## finally

```
try:  
    f = open('file.txt', 'r') ← 33/68  
    content = f.read()  
finally: →
```

11 아름다운 것이 추한 것보다 낫다  
12 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓고 봐야 하는 것

04 왜 이런 문제를 안 해야 할까요?

05 코드를 짠 후에 파일

06 인터프리터 장면점 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

```
except FileNotFoundError:  
    print("파일이 없습니다")  
finally:  
    f.close() # 항상 실행
```

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 실습: 피보나치 수열

## 반복문 방식

```
def fibonacci_iterative(n):  
    """n번째 피보나치 수를 반환"""  
    a, b = 0, 1  
    for _ in range(n):  
        a, b = b, a + b  
    return a  
  
print(fibonacci_iterative(10))  
# 55
```

## 재귀 방식

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 처음 쌩은 둘째 짜릿

04 왜 필요한 걸 알아야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

```
def fibonacci_recursive(n):
    if n <= 1:
        return n
    return fibonacci_recursive(n-1) +
           fibonacci_recursive(n-2)

print(fibonacci_recursive(10))
# 55
```

## 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 라이브러리 실습: requests

## HTTP 요청

```
import requests

# GET 요청
response =
    requests.get('https://api.github
.com')
print(response.status_code)  # 200
print(response.json())

# POST 요청
data = {"username": "python5/68",
        "password": "1234"}
response =
```



11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 샀을 때 놓고 왔다

04 왜 API 만드는 게 좋아야 할까요?

05 API 템플릿 속도 컴파일

06 인터프리터 장점 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

```
requests.post('https://api.example.com/login',
               json=data)
```

## 실무 활용

```
# 날씨 API 호출 예시
url =
  'https://api.openweathermap.org/
```

## 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 풍부한 라이브러리

풍부한 라이브러리 생태계

**“바퀴를 다시 발명하지 마라”**

```
# 직접 구현: 수백 줄  
# 라이브러리 사용: 1줄  
import requests  
data =  
requests.get('https://api.example.com').json()
```

**PyPI: 64만+ 프로젝트**

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터 장면 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

## 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# PyPI 현황

2025년 기준

지표	수치
프로젝트	640,000+
배포 파일	7,000,000+
일일 다운로드	10억+

## 설치 방법

```
pip install requests pandas  
numpy
```

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터에 대한 이해

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

## 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# Pandas 기초

## DataFrame 생성

```
import pandas as pd

# 딕셔너리에서 생성
data = {
    'name': ['Alice', 'Bob',
    'Charlie'],
    'age': [25, 30, 35],
    'city': ['Seoul', 'Busan',
    'Incheon']
}
df = pd.DataFrame(data)
print(df)
```

## 데이터 확인



11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 샀을 때 놓고 왔다

04 왜 필요한 걸 알아야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터에 대한 이해

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

```
print(df.head())      # 처음 5행  
print(df.info())     # 데이터 정보  
print(df.describe()) # 통계 요약
```

## 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# Pandas CSV 처리

## CSV 읽기/쓰기

```
import pandas as pd  
  
# CSV 파일 읽기  
df = pd.read_csv('sales.csv')  
  
# 데이터 필터링  
high_sales = df[df['sales'] > 1000]  
  
# 그룹화  
by_category = df.groupby('category')  
['sales'].sum()
```

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓고 빠진

04 왜 대화만을 안 해야 할까요?

05 연결된 숨은 컴파일

06 인터프리터 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

```
# CSV로 저장  
high_sales.to_csv('high_sales.cs  
v', index=False)
```

## 엑셀 파일

```
# 엑셀 읽기  
df =  
pd.read_excel('report.xlsx',  
sheet_name='Sheet1')
```

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# NumPy 배열 연산

## 배열 생성

```
import numpy as np

# 1차원 배열
arr = np.array([1, 2, 3, 4, 5])

# 2차원 배열
matrix = np.array([[1, 2, 3],
                  [4, 5, 6]])

# 특수 배열
zeros = np.zeros(5)
# [0, 0, 0, 0]
ones = np.ones(5)
# [1, 1, 1, 1]
```

← 40 / 68 →

11 아름다운 것이 추한 것보다 낫다  
12 단순함이 복잡함보다 낫다  
**강의 목차**

13 차를 샀을 때 놓고 왔어요?

04 왜 필요한지를 알아야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터에 대한 이해 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

```
range_arr = np.arange(10) # [0, 1, ..., 9]
```

## 벡터화 연산

```
arr = np.array([1, 2, 3, 4, 5])
print(arr * 2)      # [2, 4, 6, 8, 10]
print(arr ** 2)    # [1, 4, 9, 16, 25]
```

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# Matplotlib 시작화

## 선 그래프

```
import matplotlib.pyplot as plt  
  
x = [1, 2, 3, 4, 5]  
y = [1, 4, 9, 16, 25]  
  
plt.plot(x, y)  
plt.xlabel('X축')  
plt.ylabel('Y축')  
plt.title('제곱 그래프')  
plt.show()
```

## 막대 그래프

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 샀을 때 놓고 왔던 것

04 왜 대화만 하면 안 돼야 할까요?

05 코드를 짠 후에 파일

06 인터프리터 창에서 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

```
categories = ['A', 'B', 'C',  
              'D']  
values = [23, 45, 56, 78]
```

```
plt.bar(categories, values)  
plt.title('카테고리별 값')  
plt.show()
```

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# NumPy & Pandas

## NumPy - 수치 계산

```
import numpy as np  
arr = np.array([1, 2, 3, 4, 5])  
print(arr.mean()) # 3.0
```

## Pandas - 데이터 분석

```
import pandas as pd  
df = pd.read_csv('sales.csv')  
print(df.groupby('category').sum()  
()
```

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터를 찾는 법 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 데이터 시각화

## Matplotlib (기본)

```
import matplotlib.pyplot as plt  
plt.plot([1, 2, 3, 4], [1, 4, 9,  
16])  
plt.show()
```

## Seaborn (통계)

```
import seaborn as sns  
sns.heatmap(df.corr(),  
annot=True)
```

## Plotly (인터랙티브)

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터 장면 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# AI/머신러닝

## 주요 프레임워크

### Scikit-learn - 전통 ML

```
from sklearn.ensemble import  
RandomForestClassifier  
model = RandomForestClassifier()  
model.fit(X_train, y_train)
```

### TensorFlow - 구글 (프로덕션)

### PyTorch - 메타 (연구)

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터 장면 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 웹 프레임워크

## Django - 풀스택

- Instagram, Pinterest 사용
- Admin 패널 내장

## FastAPI - 최신 고성능

```
from fastapi import FastAPI  
app = FastAPI()  
  
@app.get('/users/{id}')  
async def get_user(id: int):  
    return {"user_id": id}
```



11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터를 찾는 법 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 자연어 처리

## Hugging Face

```
from transformers import  
pipeline  
  
classifier =  
pipeline("sentiment-analysis")  
result = classifier("I love  
Python!")  
# [{"label": "POSITIVE",  
'score': 0.9998}]
```

## NLTK, spaCy - 전통적 NLP

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터를 찾던 점 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

## 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 2025 트렌드: Polars

고성능 데이터 처리

- Rust로 작성
- **Pandas보다 10~100배 빠름**
- 멀티코어 활용

```
import polars as pl
df =
pl.read_csv('large_file.csv')
result =
df.group_by('category').agg(pl.col('sales').sum())
```

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터 장면 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

## 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 2025 트렌드: LangChain

LLM 애플리케이션 개발

```
from langchain.chat_models  
import ChatOpenAI  
  
llm =  
ChatOpenAI(temperature=0.7)  
response = llm("파이썬이란?")
```

## 주요 기능

- ChatGPT, Claude 연동
- RAG (검색 증강 생성)
- 에이전트 시스템

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터 장면 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

## 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 2025 트렌드: Ruff

초고속 린터/포매터

- Rust로 작성
- **Flake8보다 120배 빠름**
- Black, isort 통합

```
pip install ruff
ruff check .
ruff format .
```

2026년 업계 표준

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터 장면 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 업무 자동화 예제

## 엑셀 자동화

```
import openpyxl  
  
wb =  
openpyxl.load_workbook('sales.xlsx')  
sheet = wb.active  
  
# 데이터 읽기  
for row in  
sheet.iter_rows(min_row=2,  
values_only=True):  
    name, sales = row  
    if sales > 1000:  
        print(f'{name}: 목표 달성!')
```

50/68



11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 샀을 때 놓고 왔다

04 왜 대화만 하면 안 돼야 할까요?

05 코드를 짠 후에 파일

06 인터프리터 창에서 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

```
# 데이터 쓰기  
sheet['A1'] = '이름'  
sheet['B1'] = '매출'  
wb.save('updated.xlsx')
```

## 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

## 실무 활용 분야

### 10대 활용 분야

- 1. 데이터 분석
- 2. AI/머신러닝
- 3. 웹 개발
- 4. 업무 자동화
- 5. 게임 개발
- 6. IoT
- 7. 시스템 관리
- 8. 금융 (알고리즘 트레이딩)
- 9. 과학 연구
- 10. 교육

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터 장면 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 활용: 데이터 분석

## 주식 데이터

```
import yfinance as yf  
stock = yf.download('AAPL',  
start='2020-01-01')  
stock['MA_20'] =  
stock['Close'].rolling(20).mean()  
)
```

## 실무 예시

- 금융: 리스크 분석
- 마케팅: A/B 테스트
- 영업: 판매 예측

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터를 찾는 법 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 활용: AI

## 이미지 분류

```
from tensorflow.keras.applications  
import ResNet50  
model =  
ResNet50(weights='imagenet')  
predictions =  
model.predict(img_array)
```

## 실무

- Netflix: 영화 추천
- Tesla: 자율주행
- 병원: 질병 진단



11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터 장면 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 활용: 웹 개발

## 유명 사이트

### Instagram

- Django 기반
- 월간 20억 사용자

### YouTube

- Python 서비스

### Dropbox

- 데스크톱 + 서버

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짠 후에 파일

06 인터프리터 장면 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 활용: 업무 자동화

## 엑셀 자동화

```
import openpyxl  
wb =  
openpyxl.load_workbook('report.xlsx')  
sheet = wb.active  
for row in  
sheet.iter_rows(min_row=2,  
values_only=True):  
    name, sales = row  
    if sales > 1000:  
        print(f'{name}: 목표 달성!')
```

**효과:** 수작업 10시간 → 스크립트 15분/68 →

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터 장면 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 생산성 비교

## HTTP 요청

### Java - 10+ 줄

```
URL url = new  
URL("https://api.example.com");  
HttpURLConnection conn = ...  
BufferedReader in = ...  
// 계속...
```

### Python - 1줄

```
data =  
requests.get('https://api.example.com').json()
```

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸까?

06 인터프리터와 디자인 패턴 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

→ 개발 시간 50% 단축

## 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 메타인지 학습법

“내가 무엇을 알고 모르는지 아는 능력”

## 자가 점검

- 이 코드를 설명할 수 있는가?
- 왜 이렇게 동작하는지 이해했는가?
- 다른 문제에 적용할 수 있는가?

“모른다”를 인정하는 것이 시작

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짠 후에 파일

06 인터프리터와 디자인 위언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 능동적 학습

## ✓ 효과적

- 직접 타이핑하기
- 에러 메시지 읽기
- 코드 변형해보기

## ✗ 비효율적

- 영상만 보기
- 코드 복붙하기
- 이해 안 돼도 넘어가기

에러 = 학습 기회

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터 장면 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

## 꿀팁 1: 완벽주의 버리기

### ✗ 완벽주의자

“모든 문법을 다 배운 후에 시작해야지”

### ✓ 실용주의자

“일단 작동하게 만들자. 개선은 나중에”

완벽한 코드는 없다

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터 장면 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

## 꿀팁 2: 작은 성공

```
# Day 1
print("Hello, World!") # 성공!
✓

# Day 2
name = input("이름: ")
print(f"Hello, {name}!") # 성공!
✓

# Day 3
for i in range(5):
    print(i) # 성공!
✓
```

성취감 = 동기 유지



11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터 장면 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

## 꿀팁 3: 꾸준함

### 하루 30분 vs 주말 8시간

하루 30분 × 30일 = 15시간 (꾸준함)  
주말 8시간 × 1일 = 8시간 (벼락치기)

### 꾸준함이 이김

- 장기 기억 형성
- 습관화
- 지속 가능

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터 장면 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 학습 자료

## 무료 온라인

- 점프 투 파이썬 (wikidocs.net)
- 생활코딩 Python
- Codecademy

## 실습 플랫폼

- Replit.com (브라우저 IDE)
- Kaggle (데이터 분석)
- LeetCode (알고리즘)

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터 장면 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

## 결론

외계어가 아닌, 표현 수단

```
# 사고: "짝수만 필터링"  
evens = [n for n in numbers if n  
% 2 == 0]  
  
# 사고: "파일 읽기"  
with open('data.txt') as f:  
    data = f.read()
```

→ 영어에 가까운 문법

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터 장면 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

## 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 파이썬을 선택한 이유

## 5가지 핵심

-  **전 세계 1위** (29.6%)
-  **풍부한 생태계** (64만+ 라이브러리)
-  **빠른 개발** (간결한 문법)
-  **다양한 활용** (AI, 웹, 데이터…)
-  **활발한 커뮤니티** (무한한 학습 자료)



11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터 장면 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 다음 단계

## 학습 로드맵

### 1단계: 기초 (1~2주)

- 변수, 데이터 타입
- 조건문, 반복문
- 함수

### 2단계: 중급 (2~4주)

- 리스트/딕셔너리
- 파일 I/O
- 클래스/객체

### 3단계: 실전 (4주~)

11

아름다운 것이 추한 것보다 낫다

## 강의 목차

12

단순함이 복잡함보다 낫다

03

처음 쌩은 둘째 짜릿

04

왜 대입 문제를 애해해야 할까요?

05

△△△△△ 솔직 파일

06

인구 25년 전장 대전 위 언어

05

언어별 트렌드 비교

06

프로그래밍 언어 Top 10

07

파이썬의 탄생

08

이름의 유래

09

로고의 의미

10

The Zen of Python

- 라이브러리 활용
- 미니 프로젝트
- 특정 분야 집중

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 첫 걸음

## Hello, Python!

```
# 1. Python 설치  
# python.org 다운로드  
  
# 2. REPL 열기  
# 터미널에서 python 입력  
  
# 3. 첫 코드  
>>> print("Hello, Python!")  
Hello, Python!  
  
# 4. 다음 도전  
>>> name = input("이름: ")  
>>> print(f"안녕하세요, {name}님!")
```

11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터 장면 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

## 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 성공하는 프로그래머의 태도

## 5가지 태도

1. **호기심**: “이건 어떻게 작동할까?”
2. **인내심**: “에러 100번? 괜찮아”
3. **꾸준함**: “매일 30분”
4. **공유**: “배운 것 설명하기”
5. **즐거움**: “재미있어야 계속한다”

**프로그래머 = 문제 해결사**



11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터 장면 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

# 강의 목차

- 01 처음 배우는 파이썬
- 02 왜 파이썬을 배워야 할까요?
- 03 PYPL 지수란?
- 04 2025년 전 세계 1위 언어
- 05 언어별 트렌드 비교
- 06 프로그래밍 언어 Top 10
- 07 파이썬의 탄생
- 08 이름의 유래
- 09 로고의 의미
- 10 The Zen of Python

# 감사합니다

2026년 중앙대학교 디지털 직무 아카데미

```
print("파이썬 학습을 시작하신 여러분을  
응원합니다! 🎉")  
print("질문이 있으시면 언제든 물어보세요.")  
print("Happy Coding! 🐍")
```

**Q&A 시간**

궁금한 점이 있으신가요?



11 아름다운 것이 추한 것보다 낫다

## 강의 목차

12 단순함이 복잡함보다 낫다

03 차를 살 때 놓아야 하는 것은?

04 왜 대화만을 해야 할까요?

05 코드를 짜는 걸 파일

06 인터프리터를 찾는 법 위 언어

05 언어별 트렌드 비교

06 프로그래밍 언어 Top 10

07 파이썬의 탄생

08 이름의 유래

09 로고의 의미

10 The Zen of Python

