PYTHON

Day I

INTRODUCE

CHOI WOO YOUNG

- Web Developer at Disceptio
- Solutions Architect, Instructor
- Skills & Languages
  - Python, Golang, Julia, Node.js, Google Tag Manager, ..
  - Web-dev, Data Science, Digital Marketing, ..

MiraeN

JOIN SLACK

https://join.slack.com/t/miraenjakdang1st/shared_invite/enQtMjU5NzM3MDUyMTY3LWM5OTkwZTU2Mzg4OTc5ZjBhMjk5MDRkMGQyMDAyOWY0NWNhMTc2M2UxNWI1OTQ3ZDg4OTdjYTVhMzZlZTM3NGY

MiraeN

REPOSITORY

https://github.com/ulgoon/jakdang-python

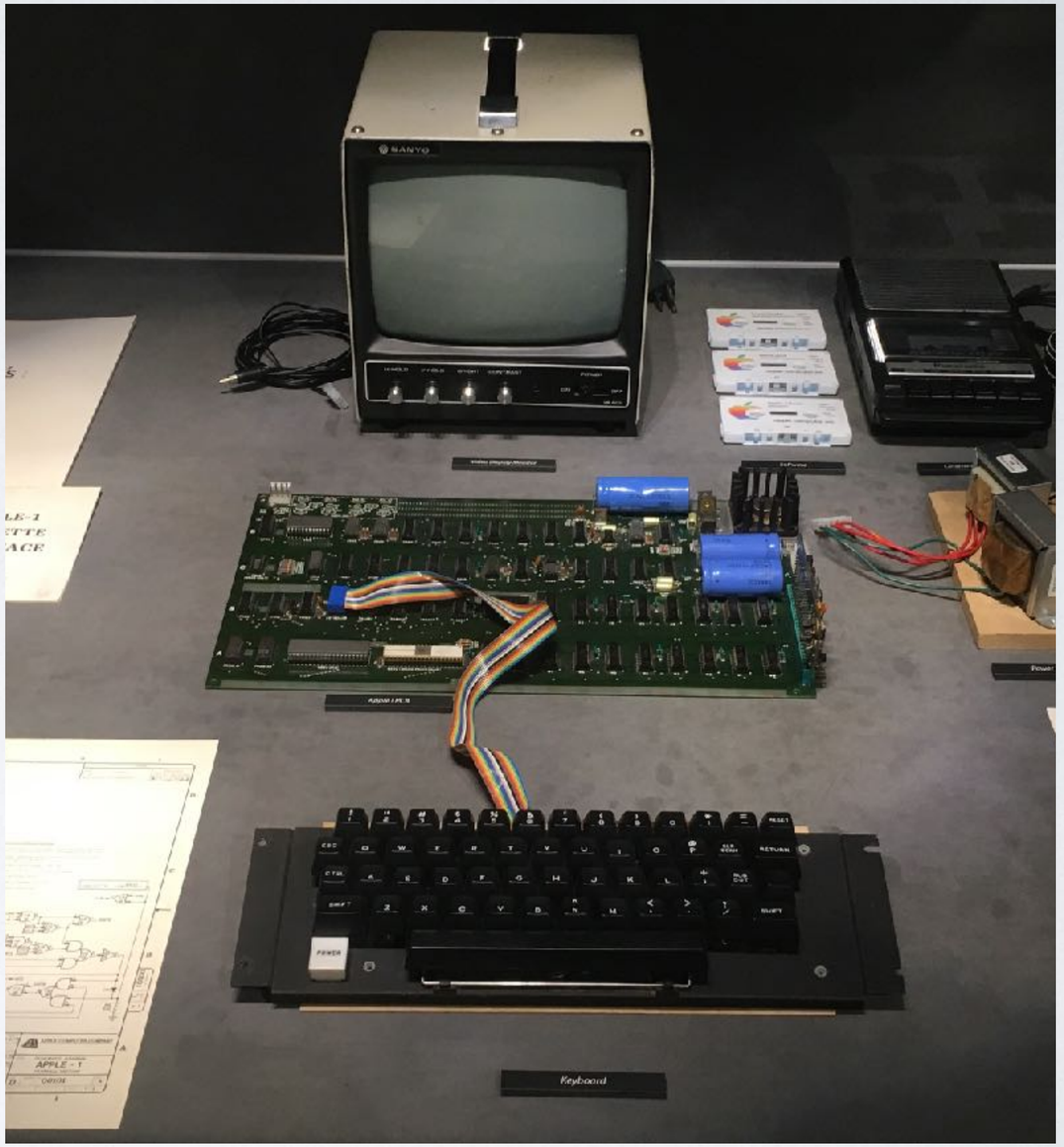
MiraeN

COMPUTER & PROGRAMMING

MiraeN

COMPUTER

Apple 1



CALCULATOR

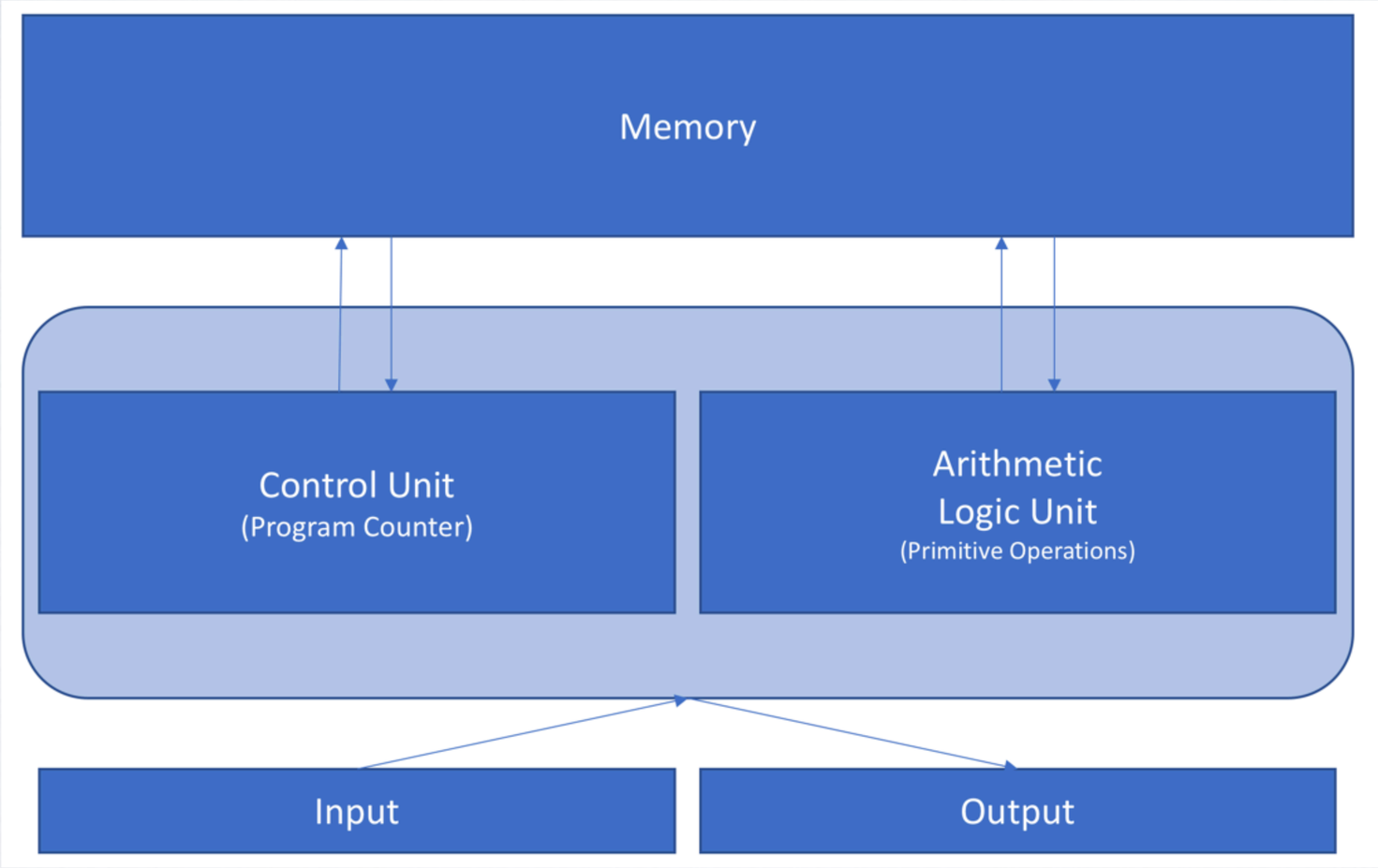
COMPUTER vs CALCULATOR

- "**calculation**" implies a strictly arithmetic process, whereas "**computation**" might involve applying rules in a systematic way
- `Stored Program` computer -> Computer
  - Stores and Executes intructions
- `Fixed Program` computer -> Calculator
  - just calculate

MiraeN

WHAT ABOUT THIS?

HOW COMPUTER WORKS?



COMPUTATIONAL THINKING

MiraeN

COMPUTATIONAL THINKING

정답이 정해지지 않은 문제에 대한 해답을 일반화하는 과정

PROCESS OF COMPUTATIONAL THINKING

1. 문제 조직화(추상화) - Problem Formulation (abstraction)
2. 솔루션 구현(자동화) - Solution Expression (automation)
3. 솔루션 실행 및 평가(분석) - Solution Execution & Evaluation (analyses)

MiraeN

CHARACTERISTICS OF COMPUTATIONAL THINKING

- 문제 분해(decomposition)
- 패턴인지 / 데이터 표현(pattern recognition / data representation)
- 일반화 / 추상화(generalization / abstraction)
- 알고리즘(algorithms)

MiraeN

PROCESS OF COMPUTATIONAL THINKING

문제 인지

- 배가 고프다!

MiraeN

문제 조직화

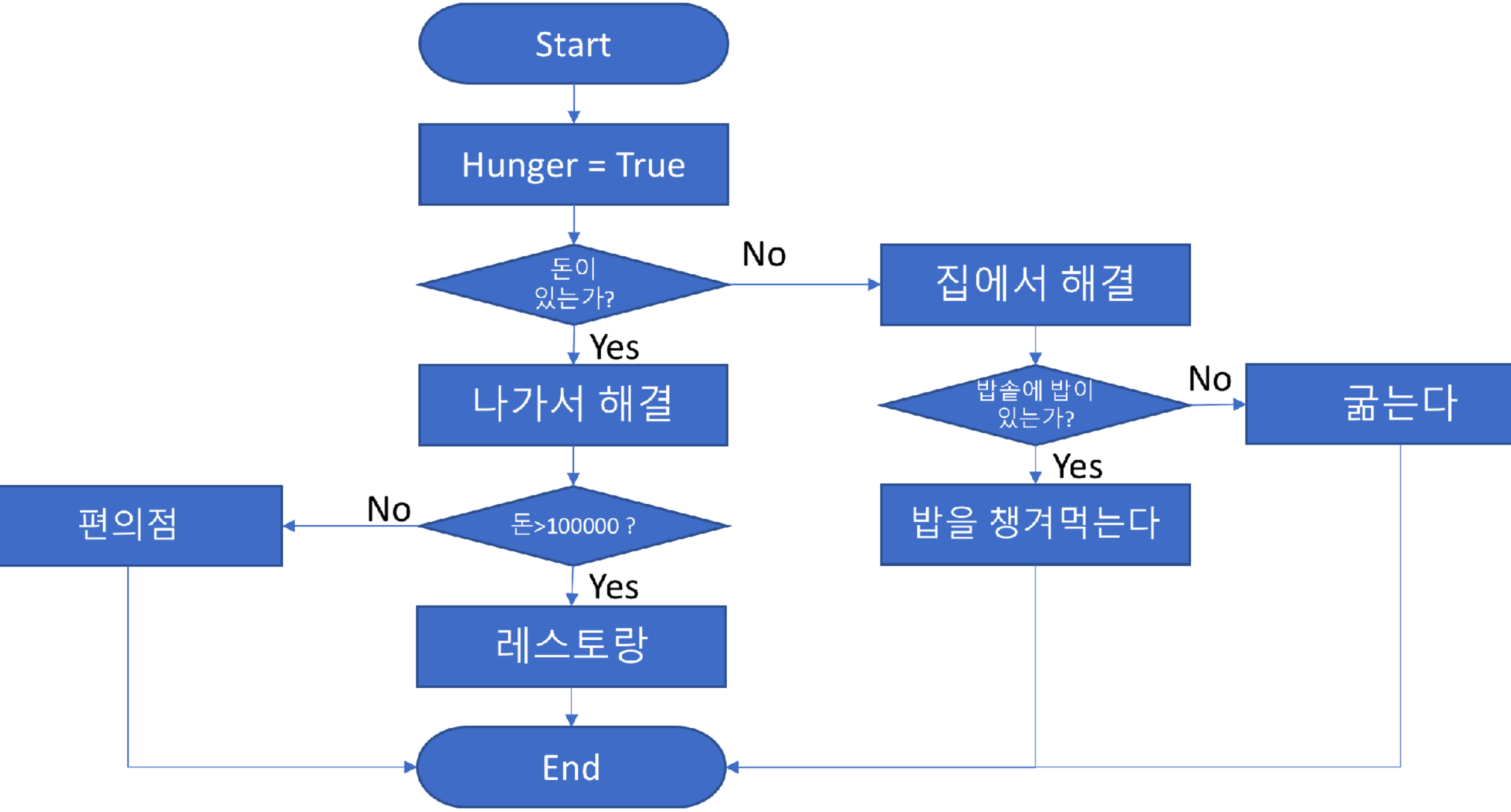
- 문제분해
  - 뭘 먹긴 해야겠다
    - 집에서 해결함
      - 냉장고엔 뭐가있지? 밥은 해놨나? 라면이라도 먹을까? ...
    - 나가서 해결함
      - 편의점? 식당? 패스트푸드? 레스토랑??

MiraeN

패턴 인지, 일반화/추상화, 알고리즘

- 패턴 인지
  - 아! 배가고프면 어디서 뭔가를 먹음으로써 Hunger가 False가 되는구나
- 일반화/추상화
  - 추상화(간결하고 명확하게 단순화, 일반화, 개념화)
    - 배가 고프면 `어디` 에서 `어떻게` 해결함
- 알고리즘

MiraeN


Start
Hunger = True
돈이
있는가?
No
Yes
집에서 해결
나가서 해결
밥솥에 밥이
있는가?
No
Yes
굶는다
밥을 챙겨먹는다
돈>100000 ?
No
Yes
편의점
레스토랑
End

솔루션 구현, 실행, 평가

- 실행 및 평가
  - 솔루션대로 실행해서 나는 배고픔을 인지하고 해결하게 되었다.
  - 돈 보유량에 따라 다양한 선택지를 추가 해야겠다.
  - 집에서 밥이 없으면 굶지 말고 밥을 해야겠다.

MiraeN

LET'S TRY

MiraeN

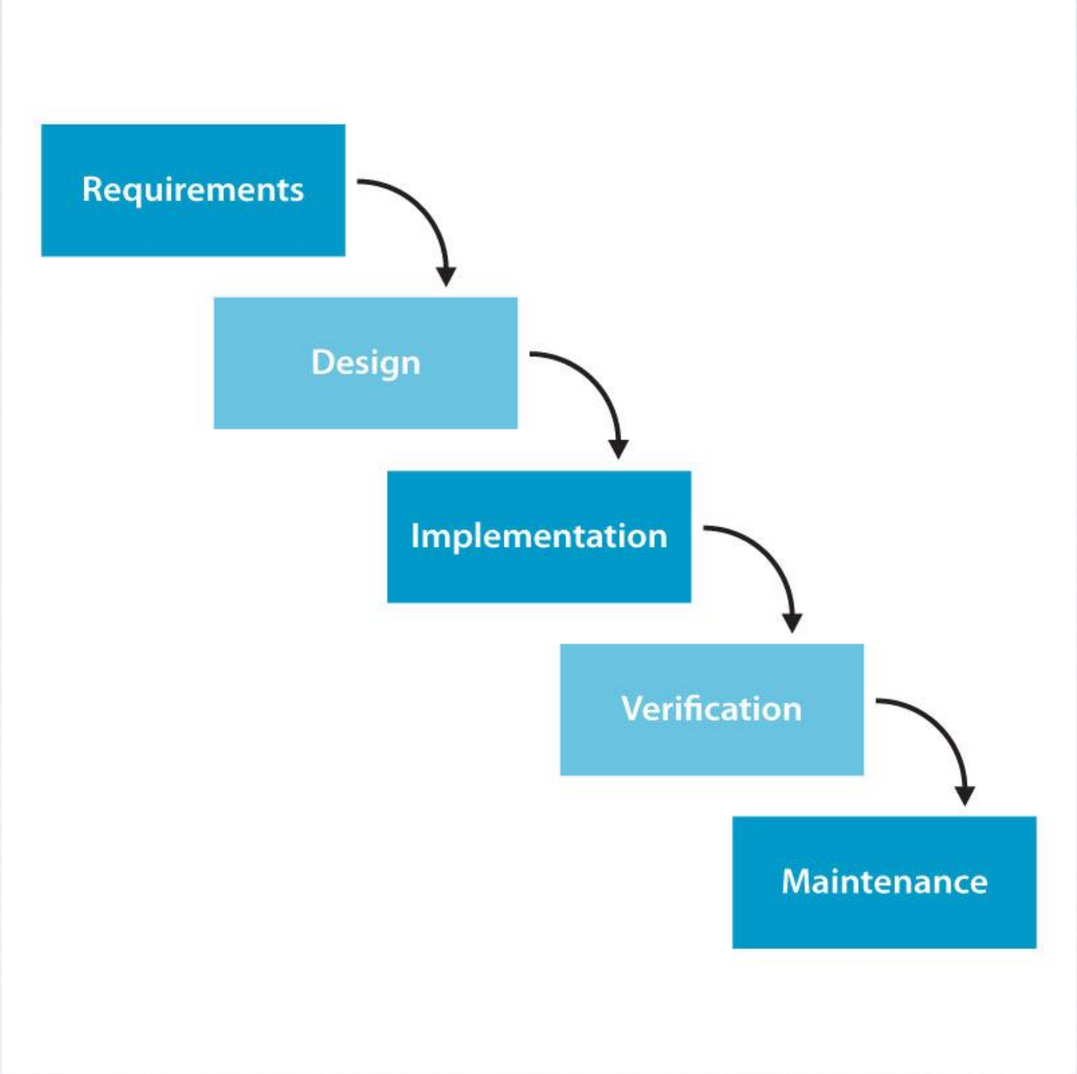
HOW DEVELOPERS WORK?

MiraeN

SOFTWARE DEV LIFECYCLE PROCESS

MiraeN

WATERFALL MODEL

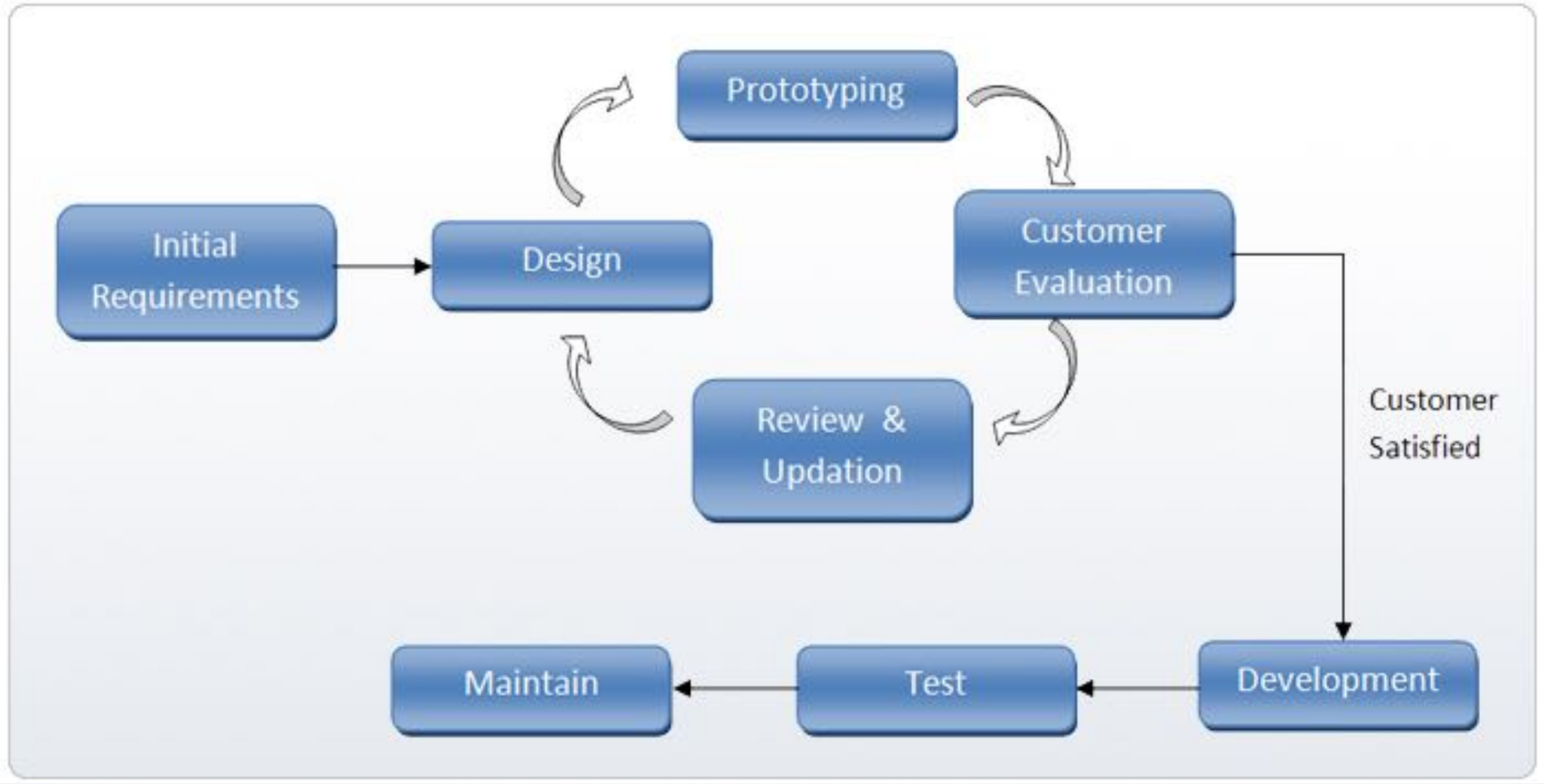


WATERFALL MODEL

- 순차적 개발모델
- 정형화된 접근 및 체계적인 문서화가 가능
- 직전단계가 완료되어야 다음 단계를 수행

MiraeN

PROTOTYPE MODEL



PROTOTYPE MODEL

- 고객의 요구사항을 적극적으로 반영
- 빠른 개발과 고객의 피드백을 빠르게 반영
- 대규모 프로젝트에 적용하기 힘듦

MiraeN

AGILE SOFTWARE DEVELOPMENT

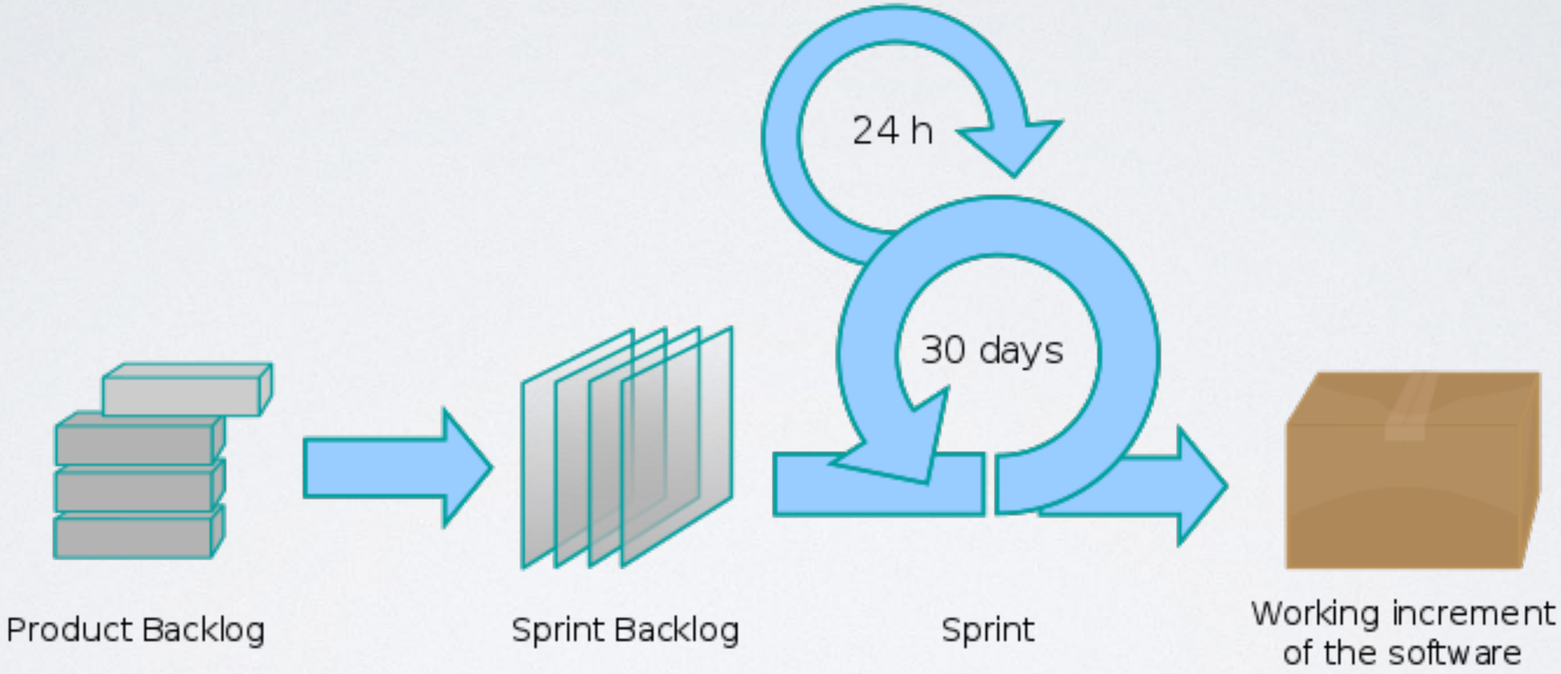
MiraeN

AGILE SCRUM

- 제품 개발과 유지를 위한 반복적인 애자일 소프트웨어 개발 프레임워크
- Product Owner: 고객의 의견을 대표하며, 비즈니스의 가치를 제시
- Dev Team: 스프린트의 종착점에 서비스가능한 제품을 만들어냄
- Scrum Master: 각 스프린트의 목표를 제시하며, Product Owner와 Dev Team의 의견충돌을 중재

MiraeN

SPRINT



PLANNING POKER

MiraeN

PLANNING POKER

- 애자일 일정 추정을 위해 사용
- 모든 팀원이 한 가지 과제에 대해 충분히 토론한 뒤, 작업시간을 추정
- deck
  - 0, 1/2, 1, 2, 3, 5, 8, 13, 20, 40, 100, ?, ∞, 커피
  - 단위 작업시간: 8시간

MiraeN

PLANNING POKER

- 플레이 방법
  1. 추정할 과제를 가장 잘 아는 사람이 해당 과제에 대해 설명
  2. 다른 사람은 추정에 필요한 정보를 얻기 위해 적극적인 질문과 토의
  3. 각자 생각하는 이 과제의 점수를 보이지 않게 내려놓음
  4. 점수를 공유하고 가장 낮은 점수, 가장 높은 점수를 낸 팀원이 이 점수를 낸 이유에 대해 설명
  5. 모든 팀원이 같은 점수를 낼 때 까지 3~4의 반복

MiraeN

LET'S PLAY

- 논현역 근처 카페에서 음료 사기
- 영화 예매부터 관람까지

MiraeN

PYTHON

MiraeN

PYTHON

- 1989년 크리스마스 연휴를 보내던 Guido Van Rossum 이 만든 고급 프로그래밍 언어
- 특징
  - 인터프리터
  - 객체지향
  - 동적 타이핑
  - 엄격한 문법

MiraeN

COMPILER, INTERPRETER

- compile: 기계가 이해 가능한 언어(기계어)로 프로그램을 해석하는 과정
- interprete: 프로그램의 소스코드를 바로 실행

MiraeN

절차지향, 객체지향

- 절차지향: 소스코드를 순차적으로 처리해 나감
- 객체지향: 데이터의 흐름을 먼저 개발한 뒤, 이를 호출해 사용

MiraeN

정적 타이핑, 동적 타이핑

- 정적 타이핑: 변수를 선언할 때, 그 변수의 자료형을 함께 선언

  int a;

- 동적 타이핑: 변수만 선언하고, 저장하는 자료형을 따라감

  a = 10

MiraeN

PYTHON의 엄격한 문법

- python은 계층을 이해하기 위해 들여쓰기를 사용

MiraeN

C, PYTHON

- int main() {
  int i;
  int sum=0;
  for (i=0;i<11;i++) {
  if (i%2==0){
  sum = sum +i;
  }
  }
  printf(“%d”, sum);
  }

- sum = 0
  for i in range(1,10+1):
  if i % 2 == 0:
  sum = sum + i
  print(sum)

MiraeN

INSTALL PYTHON

https://www.python.org/downloads/windows/

Python 3.6.1

Release Date: 2017-03-21

Python 3.6.1 is the first maintenance release of Python 3.6. The Python 3.6 series contains many new features and optimizations. See the What's New In Python 3.6 document for more information.

Major new features of the 3.6 series, compared to 3.5

Among the new major new features in Python 3.6 are:

- PEP 468, Preserving Keyword Argument Order
- PEP 487, Simpler customization of class creation
- PEP 495, Local Time Disambiguation
- PEP 498, Literal String Formatting
- PEP 506, Adding A Secrets Module To The Standard Library
- PEP 509, Add a private version to dict
- PEP 515, Underscores in Numeric Literals
- PEP 519, Adding a file system path protocol
- PEP 520, Preserving Class Attribute Definition Order
- PEP 523, Adding a frame evaluation API to CPython
- PEP 524, Make os.urandom() blocking on Linux (during system startup)

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		2d0fc9f3a5940707590e07f03ecb08b9	22540566	SIG
XZ compressed source tarball	Source release		692b4fc3a2ba0d54d1495d4ead5b0b5c	16872064	SIG
Mac OS X 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6dd08e7027d2a1b3a2c02cfacbe611ef	27511848	SIG
Windows help file	Windows		69082441d723060fb333dcda8815105e	7986690	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64, not Itanium processors	708496ebbe9a730d19d5d288afd216f1	6926999	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64, not Itanium processors	ad69fdacde90f2ce8286c279b11ca188	31392272	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64, not Itanium processors	a055a1a0e938e74c712a1c495261ae6c	1312520	SIG
Windows x86 embeddable zip file	Windows		8dff09a1b19b7a7dcb915765328484cf	6320763	SIG
Windows x86 executable installer	Windows		3773db079c173bd6d8a631896c72a88f	30453192	SIG
Windows x86 web-based installer	Windows		f58f019335f39e0b45a0ae68027888d7	1287064	SIG

Install Python 3.6.1 (64-bit)

Select Install Now to install Python with default settings, or choose
Customize to enable or disable features.

Install Now
C:₩Users₩choi₩AppData₩Local₩Programs₩Python₩Python36

Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☑ Add Python 3.6 to PATH

Cancel

Optional Features

- [x] Documentation

  Installs the Python documentation file.

- [x] pip

  Installs pip, which can download and install other Python packages.

- [x] tcl/tk and IDLE

  Installs tkinter and the IDLE development environment.

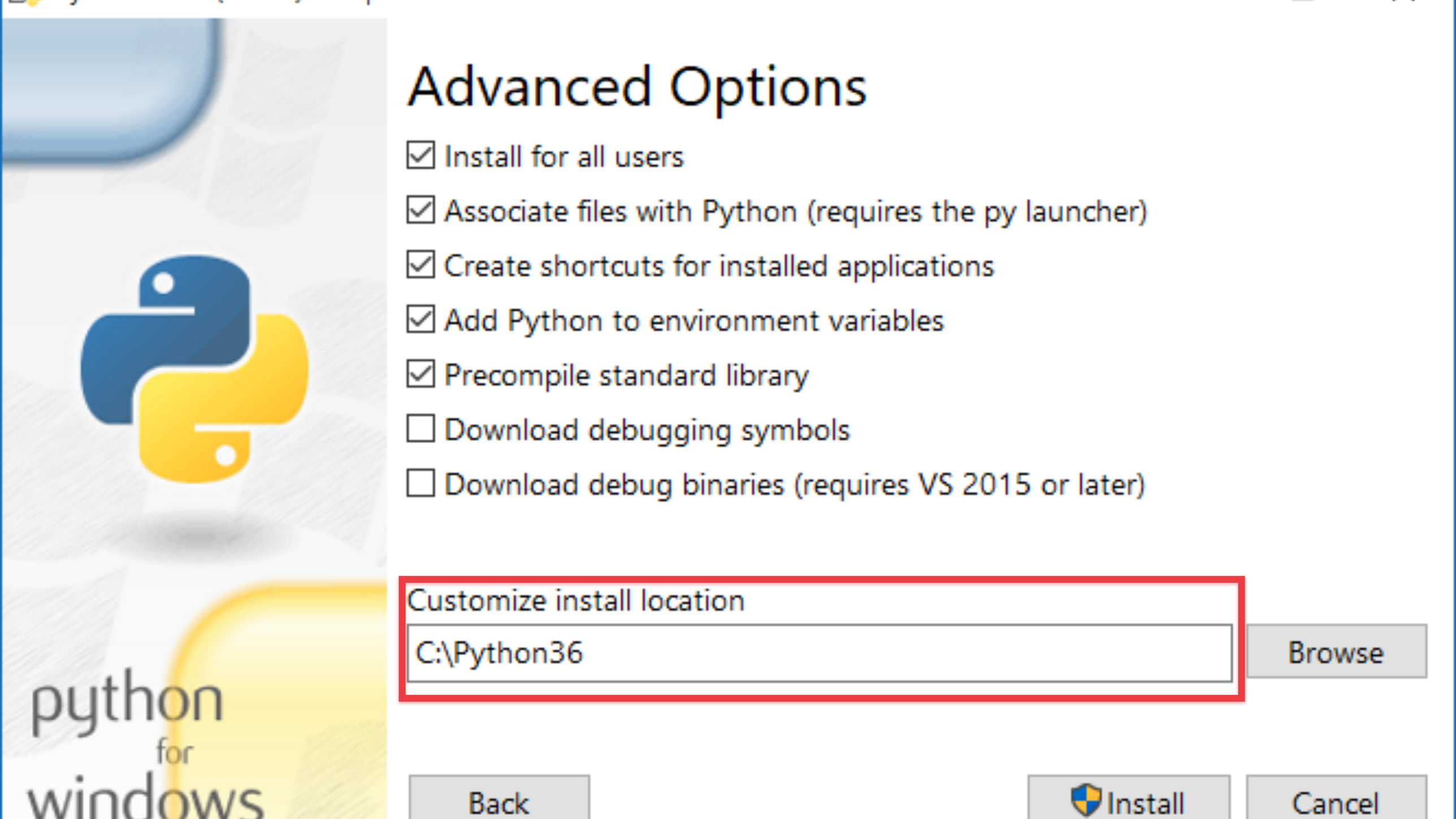
- [x] Python test suite

  Installs the standard library test suite.

- [x] py launcher [x] for all users (requires elevation)

  Installs the global 'py' launcher to make it easier to start Python.

Back Next Cancel


Advanced Options
Install for all users
Associate files with Python (requires the py launcher)
Create shortcuts for installed applications
Add Python to environment variables
Precompile standard library
Download debugging symbols
Download debug binaries (requires VS 2015 or later)
Customize install location
C:\Python36
Browse
python
for
windows
Back
Install
Cancel

Setup Progress

Installing:

Python 3.6.1 Core Interpreter (64-bit)

Cancel

Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

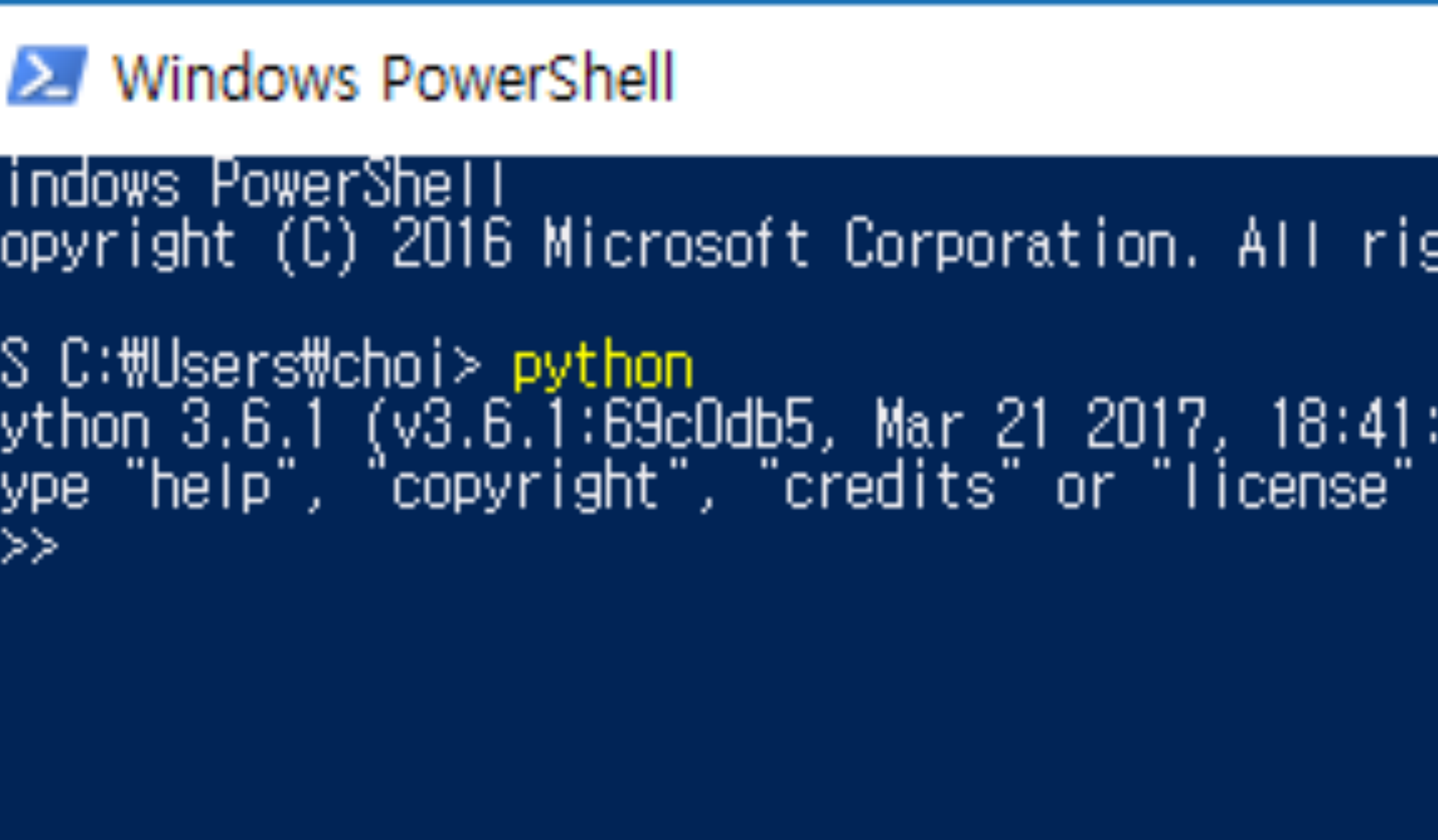
New to Python? Start with the online tutorial and documentation.

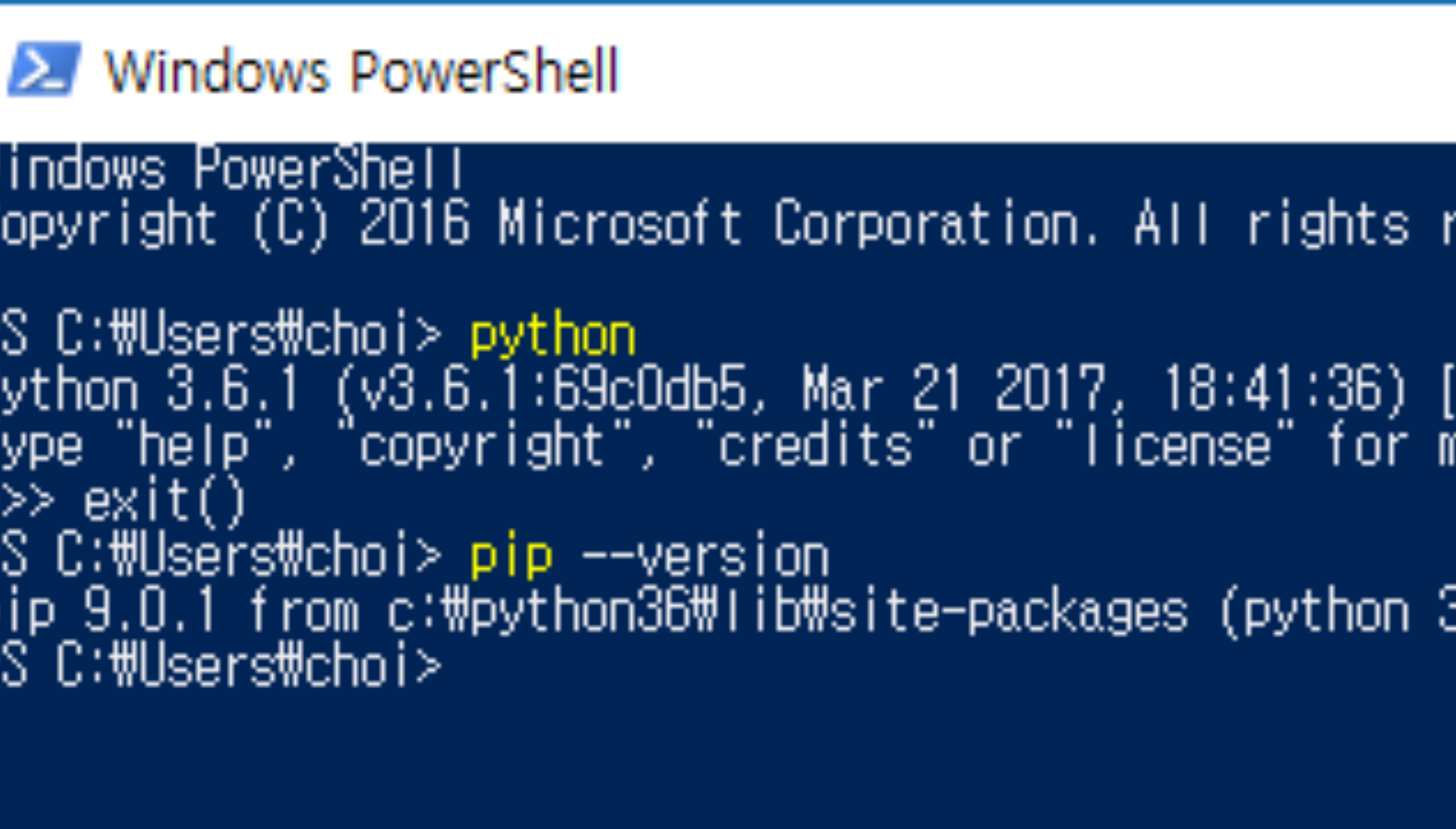
See what's new in this release.

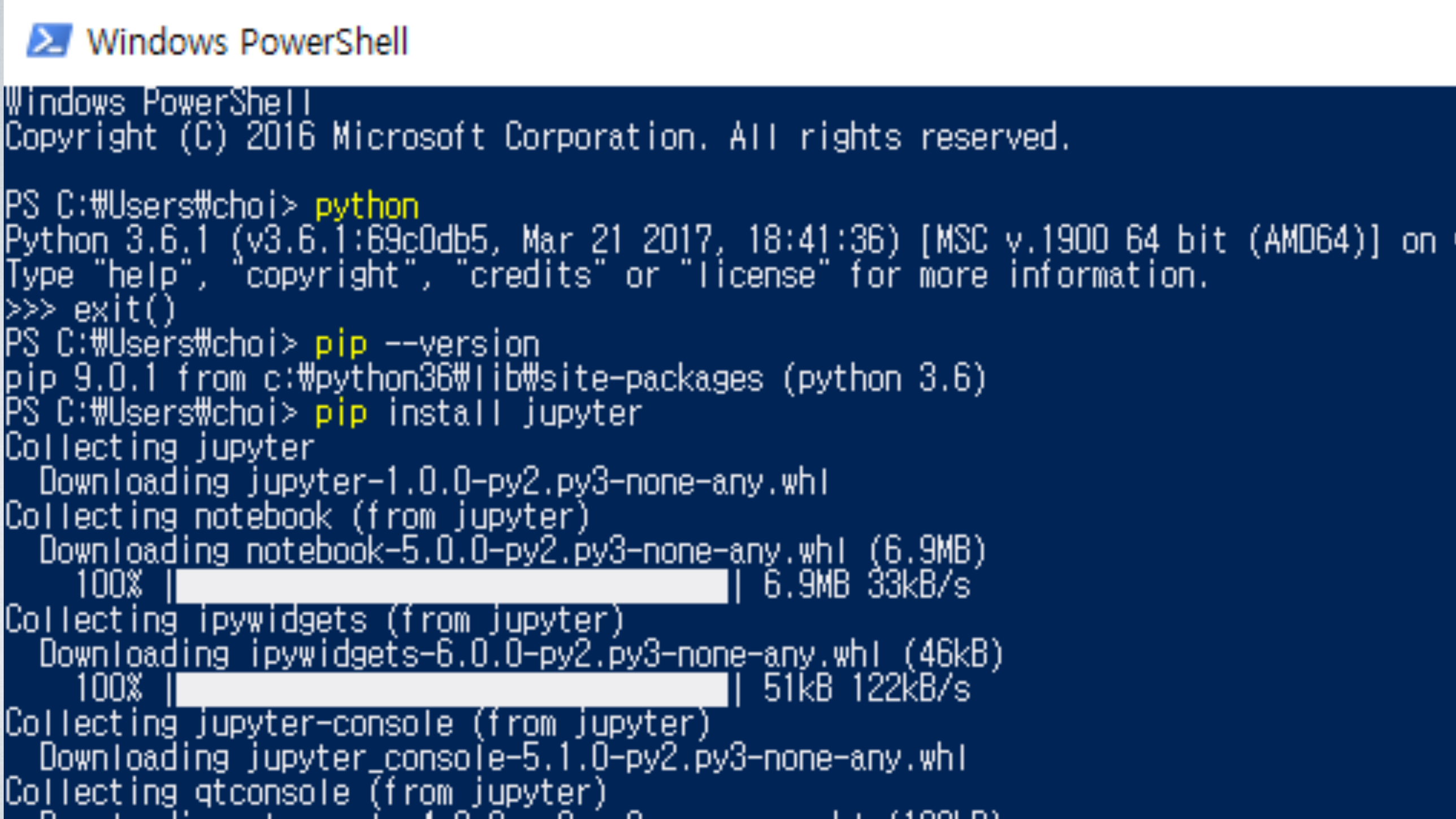
Disable path length limit

Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX_PATH" limitation.

Close


Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All righ
PS C:\Users\choi> python
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 18:41:3
Type "help", "copyright", "credits" or "license" f
>>>


Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights r
PS C:\Users\choi> python
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 18:41:36) [
Type "help", "copyright", "credits" or "license" for m
>>> exit()
PS C:\Users\choi> pip --version
pip 9.0.1 from c:\python36\lib\site-packages (python 3
PS C:\Users\choi>


Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.
PS C:¥Users¥choi> python
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 18:41:36) [MSC v.1900 64 bit (AMD64)] on
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()
PS C:¥Users¥choi> pip --version
pip 9.0.1 from c:¥python36¥lib¥site-packages (python 3.6)
PS C:¥Users¥choi> pip install jupyter
Collecting jupyter
Downloading jupyter-1.0.0-py2.py3-none-any.whl
Collecting notebook (from jupyter)
Downloading notebook-5.0.0-py2.py3-none-any.whl (6.9MB)
100% | 6.9MB 33kB/s
Collecting ipywidgets (from jupyter)
Downloading ipywidgets-6.0.0-py2.py3-none-any.whl (46kB)
100% | 51kB 122kB/s
Collecting jupyter-console (from jupyter)
Downloading jupyter_console-5.1.0-py2.py3-none-any.whl
Collecting qtconsole (from jupyter)

Windows PowerShell

```
PS C:₩Users₩choi₩Documents₩python> jupyter notebook
[I 18:56:22.871 NotebookApp] Writing notebook server cookie
book_cookie_secret
[I 18:56:24.113 NotebookApp] Serving notebooks from local di
[I 18:56:24.113 NotebookApp] 0 active kernels
[I 18:56:24.113 NotebookApp] The Jupyter Notebook is running
5f0dab2b195efd39ad39c7
[I 18:56:24.114 NotebookApp] Use Control-C to stop this serv
[C 18:56:24.124 NotebookApp]

    Copy/paste this URL into your browser when you connect f
    to login with a token:
        http://localhost:8888/?token=ae21b84bef7e38d3161194b
[I 18:56:40.563 NotebookApp] Accepting one-time-token-authen
```