



Introduction to Python Programming

Learn fundamental programming
concepts in a beginner friendly
language

Set up

- Go to <https://github.com/ariannedee/intro-to-python> and follow the installation instructions in the Readme
 - Install Python 3.6+
 - Install an IDE (I'll be using PyCharm)
 - Download the code
- Download the PDF of these slides
- Download the resources PDF



Introduction

Today's schedule

- **Introduction and set-up** (20 mins)
- **First code** (30 mins)
 - Break, Q&A
- **Learn programming basics** (80 mins)
 - Break, Q&A
- **Control the flow with conditionals** (40 mins)
 - Break, Q&A
- **Work with lists and loops** (40 mins)
- **What to learn next** (5 mins)

Break format

- 3 Breaks (10 mins)
 - Step away or work through code
- Q&A (5 mins)
 - Use Q&A feature
- Use group chat throughout for questions that anyone can answer

Poll

How much programming do you already know?

- Absolutely none
- A little bit
- A moderate amount
- A lot

Poll (multi-choice)

- What are your eventual goals with learning Python
 - Career change
 - Better understanding and communication
 - Use it in my current career (as a non-developer)
 - Use it in my current career (as a developer)
 - For fun
 - Other



Introduction

Installation

Set up

GitHub repository

- Go to <https://github.com/ariannedee/intro-to-python>
- Download project code
- Follow the installation instructions

Resources widget

- Download the PDF of these slides
- Download the PyCharm Reference PDF

Install links

- Download the code
 - <https://github.com/ariannedee/intro-to-python>
- Install Python 3.6+ for your operating system
 - <https://www.python.org/downloads/>
- Download the free, community edition of PyCharm
 - <https://www.jetbrains.com/pycharm/download/>

[Code](#)[Issues 0](#)[Pull requests 0](#)[Projects 0](#)[Wiki](#)[Insights](#)[Settings](#)

Code for the Safari Live Training - Introduction to Python Programming

[Edit](#)[Manage topics](#)[1 commit](#)[1 branch](#)[0 releases](#)[1 contributor](#)

Branch: master ▾

[New pull request](#)[Create new file](#)[Upload](#)**Click me**[Clone or download ▾](#) ariannedee Initial commit

Latest commit a79c66a 16 minutes ago

 .gitignore

Initial commit

16 minutes ago

 README.md

Initial commit

16 minutes ago

 README.md

www.python.org/downloads

python™

Search GO Socialize

About Downloads Documentation Community Success Stories News Events

Download the latest version for Mac OS X

Download Python 3.7.1

Click me

Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [Mac OS X](#), [Other](#)

Want to help test development versions of Python? [Pre-releases](#)

Looking for Python 2.7? See below for specific releases

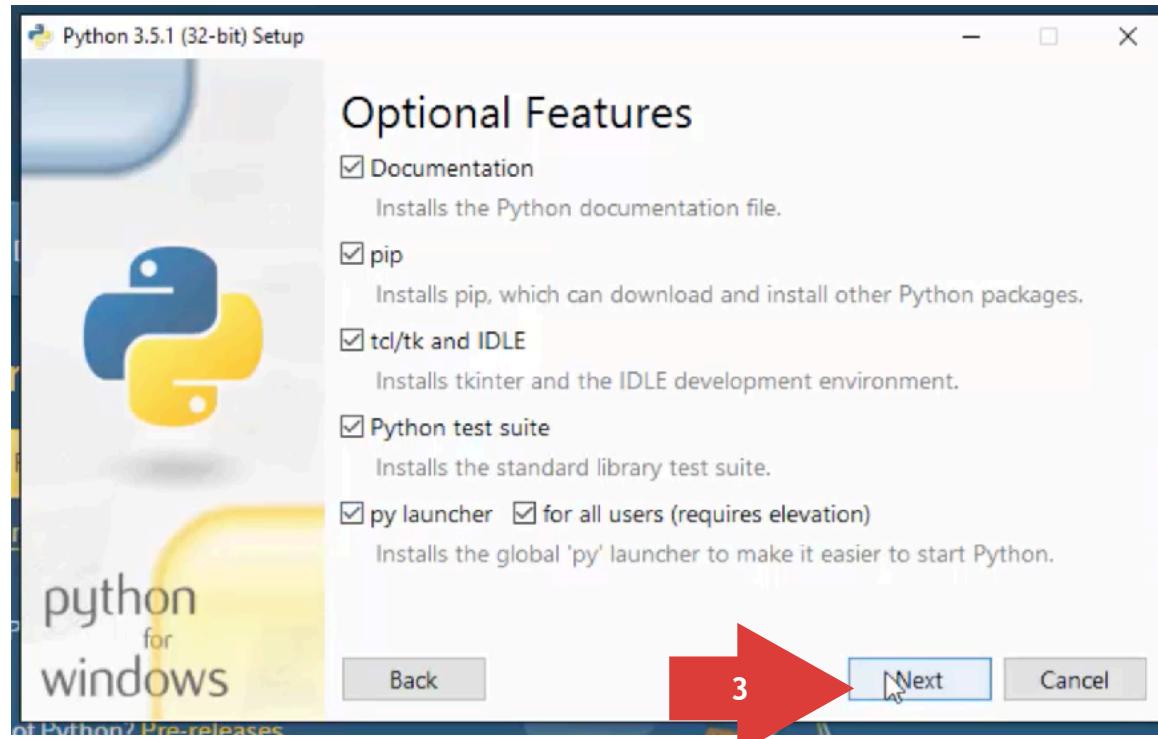


A screenshot of the Python downloads page. At the top left is the Python logo. To its right is a search bar with a magnifying glass icon and a 'GO' button. Below the search bar are links for 'Socialize'. A navigation bar with tabs 'About', 'Downloads', 'Documentation', 'Community', 'Success Stories', 'News', and 'Events' follows. The main content area features a large yellow button with the text 'Download the latest version for Mac OS X' and a yellow button below it with 'Download Python 3.7.1'. A red arrow points to the 'Download Python 3.7.1' button with the text 'Click me' written over it. Below these buttons is a paragraph of text about other operating systems. Further down are links for pre-releases and Python 2.7. On the right side of the content area is a cartoon illustration of two cardboard boxes hanging from yellow and white striped parachutes against a blue background with white clouds.

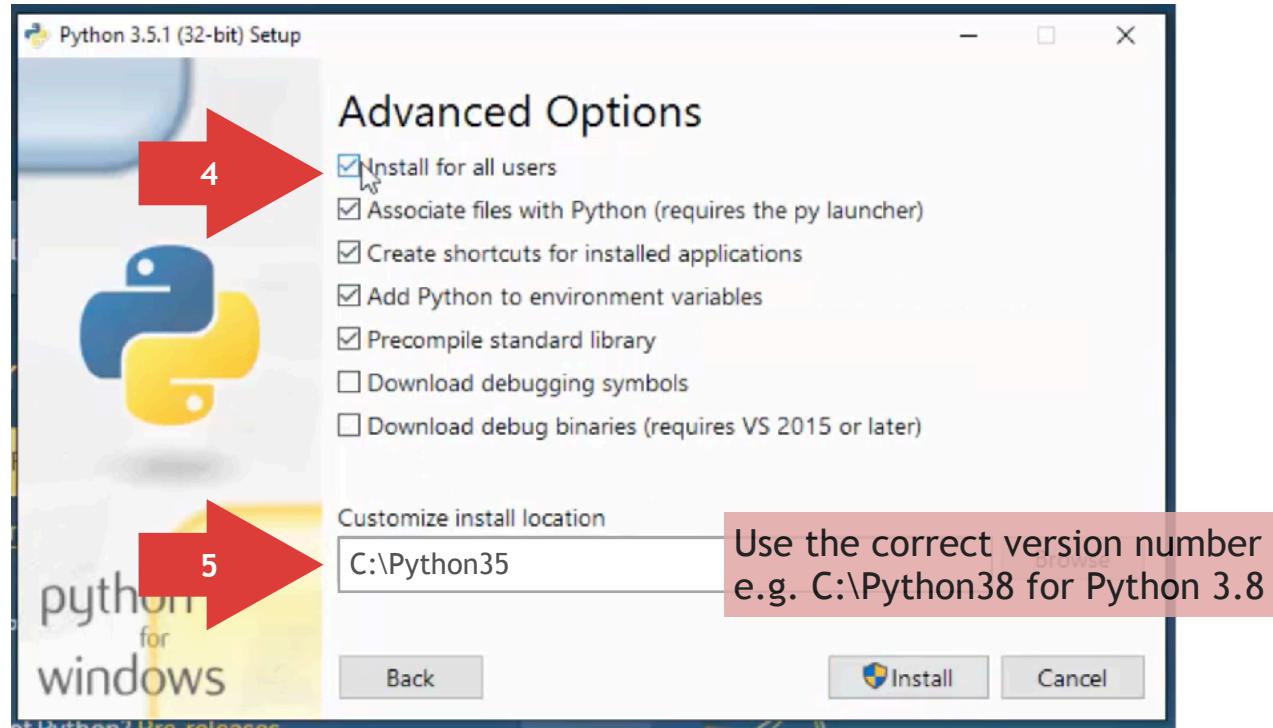
On Windows



On Windows



On Windows

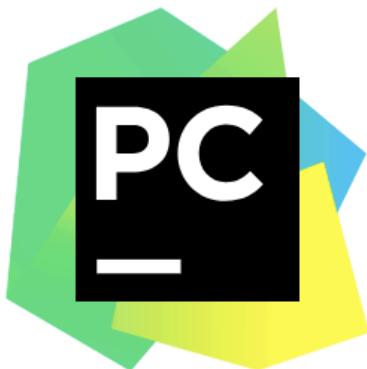


If you already installed Python

Follow the instructions to add Python to your PATH

[Link](#)

www.jetbrains.com/pycharm/download/



Download PyCharm

[Windows](#)

[macOS](#)

[Linux](#)

Professional

Version: 2018.3.2
Build: 183.4886.43
Released: December 18, 2018

Full-featured IDE
for Python & Web
development

[System requirements](#)
[Installation Instructions](#)
[Previous versions](#)

[DOWNLOAD](#)

Free trial

Community

Lightweight IDE
for Python & Scientific
development

[DOWNLOAD](#)

Free, open-source

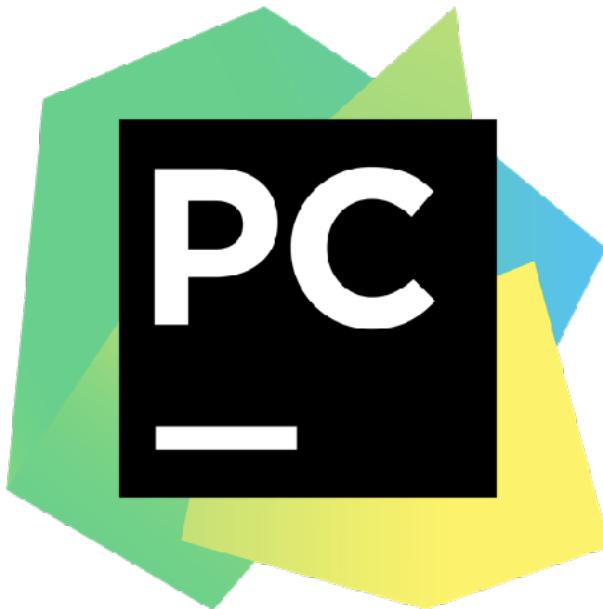
Click me

Checking python versions

Open Command Line (PC) or Terminal (Linux, Mac)

- Type `python --version`
or
- Type `python3 --version`
- One of those commands should return:
 - Python 3.8.x

Integrated Development Environment (IDE)



Why we're using PyCharm

- Handles **Python** out of the box
 - Syntax highlighting
 - Error highlighting
 - Autocomplete
- Better for **beginners** who don't know the command line
- Full-featured for **professional** Python developers

Alternatives to PyCharm

- Anaconda
 - Data scientists
- Thonny
 - Absolute beginners
- Sublime text, VS code, Atom
 - Multi-lingual programmers
 - Requires plug-ins to fully support Python
- Notepad, Notepad++, Vim, Emacs
 - Old-school programmers
- <https://realpython.com/python-ides-code-editors-guide/>



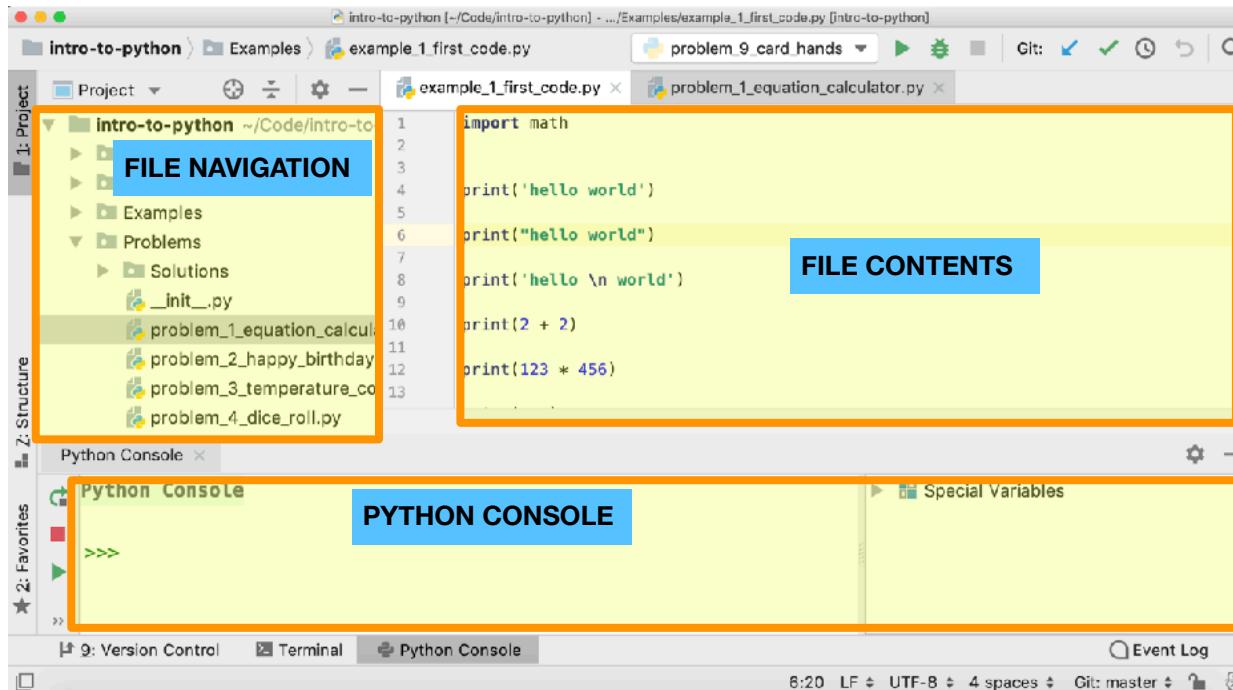
First code

Hello world!

Open project files

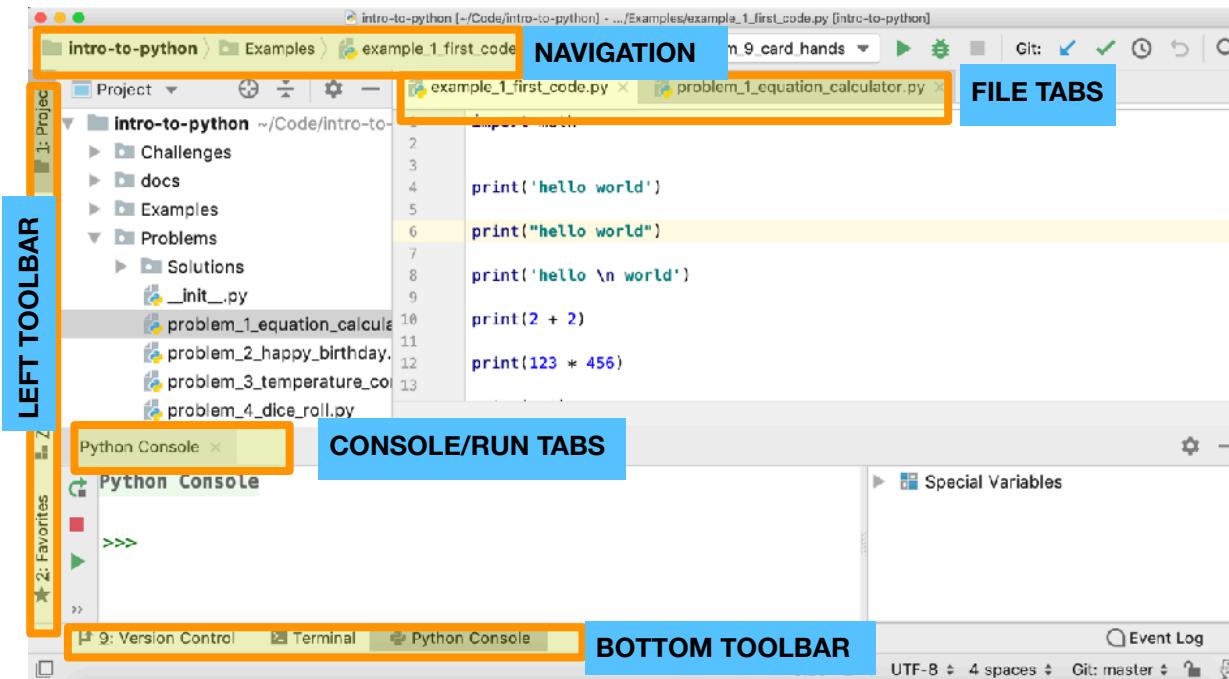


PyCharm Layout



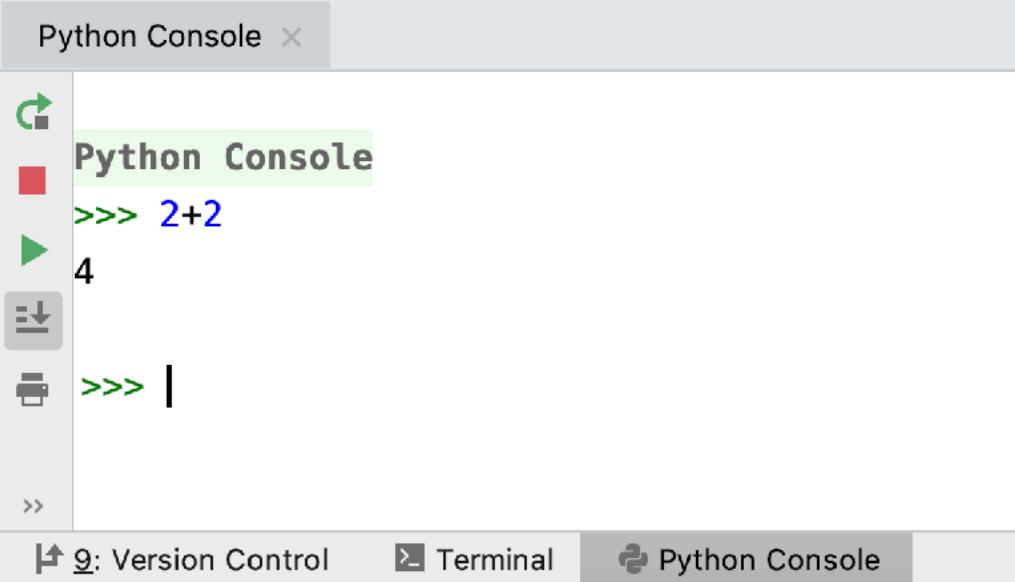
Reference: page 2

PyCharm Toolbars



Reference: page 2

Run code in the console



The screenshot shows a Python Console window with the following content:

- Header: "Python Console" with a close button.
- Left sidebar icons:
 - Green arrow icon (top)
 - Red square icon
 - Green triangle icon (top)
 - Gray downward arrow icon
 - Printer icon
 - Double greater than icon (bottom)
- Main area:
 - Text: "Python Console"
 - Text: ">>> 2+2"
 - Text: "4"
 - Text: ">>> |"
- Bottom navigation bar:
 - "9: Version Control" icon
 - "Terminal" icon
 - "Python Console" icon (highlighted in gray)

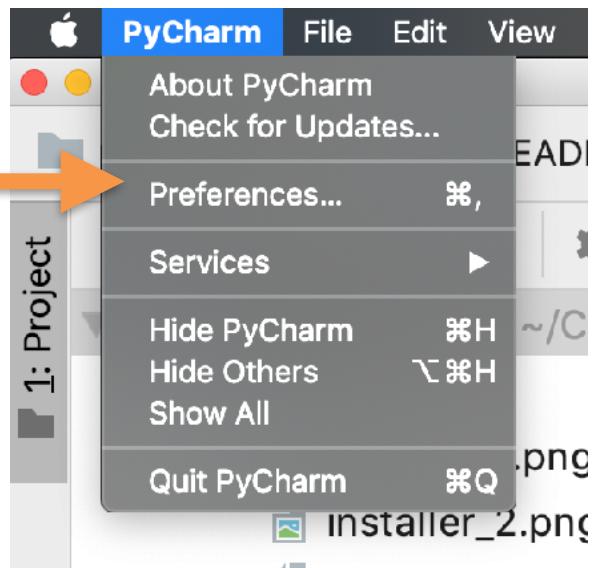
If PyCharm doesn't recognize Python3

Reference: page 4-6

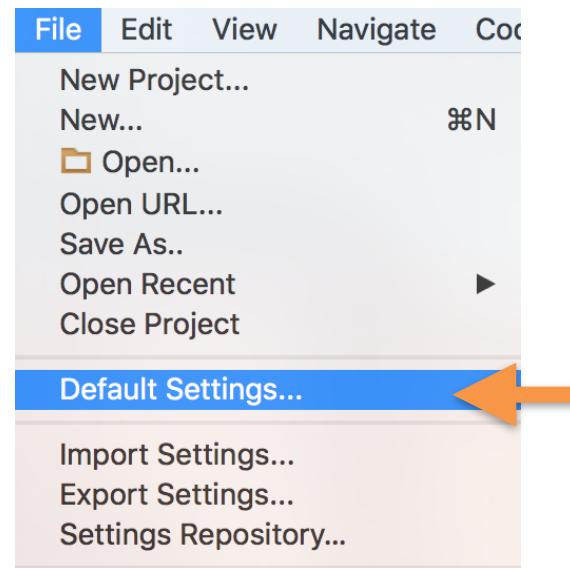
If PyCharm doesn't recognize Python3

1. Open settings/preferences

Mac

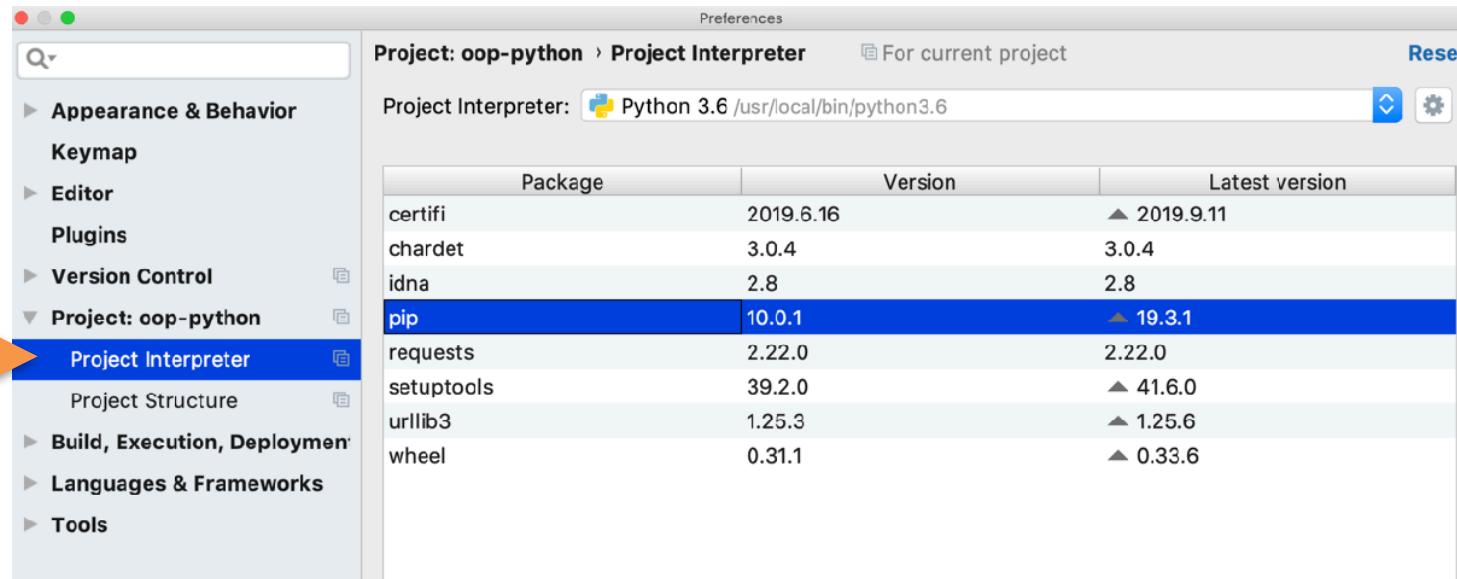


PC



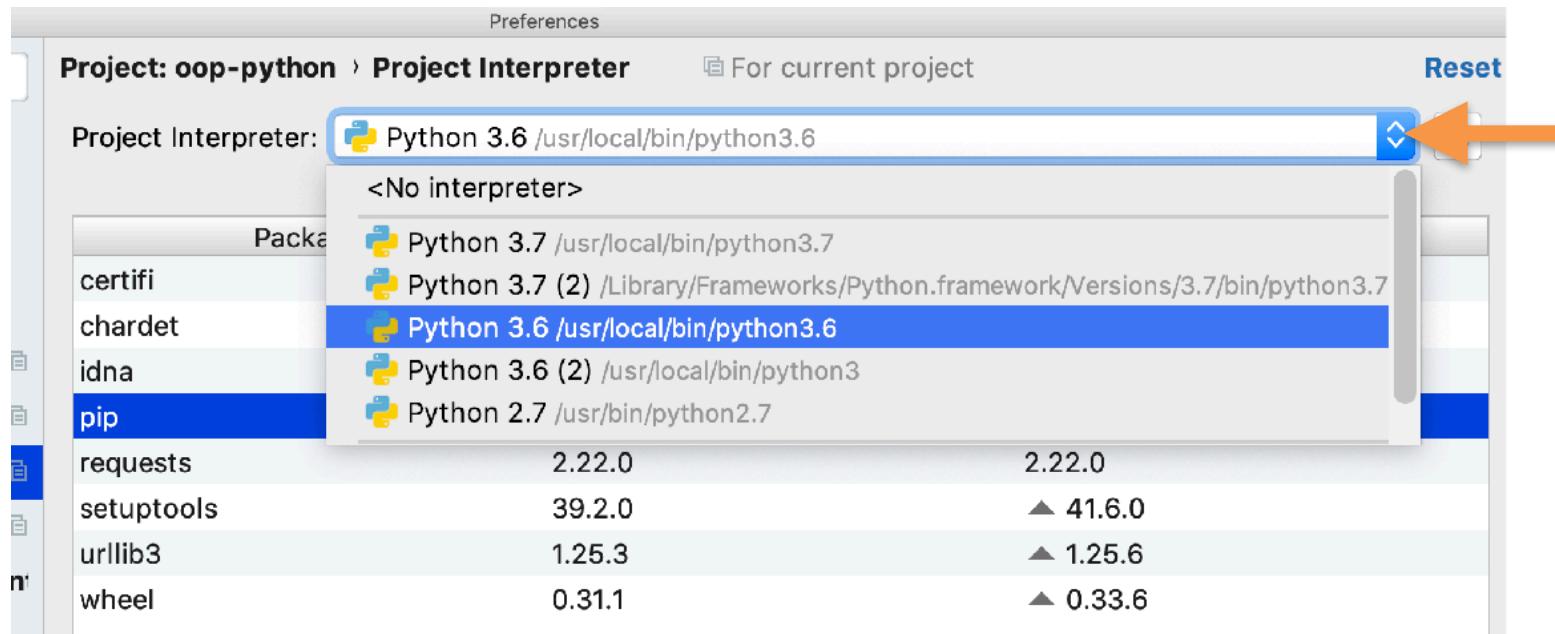
If PyCharm doesn't recognize Python3

2. Navigate to Project Interpreter



If PyCharm doesn't recognize Python3

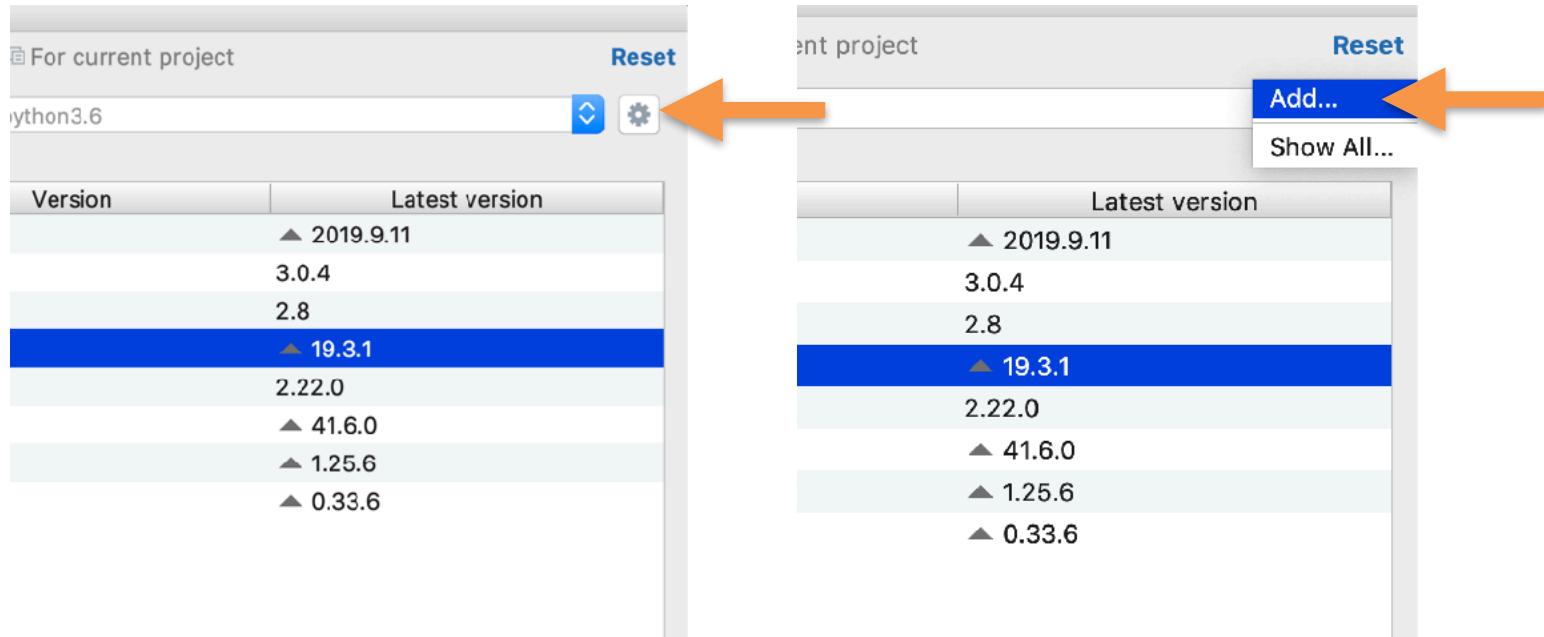
3a. Select existing interpreter



Done!

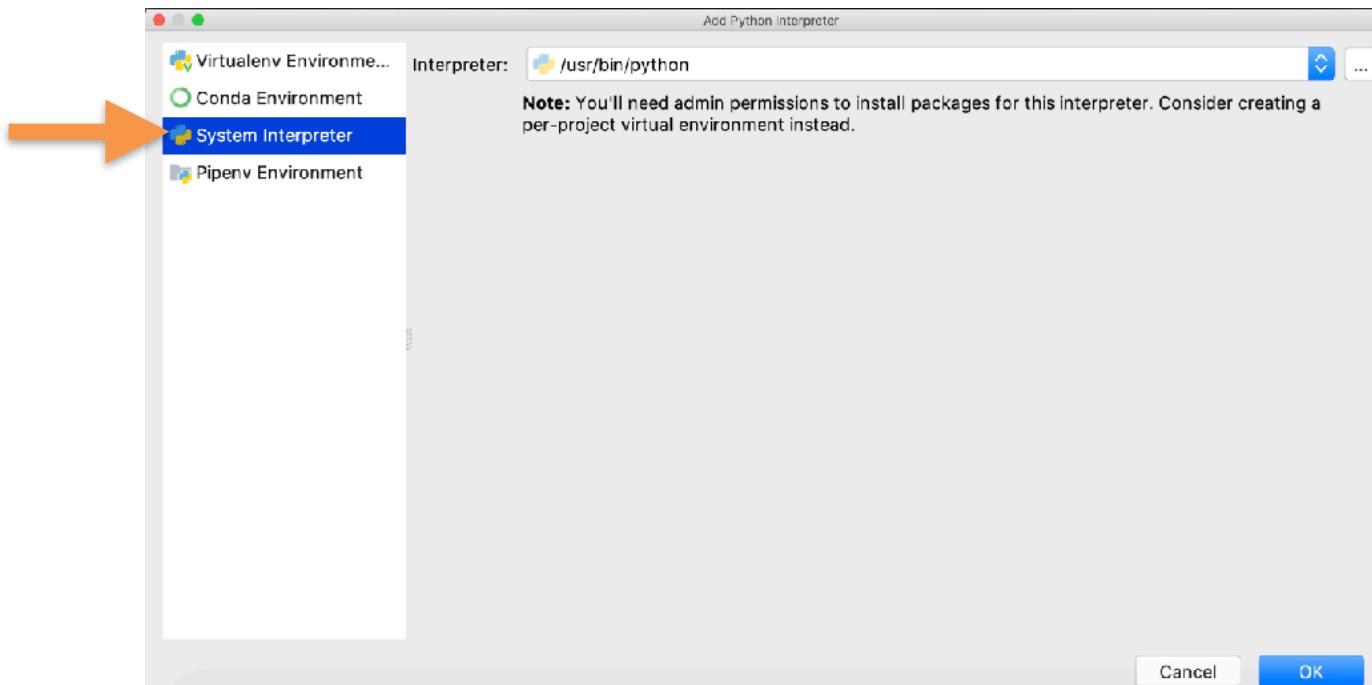
If PyCharm doesn't recognize Python3

3b. Add new interpreter



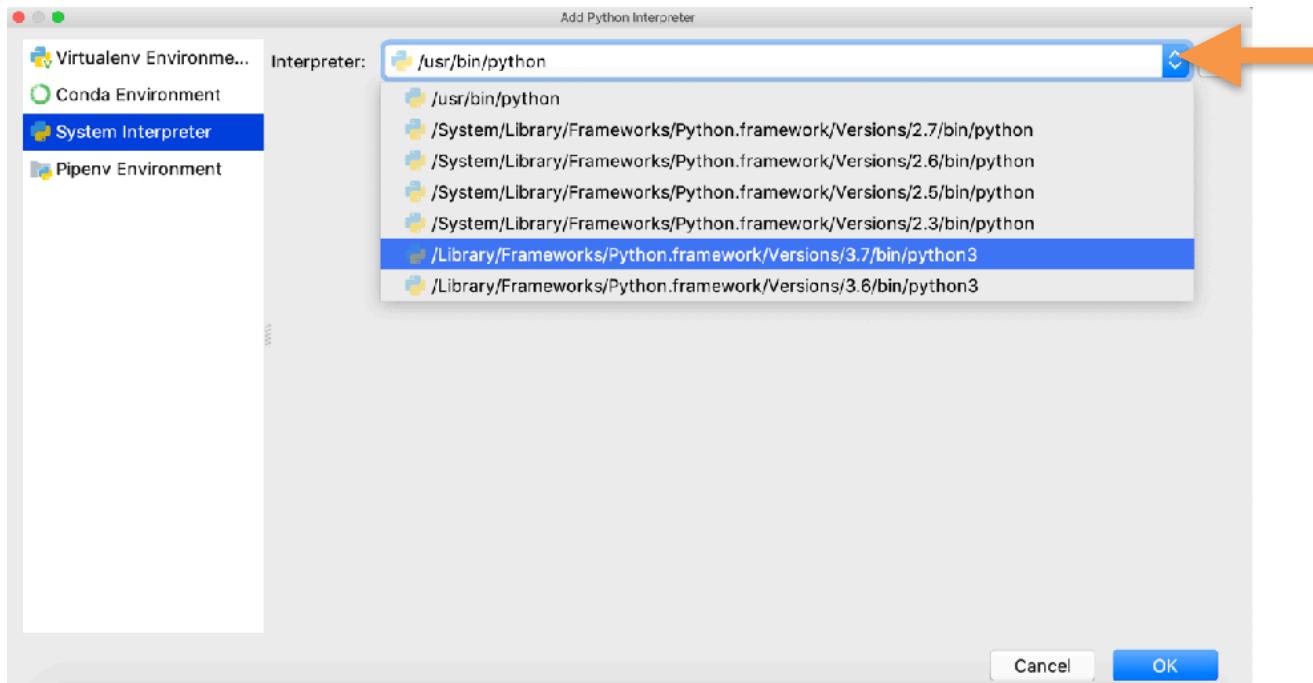
If PyCharm doesn't recognize Python3

4. Add a system interpreter



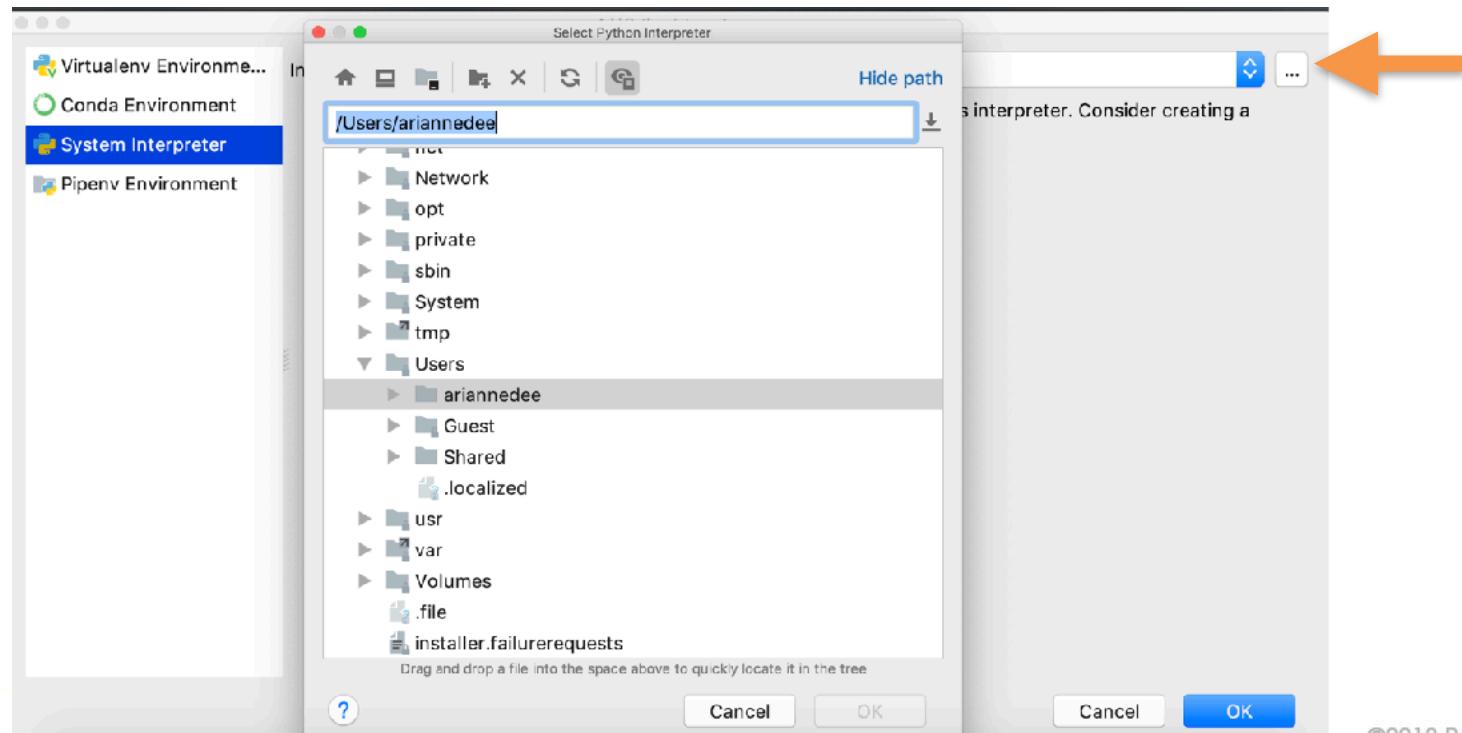
If PyCharm doesn't recognize Python3

5a. Select existing system interpreter



If PyCharm doesn't recognize Python3

5b. Find system interpreter

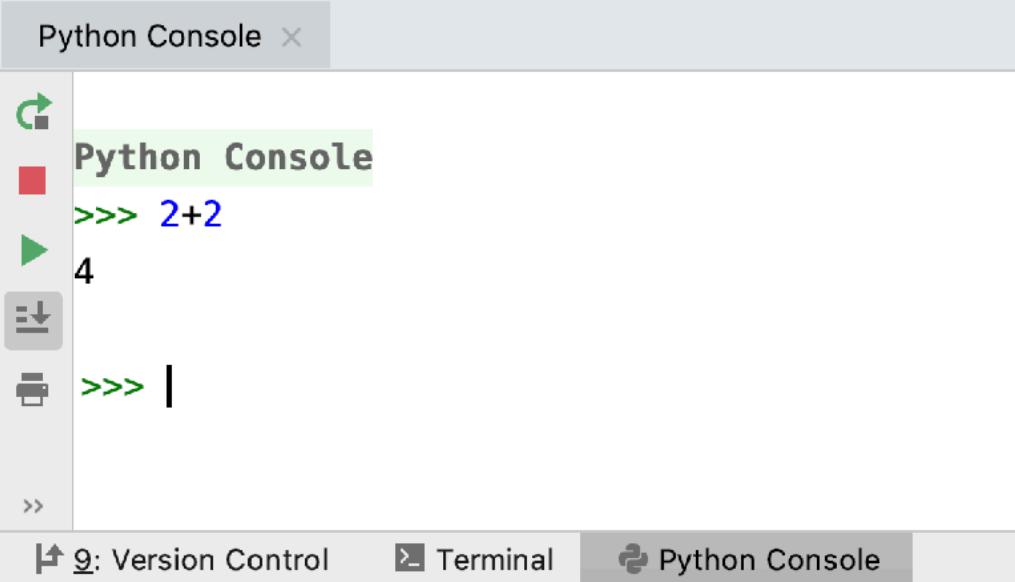


Potential system interpreter locations

- **Windows** (looking for `python.exe`)
 - `C:\Python38`
 - `C:\Program Files\Python38`
 - `C:\Users\username\AppData\Local\Programs\Python\Python38-XX`
- **Mac** (looking for `python3.8`)
 - `/usr/local/bin/`
 - `/Library/Frameworks/Python.framework/Versions/3.8/bin/`
 - `/usr/local/Cellar/python/3.8.X_X/bin/`
 - `/Users/username/anaconda/bin/`
- **Linux** (looking for `python3.8`)
 - `/usr/bin/`

Reference: page 6

Run code in the console



The screenshot shows a Python Console window with the following content:

- Header: "Python Console" with a close button.
- Left sidebar icons:
 - Green arrow icon (top)
 - Red square icon
 - Green triangle icon (top)
 - Gray downward arrow icon
 - Printer icon
 - Double greater than icon (bottom)
- Main area:
 - Text: "Python Console"
 - Text: ">>> 2+2"
 - Text: "4"
 - Text: ">>> |"
- Bottom navigation bar:
 - "9: Version Control" icon
 - "Terminal" icon
 - "Python Console" icon (highlighted in gray)

Run code from a file

Right click in file

The screenshot shows a code editor window with a context menu open over a specific line of code. The code in the editor is:

```
1 import math
2
3
4 print('hello world')
5
6 print("hello world")
7
8 print('hello \n world')
9
10 print(2 + 2)
11
12 print(123 * 456)
13
14 print(1/2)
15
```

The line `print('hello \n world')` is highlighted with a yellow background. A context menu is open to the right of the code, listing several options: Column Selection Mode, Refactor, Folding, Go To, Generate..., Run 'example_1_first_code' (highlighted with a blue selection bar and an orange arrow pointing to it), Debug 'example_1_first_code', Create 'example_1_first_code'..., Reveal in Finder, Open in Terminal, Local History.

Comments

```
1      #####
2      Calculate the gravitational force between Earth and Venus
3      #####
4
5      G = 6.67e-11 # Gravitational constant
6
```

Problem #1

Gravitational force calculator

$$F_g = \frac{Gm_1m_2}{r^2}$$

Other ways to run Python code

IDLE

Terminal / Command Prompt



First code

About Python

“Hello World” in different languages

Roughly from high - low level of abstraction

“Hello World” in different languages

Python

```
print("Hello World")
```

“Hello World” in different languages

JavaScript

```
console.log("Hello World!");
```

“Hello World” in different languages

C#

```
using System;

class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Hello, world!");
    }
}
```

“Hello World” in different languages

Java

```
class HelloWorldApp {  
    public static void main(String[] args) {  
        System.out.println("Hello World!"); // Prints the string to the  
        console.  
    }  
}
```

“Hello World” in different languages

C++

```
#include <iostream>

int main()
{
    std::cout << "Hello, world!\n";
    return 0;
}
```

“Hello World” in different languages

Assembly

```
global _main
extern _printf

section .text
_main:
    push    message
    call    _printf
    add     esp, 4
    ret
message:
    db  'Hello, World', 10, 0
```

“Hello World” in different languages

Machine Code

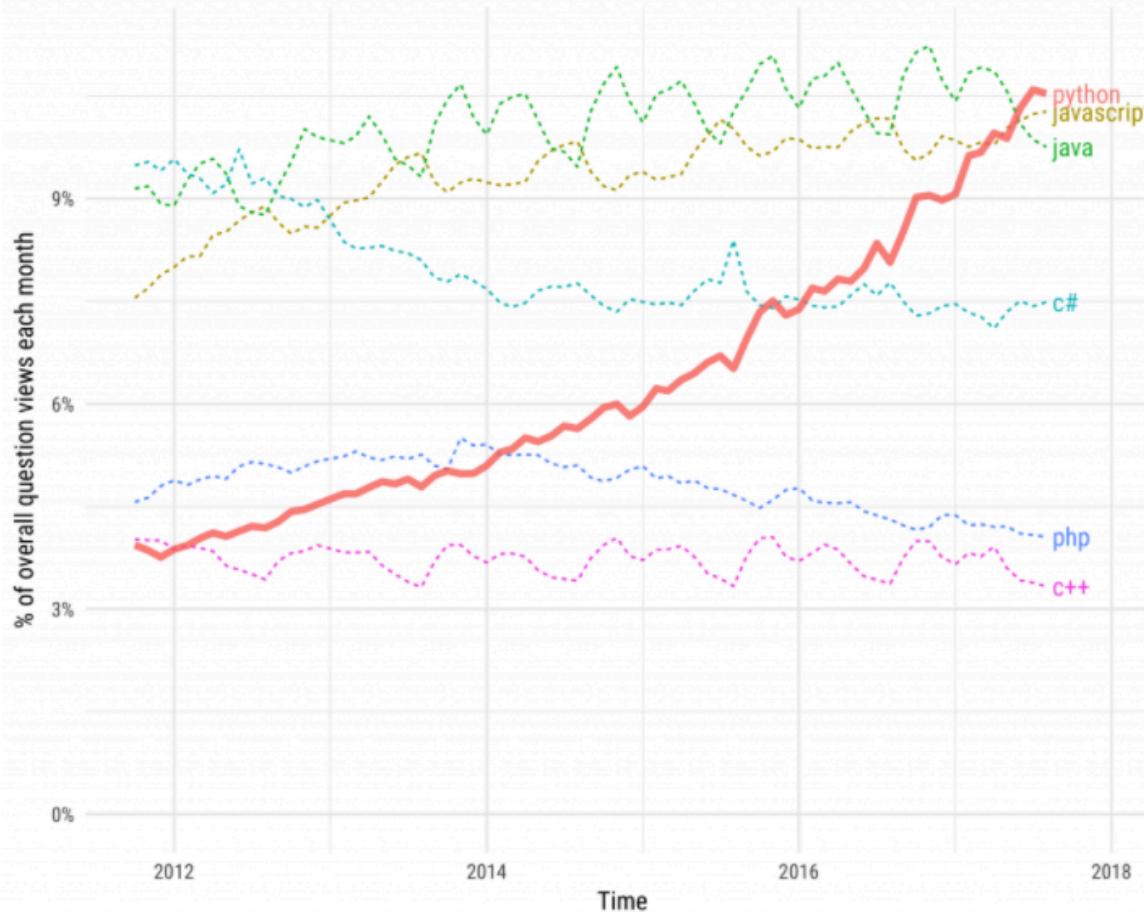
```
b8    21 0a 00 00  #moving "!\n" into eax
a3    0c 10 00 06  #moving eax into first memory location
b8    6f 72 6c 64  #moving "orld" into eax
a3    08 10 00 06  #moving eax into next memory location
b8    6f 2c 20 57  #moving "o, W" into eax
a3    04 10 00 06  #moving eax into next memory location
b8    48 65 6c 6c  #moving "Hell" into eax
a3    00 10 00 06  #moving eax into next memory location
b9    00 10 00 06  #moving pointer to start of memory location into
ecx
ba    10 00 00 00  #moving string size into edx
bb    01 00 00 00  #moving "stdout" number to ebx
b8    04 00 00 00  #moving "print out" syscall number to eax
cd    80            #calling the linux kernel to execute our print to
stdout
b8    01 00 00 00  #moving "sys_exit" call number to eax
cd    80            #executing it via linux sys_call
```

Python

- High-level language
 - Is closer to English than most others
- Simple syntax
 - Easy to learn and get stuff done
- Open source
 - Everything is free, lots of things are well-maintained

Growth of major programming languages

Based on Stack Overflow question views in World Bank high-income countries



Great for

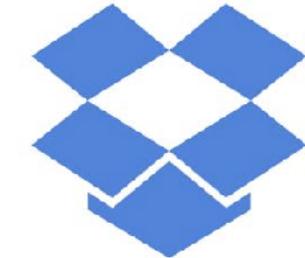
- Prototyping
- Scripting (automation tasks, managing servers)
- Data analysis and machine learning
- Teaching
- Low - medium traffic web apps
- RaspberryPi

Not great for

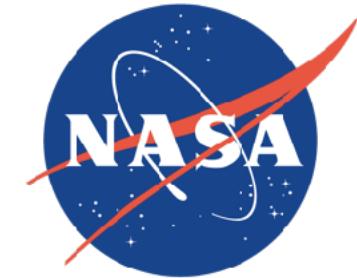
- High speed applications
- Multi-threaded applications
- Mobile development
- Easy to learn, hard to master and progress

Where is it used?

Web apps



Data analysis



More common options

- **Desktop apps**
 - Java, Swift/Objective-C (*Mac*), C# (*Windows*), JavaScript (*with Electron*)
- **Mobile apps**
 - Kotlin/Java(*Android*), Swift/Objective-C (*iOS*), C# (*with Unity*), JavaScript (*with React Native*)
- **High speed, high reliability, multi-threading**
 - C/C++, Go, Rust

About Python



About Python

- Released in 1990
- Created by Guido Van Rossum
- Python Enhancement Proposals (PEPs)



About Python

- Released in 1990
- Created by Guido Van Rossum
- Python Enhancement Proposals (PEPs)



About Python

- Released in 1990
- Created by Guido Van Rossum
- Python Enhancement Proposals (PEPs)

Style Guide (PEP 8)

Indentation

Use 4 spaces per indentation level.

Tabs or Spaces?

Spaces are the preferred indentation method.

Tabs should be used solely to remain consistent with code that is already indented with tabs.

Zen of Python (PEP 20)

Beautiful is better than ugly

Explicit is better than implicit

Simple is better than complex

Complex is better than complicated

Readability counts

...

Try typing “`import this`” into the interpreter.
Next, try typing “`import antigravity`”.

“Code is more often read than written.”

- Guido van Rossum

More about Python

- Why you should learn Python
 - <https://yourstory.com/mystory/interesting-facts-about-python-language>
- Python Developer Survey 2019
 - <https://www.jetbrains.com/lp/python-developers-survey-2019/>
- StackOverflow developer survey 2019
 - <https://insights.stackoverflow.com/survey/2019#technology>
- Python fun facts
 - <https://data-flair.training/blogs/facts-about-python-programming/>

Question & Answer

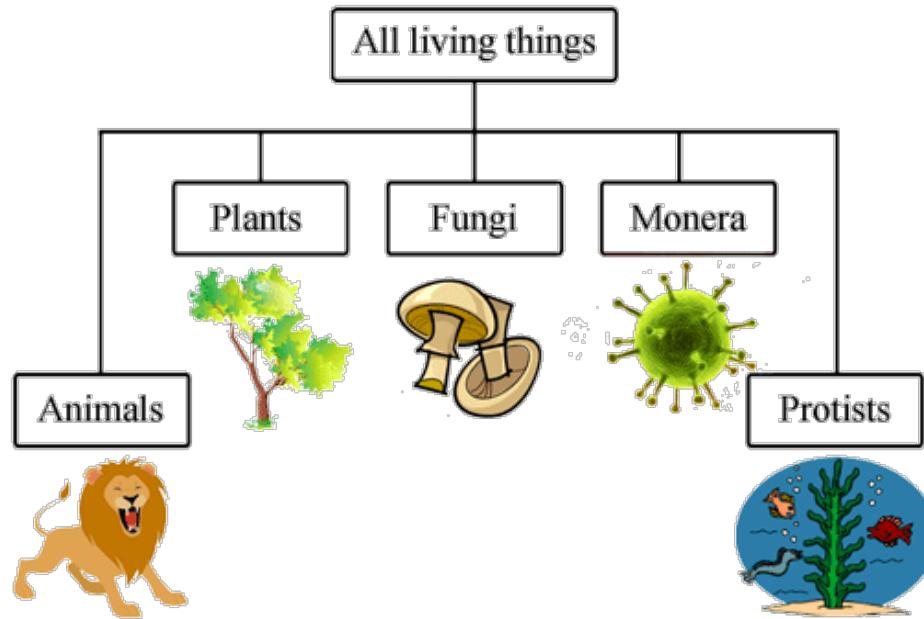


Learn programming basics

Fundamental concepts

- ❑ Types
- ❑ Variables
- ❑ Errors
- ❑ Functions and methods
- ❑ Libraries
- ❑ Comparisons
- ❑ Conditionals (if/else/elif)
- ❑ Looping (while/for)
- ❑ Lists

Types



Types

- String - str - a string of characters, e.g. “Hello 123!”
- Integer - int - whole number
- Float - float - with decimal place
- Boolean - bool - True or False
- NoneType - None - nada, zilch, nothing

Variables



Naming variables

- Starts with letter or underscore

name

_name

- Followed by letters, numbers, or underscores

name_1

- Case sensitive

name_1, Name_1, and name1 are all different

- Readable and descriptive

name instead of n

Keywords in Python programming language

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

<https://www.programiz.com/python-programming/keyword-list>

Built-in Functions				
<code>abs()</code>	<code>dict()</code>	<code>help()</code>	<code>min()</code>	<code>setattr()</code>
<code>all()</code>	<code>dir()</code>	<code>hex()</code>	<code>next()</code>	<code>slice()</code>
<code>any()</code>	<code>divmod()</code>	<code>id()</code>	<code>object()</code>	<code>sorted()</code>
<code>ascii()</code>	<code>enumerate()</code>	<code>input()</code>	<code>oct()</code>	<code>staticmethod()</code>
<code>bin()</code>	<code>eval()</code>	<code>int()</code>	<code>open()</code>	<code>str()</code>
<code>bool()</code>	<code>exec()</code>	<code>isinstance()</code>	<code>ord()</code>	<code>sum()</code>
<code>bytearray()</code>	<code>filter()</code>	<code>issubclass()</code>	<code>pow()</code>	<code>super()</code>
<code>bytes()</code>	<code>float()</code>	<code>iter()</code>	<code>print()</code>	<code>tuple()</code>
<code>callable()</code>	<code>format()</code>	<code>len()</code>	<code>property()</code>	<code>type()</code>
<code>chr()</code>	<code>frozenset()</code>	<code>list()</code>	<code>range()</code>	<code>vars()</code>
<code>classmethod()</code>	<code>getattr()</code>	<code>locals()</code>	<code>repr()</code>	<code>zip()</code>
<code>compile()</code>	<code>globals()</code>	<code>map()</code>	<code>reversed()</code>	<code>__import__()</code>
<code>complex()</code>	<code>hasattr()</code>	<code>max()</code>	<code>round()</code>	
<code>delattr()</code>	<code>hash()</code>	<code>memoryview()</code>	<code>set()</code>	

<https://docs.python.org/3.6/library/functions.html>

Problem #2

Sing Happy Birthday

Happy birthday to you
Happy birthday to you
Happy birthday dear {name}
Happy birthday to you

Problem #3

Temperature converter

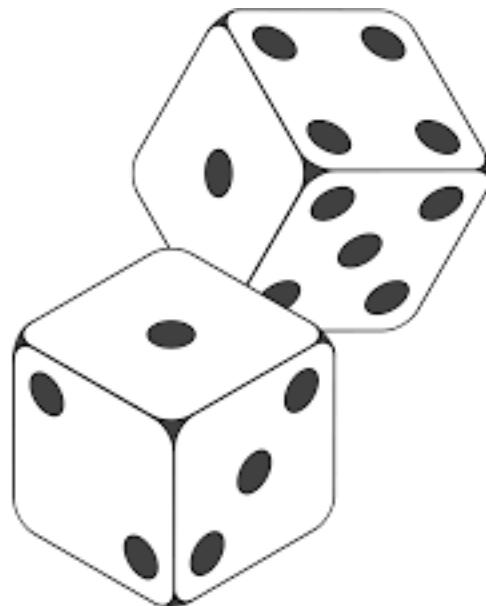
$$T_{(\text{°C})} = (T_{(\text{°F})} - 32) \times 5/9$$

Errors



Problem #4

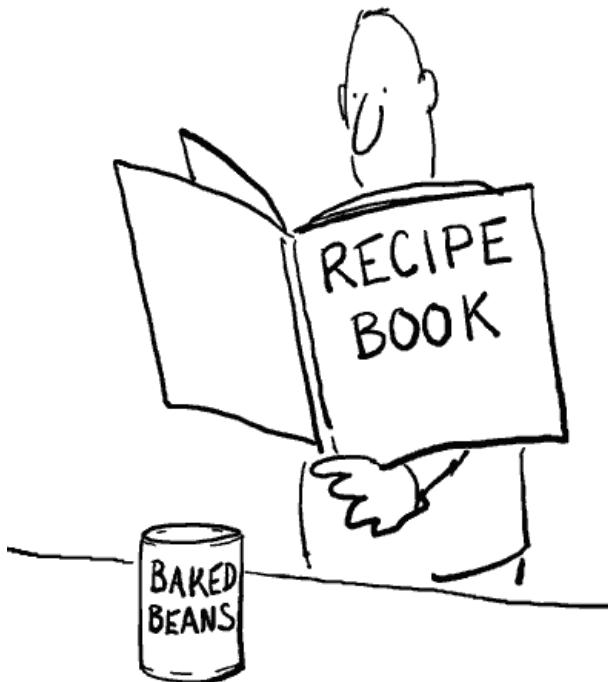
Dice simulator



Libraries



Functions



Problem #5

Circle stats

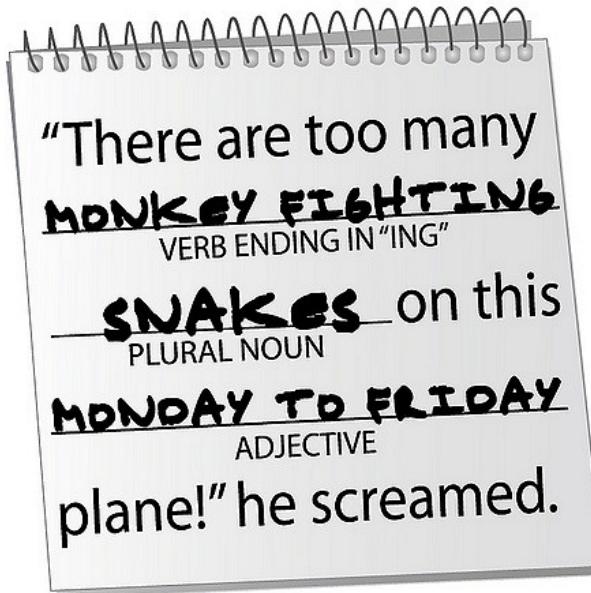
What is your circle's radius?

Methods



Challenge #1

Write a Mad Libs program



Fundamental concepts

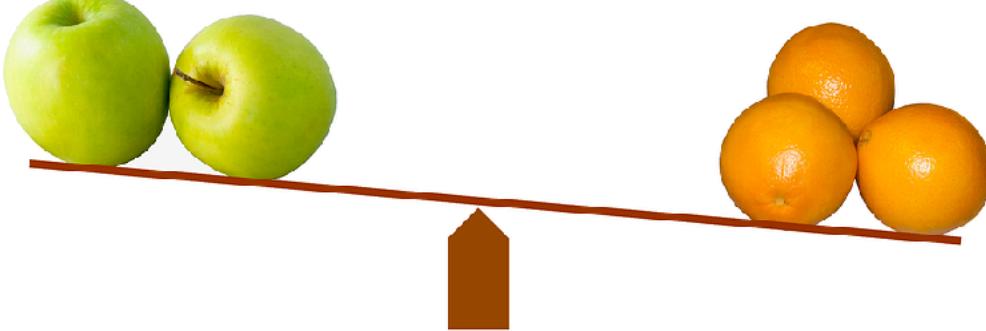
- Variables
- Types
- Errors
- Functions and methods
- Libraries
- Comparisons
- Conditionals (if/else/elif)
- Looping (while/for)
- Data structures (list, dictionary, set, tuple)
- Exceptions



Control the flow with
conditionals

What's True? What's False?

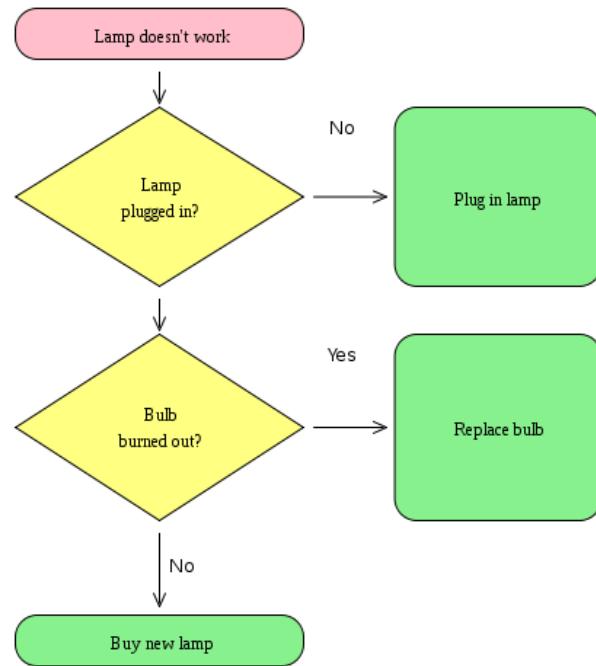
Comparisons



Logical Operators

A	B	A AND B	A OR B	NOT A
False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False

Conditionals



Problem #6

Lucky guess

Guess a number between 1 and 10

Challenge #2

Number guessing game

Guess a number between 1 and 20

Tell them if the answer is higher or lower than their guess

Give the user 4 tries to get it right

Fundamental concepts

- Variables
- Types
- Errors
- Functions and methods
- Libraries
- Comparisons
- Conditionals (if/else/elif)
- Looping (while/for)
- Lists



Work with lists and
loops

Don't repeat yourself!

While loops



Problem #7

New Year countdown

Happy New Year! 

Challenge #2b

Number guessing game

Keep it DRY
Don't Repeat Yourself

Lists



For loops



Problem #8

Vowel counter

Find the bug

while



for



Challenge #3

Word guessing game

Create a list of words

Choose one at random

The user guesses the word, one letter at a time

They have 6 wrong guesses before they lose

Fundamental concepts

- Variables
- Types
- Errors
- Functions and methods
- Libraries
- Comparisons
- Conditionals (if/else/elif)
- Looping (while/for)
- Lists



What to learn next

Software engineering skills

- Handling exceptions
- Testing
- Debugging
- Refactoring

More data structures

- Dictionaries (`dict`)
- Sets
- Tuples

Intermediate Concepts

- Object-oriented programming
 - **Classes**
 - Inheritance
- Intermediate Python
 - **List comprehension**
 - **Nested functions**
 - Decorators
 - Lambda functions

Other essential skills

- Importing external libraries (pip)
- Creating virtual environments (virtualenv / pipenv)
- **Reading/writing to files**
- Making API requests

Word guess bonus solutions

- 3b: word_guess_from_file
 - Choose random word from a separate text file
- 3c: word_guess_with_nested_functions
 - Inner functions can use variables from outer function
- 3d: word_guess_with_classes
 - Creates a WordGame class and define methods on it
- 3e: word_guess_refactored
 - Uses list comprehension
 - Adds some validation so only 1 letter guesses are allowed
 - challenge_3e_word_guess_tests.py tests this version

Specialization

- Data Analysis
 - Jupyter notebooks, **Anaconda distribution**
 - **Pandas, NumPy** for manipulating data
 - **Matplotlib or Seaborn** for visualizations
- Web development
 - **Django** or **Flask** frameworks
 - API creation with **Django Rest Framework, Graphene**
- Scripting
 - Command line, bash
 - Web scraping with **beautiful-soup**, API requests with **requests**

How to learn them

- Tutorials
- Documentation
- Books
- Live Trainings
- Videos
- Courses
- Bootcamps

Next steps

- Project Euler math problems - <https://projecteuler.net/>
- Pick small projects that are appropriate for your level
 - More text-based games
 - Choose your own adventure stories
- Learn intermediate concepts
- Learn Object-Oriented programming (e.g. class)
 - GUIs for your programs (graphical user interface)
 - Make small games with PyGame - [PDF tutorial](#)

Live Trainings by Arianne

- **Programming with Python: Beyond the Basics**
 - Next class: July 7 ([link](#))
 - Beginner - recommended if you're new to programming
- **Object-Oriented Programming in Python**
 - Next class: July 14 ([link](#))
 - Intermediate - recommended if you know Java
- **Rethinking REST: A hands-on guide to GraphQL and queryable APIs**
 - Advanced

Video courses by Arianne

- **Introduction to Python LiveLessons**
 - Lessons 1-4 is the same content as this class
 - Lessons 5-7 is further content
 - [Link](#)
- **Rethinking REST: A hands-on guide to GraphQL and Queryable APIs**
 - [Link](#)

Books

- **Treading on Python Volume 1: Foundations of Python**
- **Python Crash Course: A Hands-On, Project-Based Introduction to Programming**
- **Automate the Boring Stuff with Python: Practical Programming for Total Beginners**
- **Learn Python the Hard Way**

Complementary Live Trainings

- **Business data analytics using Python** - Walter R. Paczkowski
- **Getting started with Python's pytest** - Matt Harrison
- **Introduction to JavaScript Programming** - Elisabeth Robson

Thanks!

Questions?

Email me at arianne.dee.studios@gmail.com