

# Python Official Tutorial — Reference Guide

## Installation & Setup

```
python3 --version
pip install --upgrade pip
python3 -m venv myenv
source myenv/bin/activate
```

## Standard Library Highlights

Module	Purpose	Example
os	File system operations	os.path.exists("file.txt")
json	JSON parsing	json.loads('{"a": 1}')
datetime	Date/time handling	datetime.now()
re	Regular expressions	re.findall(r"\d+", text)
collections	Specialized containers	Counter(["a", "b", "a"])
pathlib	Modern file paths	Path("data") / "file.csv"

## Object-Oriented Programming

```
class Student:
    def __init__(self, name, grade):
        self.name = name
        self.grade = grade

    def is_passing(self):
        return self.grade >= 60

    def __repr__(self):
        return f"Student({self.name}, {self.grade})"
```

# Exception Handling

```
try:  
    result = 10 / 0  
except ZeroDivisionError as e:  
    print(f"Error: {e}")  
except Exception as e:  
    print(f"Unexpected: {e}")  
finally:  
    print("Cleanup done")
```

# Virtual Environments & Dependencies

```
python3 -m venv venv  
source venv/bin/activate  
pip install flask pandas  
pip freeze > requirements.txt  
pip install -r requirements.txt
```

## Key Takeaways

1. Always use virtual environments to isolate project dependencies.
  2. Python's standard library is extremely powerful — explore it.
  3. Use `__repr__` and `__str__` for readable object representations.
  4. Handle specific exceptions, not bare `except:`.
-