

Week 1: AI Student

[January 23, 2025]

Martín

Tim



Vision

Setup – Visual Studio Code & Github account



Variables and Data Types

```
temperature: int = 25
```

```
distance: float = 19.99
```

```
city: str = "New York"
```

```
is_active: bool = True
```

```
temperature = 25
```

```
print(type(temperature))
```



```
<class 'int'>
```

Loops and Conditionals

```
number = 10
```

```
if number != 0:
```

```
    print(f"{number} is not zero")
```

Loops and Conditionals

```
numbers = [0, 1, 2, 3, 4]
for number in numbers:
    if number != 0:
        print(f"{number} is not zero")
    else:
        print(f"{number} is zero")
```

Loops and Conditionals

```
for fruit in ["apple", "banana", "cherry", "orange"]:  
    if fruit == "banana":  
        continue  
    print(f"I like {fruit}")
```


Loops and Conditionals

```
secret_number = 7
guess = 0
while guess != secret_number:
    guess = int(input("Guess the number: "))
    if guess < secret_number:
        print("Too low!")
    elif guess > secret_number:
        print("Too high!")
print("You guessed it!")
```

Iterable Data Types

```
fruits: list = ["apple", "banana", "cherry"]
```

```
coordinates: tuple = (10, 20, 30)
```

```
unique_numbers: set = {1, 2, 3, 4}
```

```
word: str = "hello"
```

```
for letter in word:
```

```
    print(letter)
```

Iterable Data Types

Element 0

```
fruits: list = ["apple", "banana", "cherry"]
```

```
coordinates: tuple = (10, 20, 30)
```

```
unique_numbers: set = {1, 2, 3, 4}
```

```
word: str = "hello"
```

Unordered

```
for letter in word:
```

```
    print(letter)
```

Iterable Data Types

```
fruits: list = ["apple", "banana",  
"cherry"]
```

```
coordinates: tuple = (10, 20, 30)
```

```
unique_numbers: set = {1, 2, 3, 4}
```

```
word: str = "hello"
```

```
for letter in word:  
    print(letter)
```

Changeable
(mutable)

Unchangeable
(immutable)

Unique elements

Iterable Data Types

key value

```
student: dict = {  
    "name": "Alice",  
    "age": 25,  
    "grade": "A"  
}
```

The diagram illustrates the structure of a Python dictionary. The word 'key' is positioned to the left of the dictionary definition, with three arrows pointing to the strings 'name', 'age', and 'grade' within the dictionary. The word 'value' is positioned to the right, with three arrows pointing to the values 'Alice', 25, and 'A' within the dictionary.

```
student.get("grade")
```

```
Output: 'A'
```

```
for key, value in student.items():
```

```
    print(f"{key}: {value}")
```

Iterable Data Types

```
list(range(10))
```

```
Output: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
for x in range(10):
```

```
    if x % 2 == 0:
```

```
        print(x, sep=" ")
```

```
Output: 0 2 4 6 8
```

Iterable Data Types

```
even_numbers = []  
for x in range(4):  
    if x % 2 == 0:  
        even_numbers.append(x)  
  
for i, e in enumerate(even_numbers):  
    print(f"Index {i}: Number {e}")
```

Output:

```
"Index 0: Number 0 "  
"Index 1: Number 2 "
```

Repetition & Larger Codebases

```
list_1 = [1, 2, 3]
```

```
list_2 = [4, 5, 6]
```

```
list_3 = [7, 8, 9]
```

```
total = 0
```

```
for number in list_1:
```

```
    total += number
```

```
for number in list_2:
```

```
    total += number
```

```
for number in list_3:
```

```
    total += number
```

```
print(total)
```


Functions

```
def sum_list(numbers: list[int]) -> int:  
    total = 0  
    for number in numbers:  
        total += number  
    return total
```

```
list_1 = [1, 2, 3]  
list_2 = [4, 5, 6]  
list_3 = [7, 8, 9]
```

```
total = sum_list(list_1) + sum_list(list_2) + sum_list(list_3)
```

```
print(total)
```

Importing Modules

File tree

```
project/  
├─ main.py  
└─ utilities.py
```

utilities.py

```
def greet(name):  
    return f"Hello, {name}!"
```

main.py

```
import utilities  
  
# Use the custom module  
message =  
utilities.greet("Alice")  
print(message)
```

Classes and Methods

```
import datetime
```

```
class Task:
```

```
    def __init__(self, title: str, description: str, due_date: datetime.date):
```

```
        self.title = title
```

```
        self.description = description
```

```
        self.due_date = due_date
```

```
    def __str__(self):
```

```
        return f"Task(title={self.title}, due_date={self.due_date})"
```

Classes and Methods

```
class RecurringTask(Task):  
    def __init__(self, title, description, due_date, frequency):  
        super().__init__(title, description, due_date)  
        self.frequency = frequency # e.g., "daily", "weekly", "monthly"  
  
print(  
    RecurringTask(  
        "Cook",  
        "Cook dinner for the family",  
        "daily"  
    )  
)
```

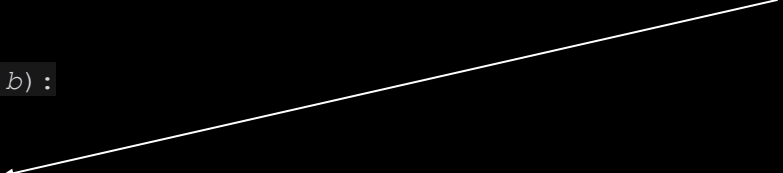
Exceptions

Sometimes your code does not work like you imagined, then exceptions are raised


Example:

```
def divide_numbers(a, b):  
    try:  
        return a / b  
    except ZeroDivisionError:  
        return "Error: Division by zero is not allowed."  
  
print(divide_numbers(10, 2))  
print(divide_numbers(10, 0))
```

You are not allowed to divide by zero



We can catch exceptions instead of having the program crash



Decorators

Library to use for
static type checking
(verifying types)

```
# pip install beartype  
from beartype import beartype
```

```
@beartype  
def divide_numbers(a: int, b: int) -> float:  
    return a / b
```

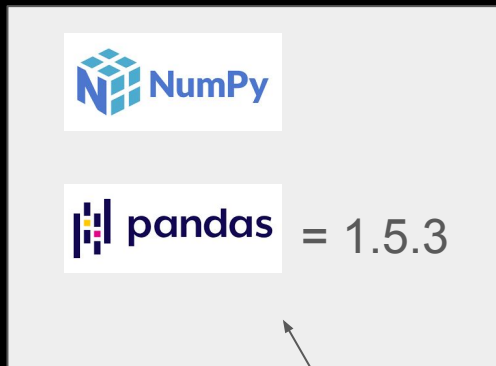
Decorator – Adds
functionality to the
divide_numbers
function

```
print(divide_numbers(10, 2))  
print(divide_numbers(10, 1.0))
```

Environment Setup

Code in VS Code

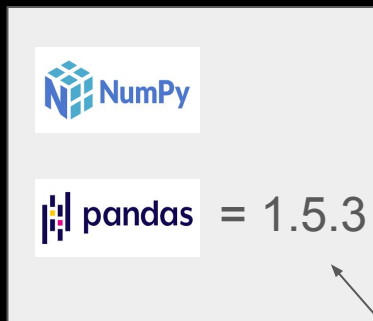
Global environment



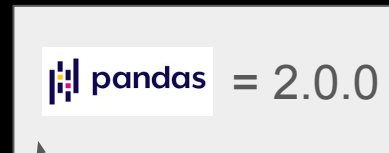
 pandas = 2.0.0

An arrow points from this text block up to the "pandas = 1.5.3" text inside the global environment box.

Project A



Project B



venv's

Two arrows point from this text to the pandas logos in Project A and Project B boxes.

Git & GitHub



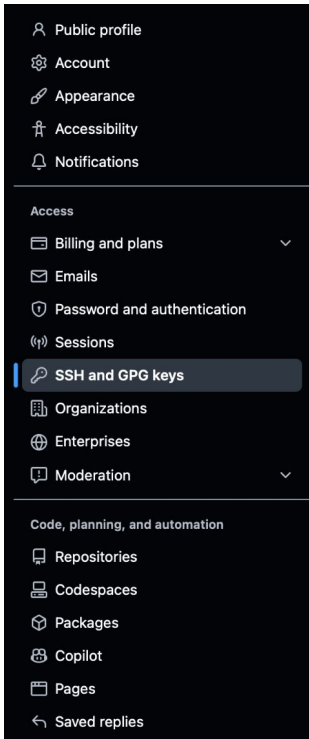
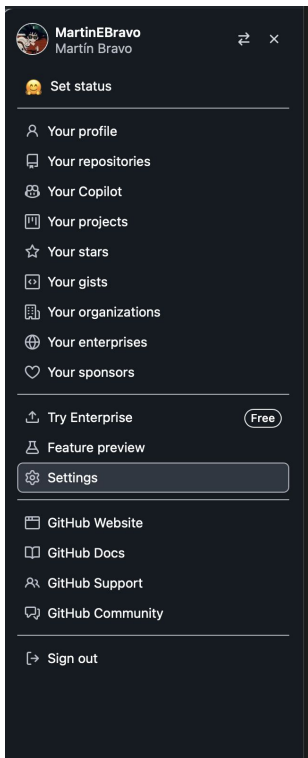
[Generate](#)

[Adding key](#)

Git & GitHub

Generate

Adding key

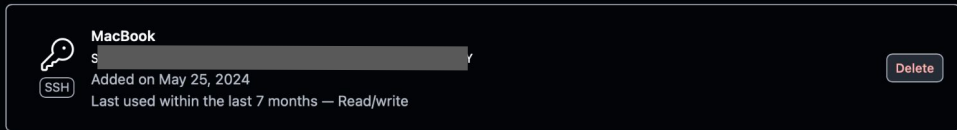


SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys



Check out our guide to [connecting to GitHub using SSH keys](#) or troubleshoot [common SSH problems](#).

GPG keys

New GPG key

There are no GPG keys associated with your account.

Learn how to [generate a GPG key and add it to your account](#).

Vigilant mode

Flag unsigned commits as unverified

This will include any commit attributed to your account but not signed with your GPG or S/MIME key. Note that this will include your existing unsigned commits. [Learn about vigilant mode.](#)



Git & GitHub

`git clone <repo>` → Clone a repository

`git add .` → Add the changes to the future commit

`git commit -m "Initial commit"` → Add a name and commit

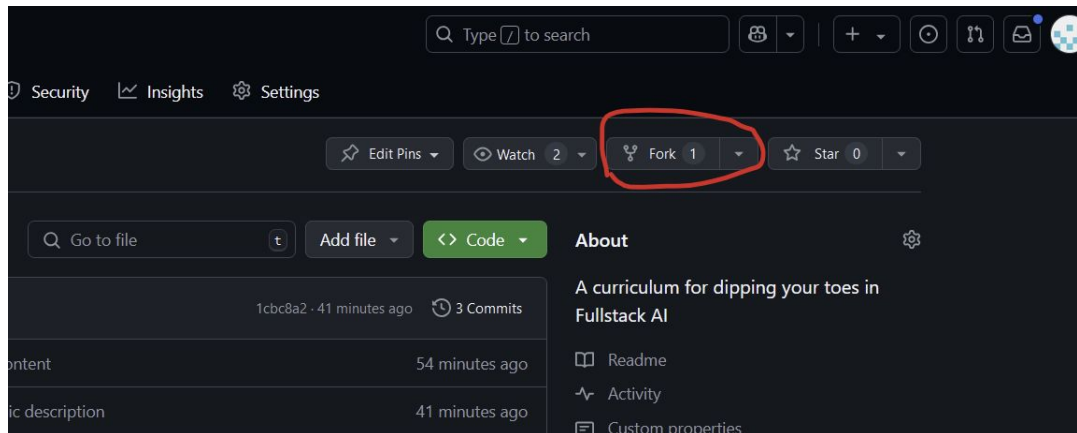
`git push origin main` → Upload the commit to the main branch



Git repo in this course

1. Go to [course github](#)

2. Fork



3. `git clone`
`git@github.com:<your_github_username>/AIStudent.git`