



VACATION CLASS

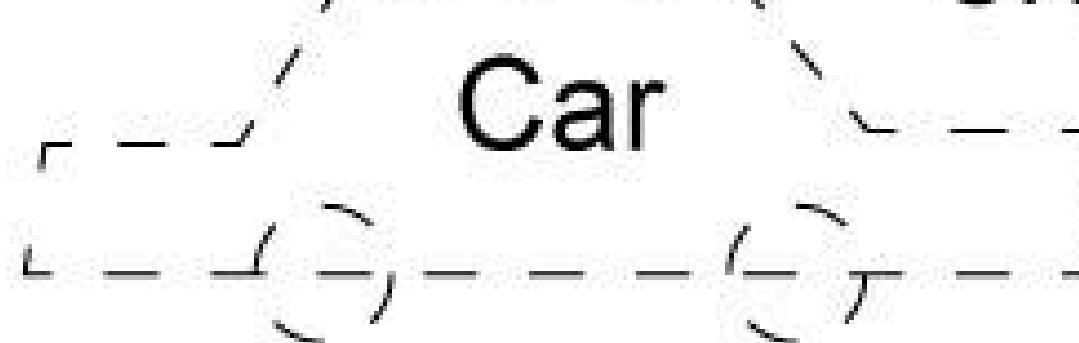
**ALGORITHM AND
DATA STRUCTURE**

Week 3

19 September 2025

YEM DARO

Class



Object

Properties	Methods - behaviors
color	start()
price	backward()
km	forward()
model	stop()

Property values	Methods
color: red	start()
price: 23,000	backward()
km: 1,200	forward()
model: Audi	stop()

CONSTRUCTOR

Usage:

```
class Student():
    def __init__(self, first_name, last_name):
        self.first_name = first_name
        self.last_name = last_name
```

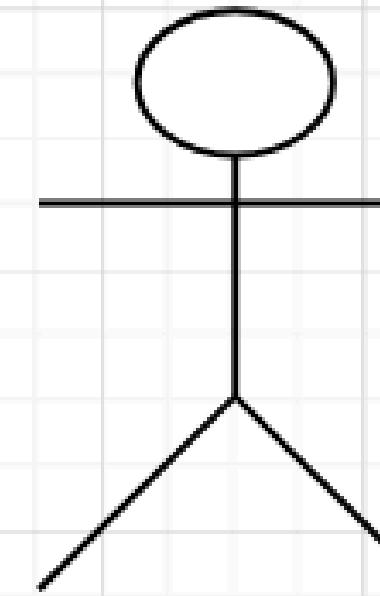
Create / Instantiate a Student Object:

```
s = Student("John", "Cena")
```

INHERITANCE

Parent class

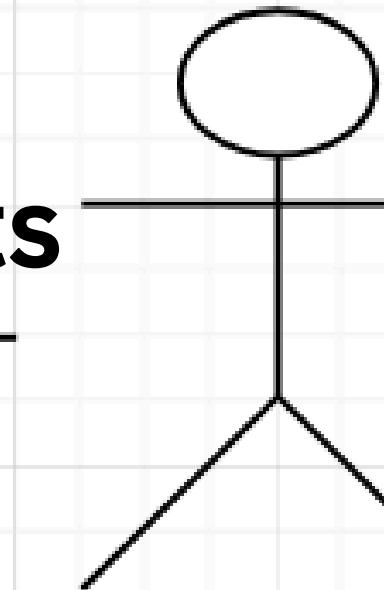
Student



first_name
last_name

Child class

Graduate Student



major

Inherits



CONSTRUCTOR

Method to create an Object
from a Class

```
class Student():
    def __init__(self, first_name, last_name):
        self.first_name = first_name
        self.last_name = last_name
```

Constructor

INHERITANCE

```
class Student():
    def __init__(self, first_name, last_name):
        self.first_name = first_name
        self.last_name = last_name

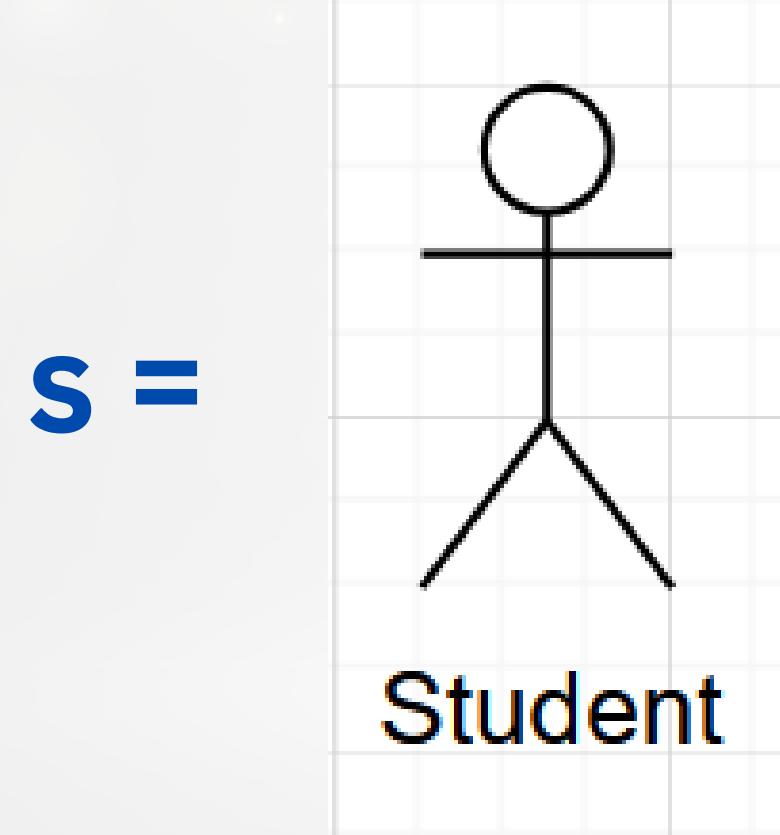
class GraduateStudent(Student):
    def __init__(self, first_name, last_name, major):
        super().__init__(first_name, last_name)
        self.major = 'CS'
```

Graduate Student inherits from Student

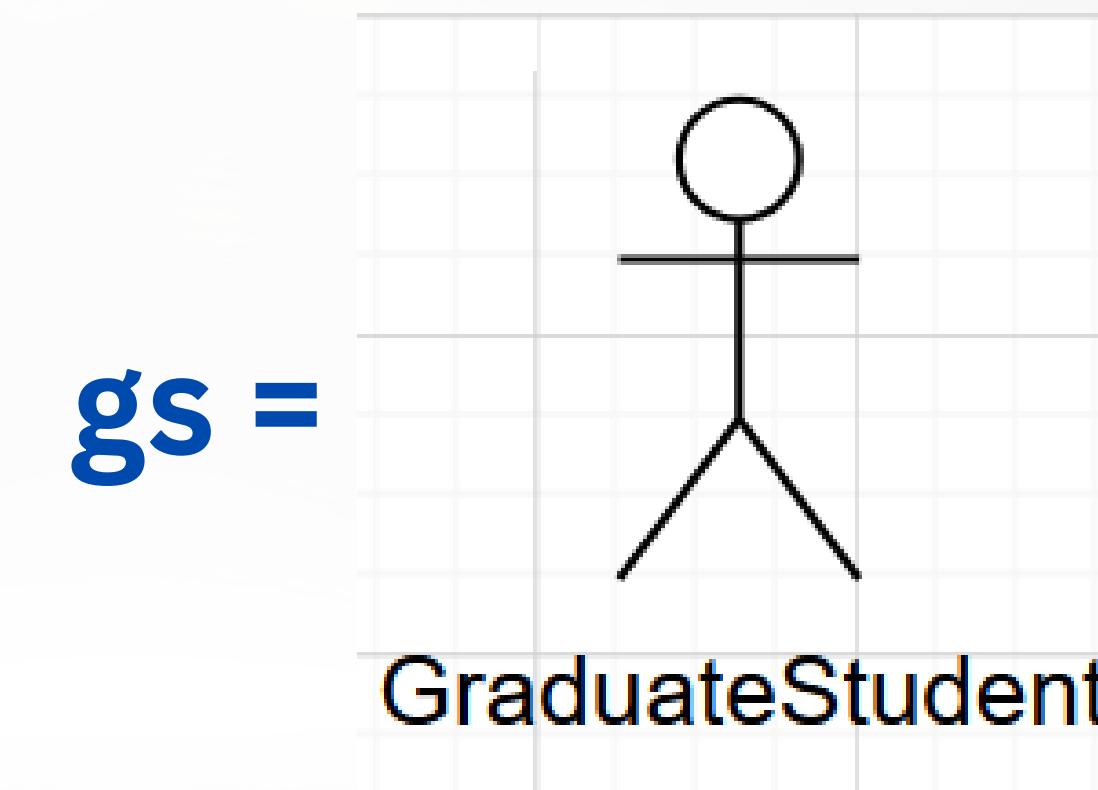
CREATE OBJECTS FROM CLASS

```
s = Student("Alice", 20, "id-1")
gs = GraduateStudent("Charlie", 24, "id-2", "AI", 123)
```

Object



Object



POLYMORPHISM

Function can accept different types

- Different data types
- Objects of different classes

```
def display_info(student):  
    student.display_info()
```

```
display_info(s)  
display_info(gs)
```

Function that accepts
different types

s: Student

gs: GraduateStudent

LET'S REFACTOR PORTAL



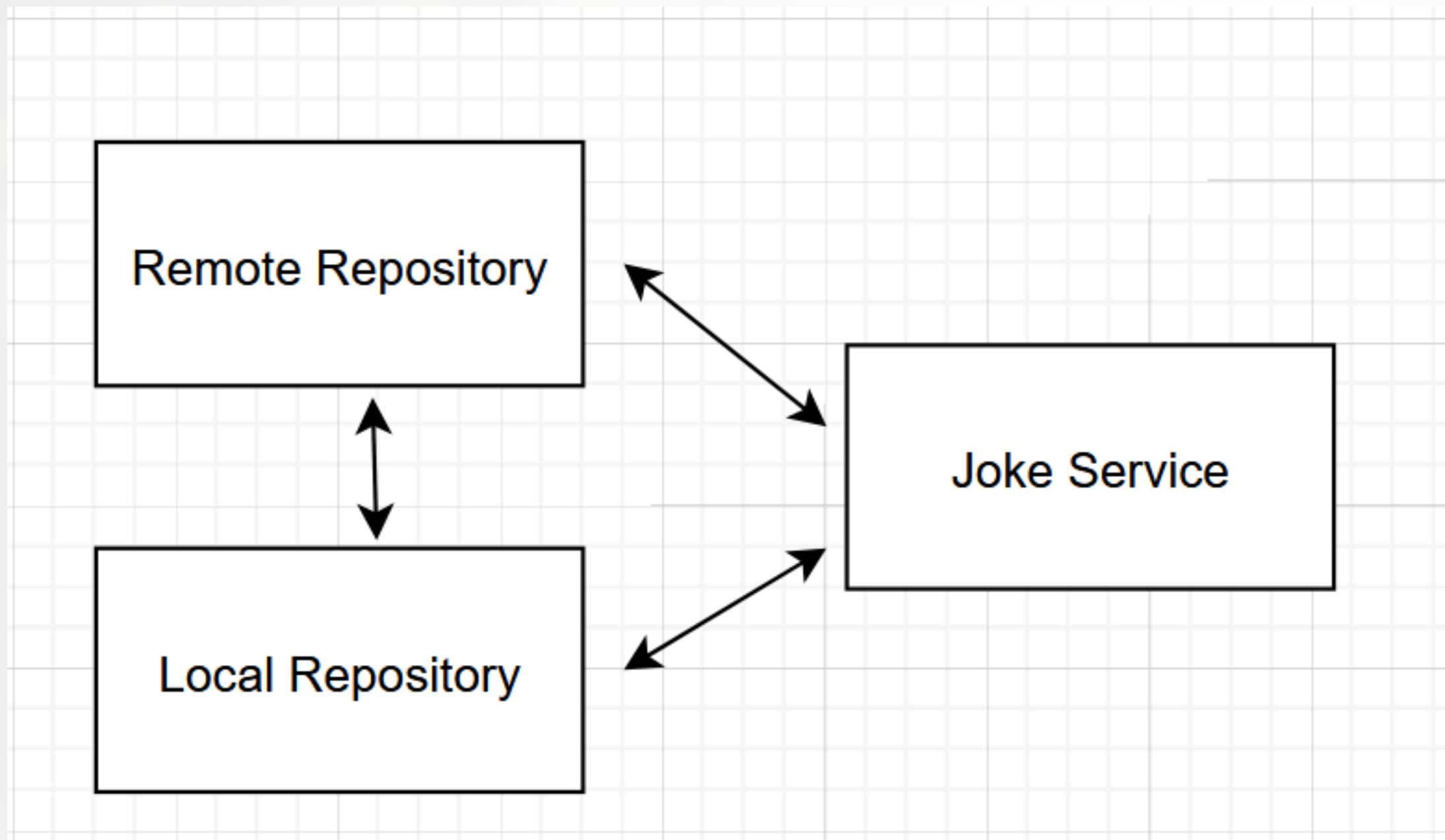
+

oop

github.com/darororo/vacation-class/tree/portal-oop

PRACTICAL DEMO

Application to get Jokes from the internet

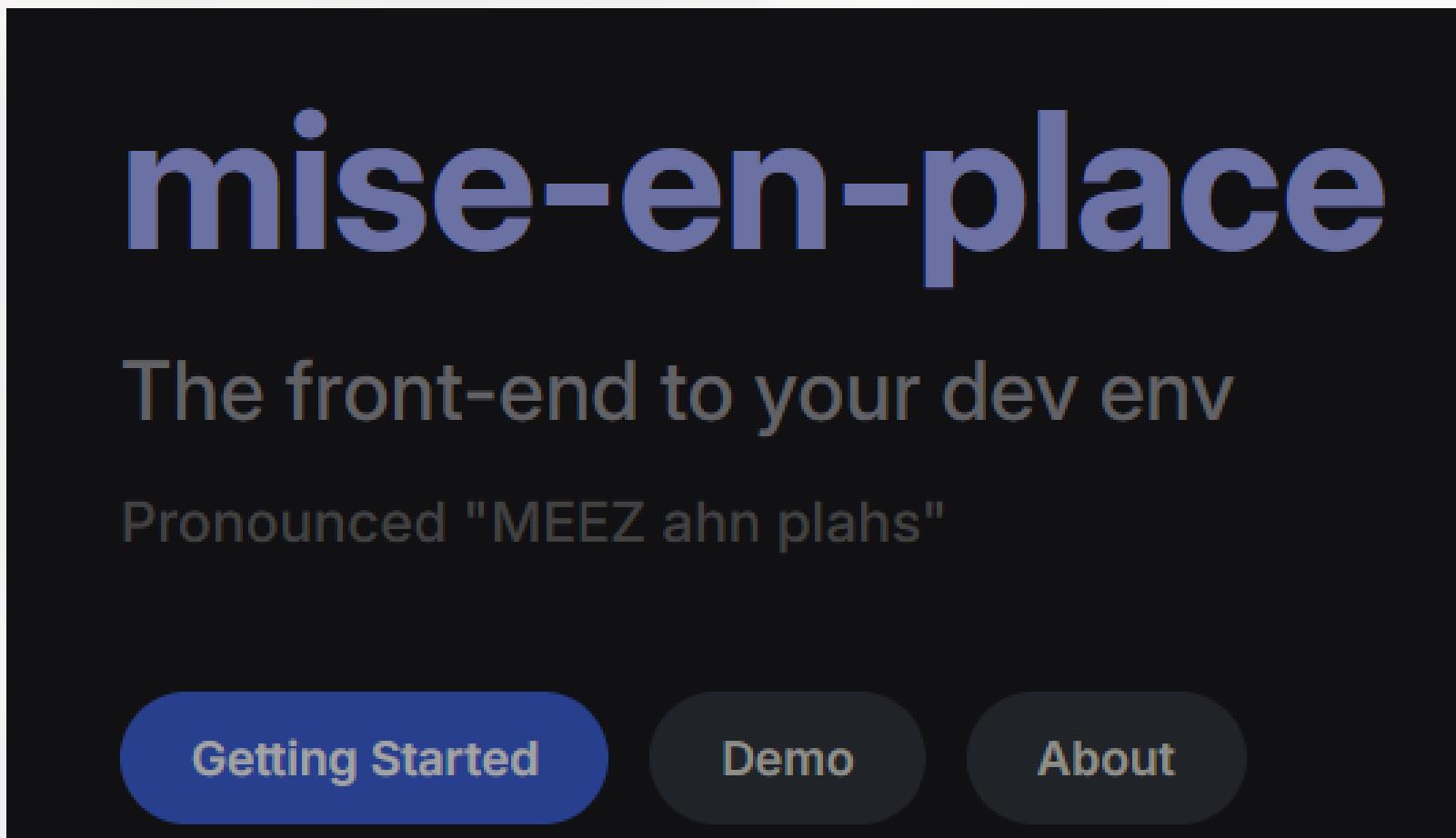


need funny :(



<https://github.com/darororo/vacation-class/blob/oop-02/main.py>

COOL TOOL



The screenshot shows the homepage of the `mise-en-place` website. At the top, the title "COOL TOOL" is displayed in large blue letters. Below it, the main heading "mise-en-place" is shown in a large, stylized blue font. A subtitle "The front-end to your dev env" follows. A pronunciation guide "Pronounced 'MEEZ ahn plahs'" is provided. At the bottom, there are three buttons: "Getting Started" (blue), "Demo" (dark grey), and "About" (light grey).

mise-en-place

The front-end to your dev env

Pronounced "MEEZ ahn plahs"

Getting Started Demo About

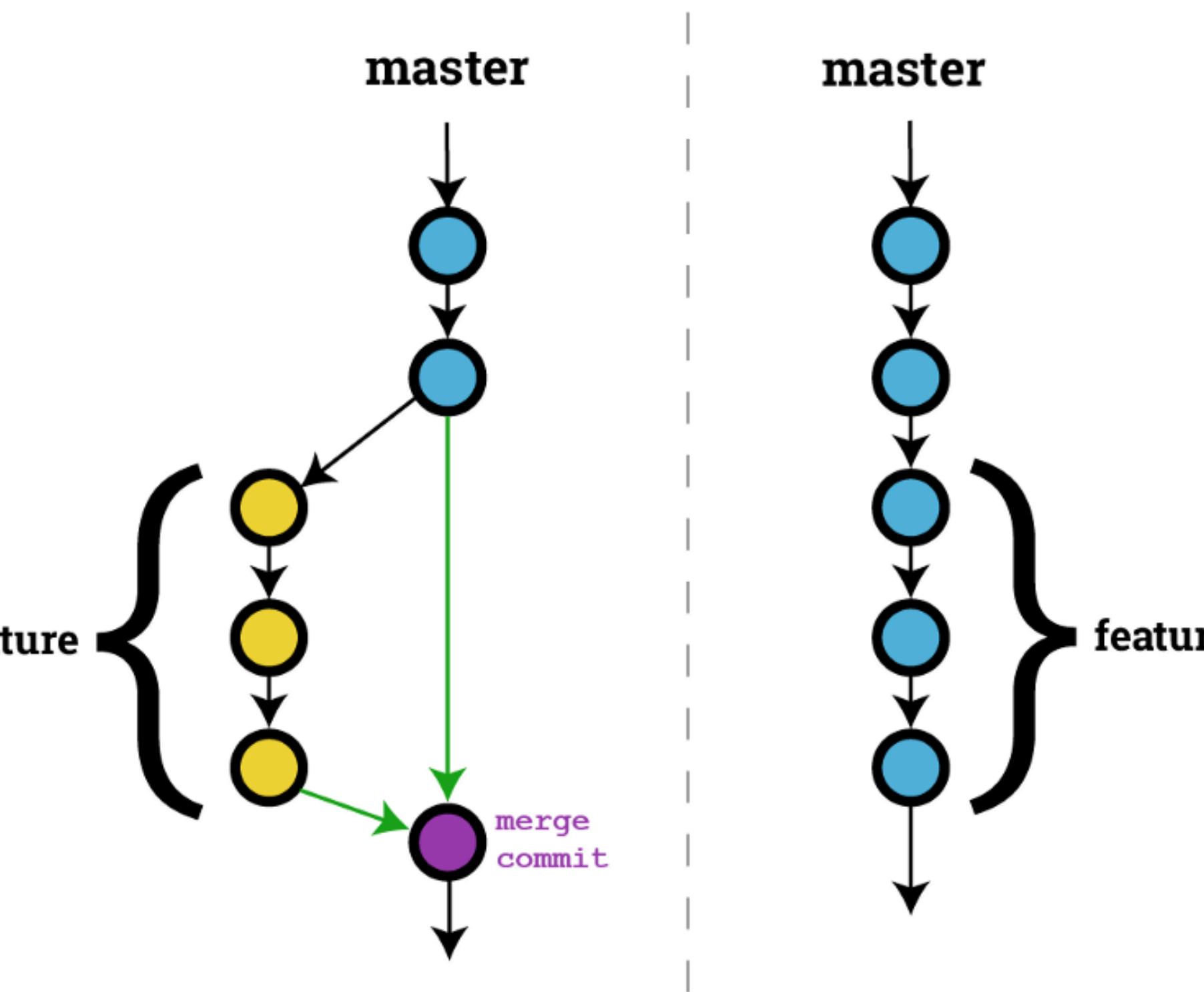


<https://mise.jdx.dev/>

- **Manage development environment**
 - Install languages easily
 - Switch between versions of the same language

GIT FOR TEAMWORK

- Know how to merge codes between branches



git merge master feature

branch

GIT FOR TEAMWORK

- What happens if there are changes on the same location?
 - **Merge Conflict**, it's not that bad :DD
 - **Easily solvable on VSCode**, ...

GIT FOR TEAMWORK

```
E:\Code\vacation-class [portal-oop|MERGING ↑1 +0 ~0 -0 !1 | +0 ~0 -0 !1 !] > git status
(use "git push" to publish your local commits)

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:  main.py
```



conflicted file

GIT FOR TEAMWORK

```
# --- Player class ---
class Player:
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
<<<<< HEAD (Current Change)
  def __init__(self, x, y, radius=25, speed=4, name='john'):
    self.pos = [x, y]
=====
  def __init__(self, x, y, radius=25, speed=4, z=1):
    self.pos = [x, y, z]
>>>>> feature/update-class (Incoming Change)
    self.radius = radius
```

branch name

incoming change

Code in the
current branch

GIT FOR TEAMWORK

Choose what to keep

```
# --- Player class ---
class Player:
    Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
    <<<<< HEAD (Current Change)
        def __init__(self, x, y, radius=25, speed=4, name='john'):
            self.pos = [x, y]
    ====
        def __init__(self, x, y, radius=25, speed=4, z=1):
            self.pos = [x, y, z] CO
    >>>> feature/update-class (Incoming Change)
            self.radius = radius
```

GIT FOR TEAMWORK

```
Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified: main.py

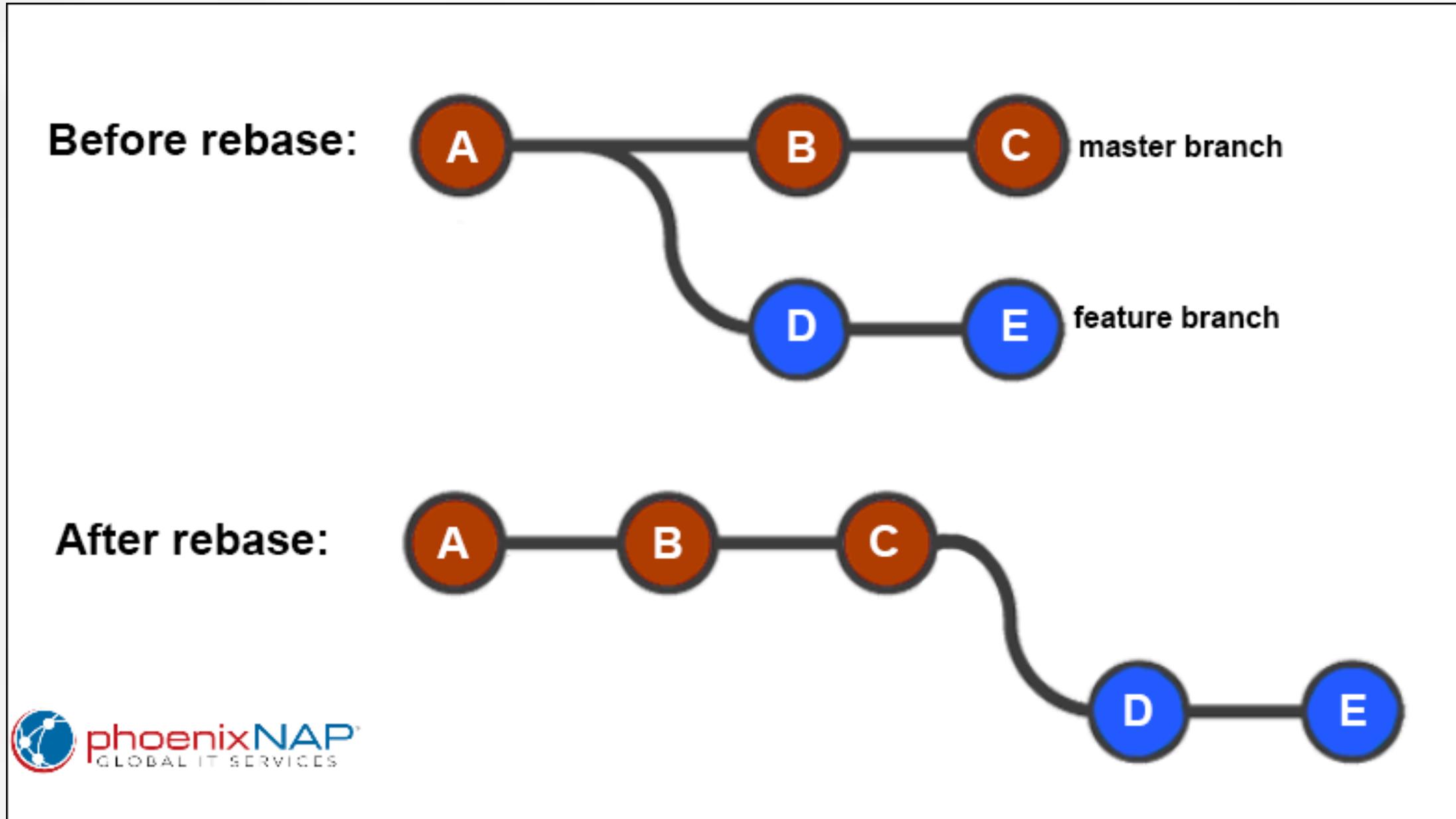
no changes added to commit (use "git add" and/or "git commit -a")
• E:\Code\vacation-class [portal-oop|MERGING ↑1 +0 ~0 -0 !1 | +0 ~0 -0 !1 !]> git add main.py
• E:\Code\vacation-class [portal-oop|MERGING ↑1 +0 ~1 -0 ~]> git commit -m "merge with feat/update-class"
[portal-oop 49bcf62] merge with feat/update-class
```

git add <FILES> Example: **git add .**

git commit : To finish merging

GIT FOR TEAMWORK

To combine your code, you can also use
git rebase



Using only **git merge** is also enough...

GIT TIPS :DD

SHORTCUTS

- git config --global alias.co “checkout”
- git config --global alias.br “branch”
- git config --global alias.lo “log --oneline”
- git config --global alias.cm “commit -m”
- git config --global alias.st “status”

COOL PLACE TO LEARN

Coding

<https://www.youtube.com/@BroCodez>

<https://www.youtube.com/@FireShip>

Game + Math <https://www.youtube.com/@acegikmo>

Graphics

<https://www.youtube.com/@javidx9>

https://www.youtube.com/@Acerola_t

COOL PLACE TO LEARN

Be Curious ! :DD





THANK YOU !

:DDDD