Introduction to Python & Git/GitHub

bangkit

Learning Objectives

1. Depict how to program with Python
2. Write a simple program in Python
3. Define about Git and GitHub
4. Use Git and GitHub to track changes and collaboration

bangkit

Today's Agenda

- Introduction to Python
- Python Data Structures
- Introduction to Git
- Collaboration using GitHub

bangkit

Introduction to Python

bangkit

Hello, Python

What is Python? A dynamic, interpreted (bytecode-compiled) language.

Why programming with Python?

- Easy syntax
- Omnipresent
- Most chosen language for IT
- Versatile

Hello, World! in Python

```python
print('Hello, World!')
```

Python Environment Setup

How to install Python?

- Official Python distribution
- Anaconda distribution

How to check an installed Python?

- open a terminal or command prompt.
- execute Python command.
- passing **--version** as a parameter.

Check on terminal

```
$ python --version

$ python3 --version
```

Check on Command Prompt or PowerShell

```
C:\Users\bangkit> python --version
```

bangkit

Basic Python Syntax: Data Types

String (str)	Text.
Integer (int)	Numbers, without fraction.
Float (float)	Numbers with fraction.
Boolean (bool)	Data type which only has two values (True/False).

bangkit

Basic Python Syntax: Variable

- Name to certain values.
- The process of storing a value inside a variable is called an **assignment**.
- Can only be made up of **letters**, **numbers**, and **underscore**.
- Can't be Python **reserved keywords**.

Do This

```
length = 10
width = 2
name = "Saturnus"
```

Don't Do This

```
def = "Function"
class = "Class"
```

Cannot be used, will expected error

bangkit

Basic Python Syntax: Operators

Operators are used to perform **operations** on **variables** and **values**.

Arithmetic	Identity
Assignment	Membership
Comparison	Bitwise
Logical	

```
a = 10
b = 30.0
a + b
```

Operator

```
print(1 < 10)
#True

print("Linux" == "Windows")
#False

print(1 != "1")
# True
```

bangkit

Basic Python Syntax: Functions

- Define function with def keyword.
- Function has body, written as a block after colon in function definition. The block has indented to the right.
- To get value from a function use the return keyword.

Call Function

Function name Input/Parameter

```
def greeting(name):
    return 'hello, ' + name
```

Indentation with 4 spaces

Call Function

```
print(greeting('bangkit'))
# hello, bangkit
```

bangkit

Basic Python Syntax: If Statements

- The ability of a program to **alter its execution sequence** is called branching.
- The if block will be executed only if the condition is True.
- Use elif & else statement to handle multiple conditions.

Condition of Comparison

Condition Evaluations

```
if hour < 12:
  print("Good morning!")

def check(number):
  if number > 0:
    return "Positive"
  elif number == 0:
     return "Zero"
  else:
     return "Negative"
```

Basic Python Syntax: Loops

- The ability of a program to **execute a sequence multiple times** is called looping.
- while loop instruct computer to continuously execute code based on the value of a condition.
- for loop iterates over a sequence of values.

while loop

```
x = 7  # Initialization of variable

while x > 0:
  print("positive x=" + str(x))
  x = x - 1

print("now x=" + str(x))
```

The sequence

for loop

```
for x in range(3):  # 0, 1, 2
  print("x=" + str(x))
```

bangkit

Basic Python Syntax: break & continue

- Both while and for loops can be interrupted using the break keyword.
- Use the continue keyword to skip the current iteration and continue with the next one.

break

```python
for x in range(3):
  print("x=" + str(x))
  if x == 1:
    break  # quit from loop
```

continue

```python
for x in range(3, 0, -1):
  if x % 2 == 0:
    continue  # skip even
  print(x)
```

Python Data Structure

bangkit

What is Data Structure?

A specific container (variable) type that can be used to **collect**, **access**, and **organize data** more efficiently.

List | Tuple | Dictionary

Data Structure: List

- In Python list can contain a different value.
  - Ordered
  - Mutable
- Python use square brackets [] to indicate where the list starts and ends.

```python
paths = ['ML', 'Cloud']

print(paths[0])

paths.append('Android')
paths.remove('Cloud')
paths.insert(1, 'Mobile')
paths.pop(-1) # remove 'Android'
```

Data Structure: Tuple

- In Python tuple can contain a different value.
  - Ordered
  - Immutable
- Python using parentheses () to indicate where the tuple starts and ends.

```
paths = ('ML', 'Cloud')

print(paths[0])

paths[1] = 'Machine Learning'
```

Error

Data Structure: Dictionary

- Dictionary in Python contain **pairs of keys and values**.
- To get a **dictionary value**, use its corresponding **key**.
- Dictionary in Python are **mutable**.
- Python use curly brackets {} to indicate where the dictionary starts and ends.

```python
students = {
  'ml': 500,
  'mobile': 700,
  'cloud': 900
}

print(students['cloud'])  # 900

students['ml'] = 1000
```

Introduction to Git

bangkit

What is Git?

An **Open Source VCS** (Version Control System) created in 2005 by Linus Torvalds.

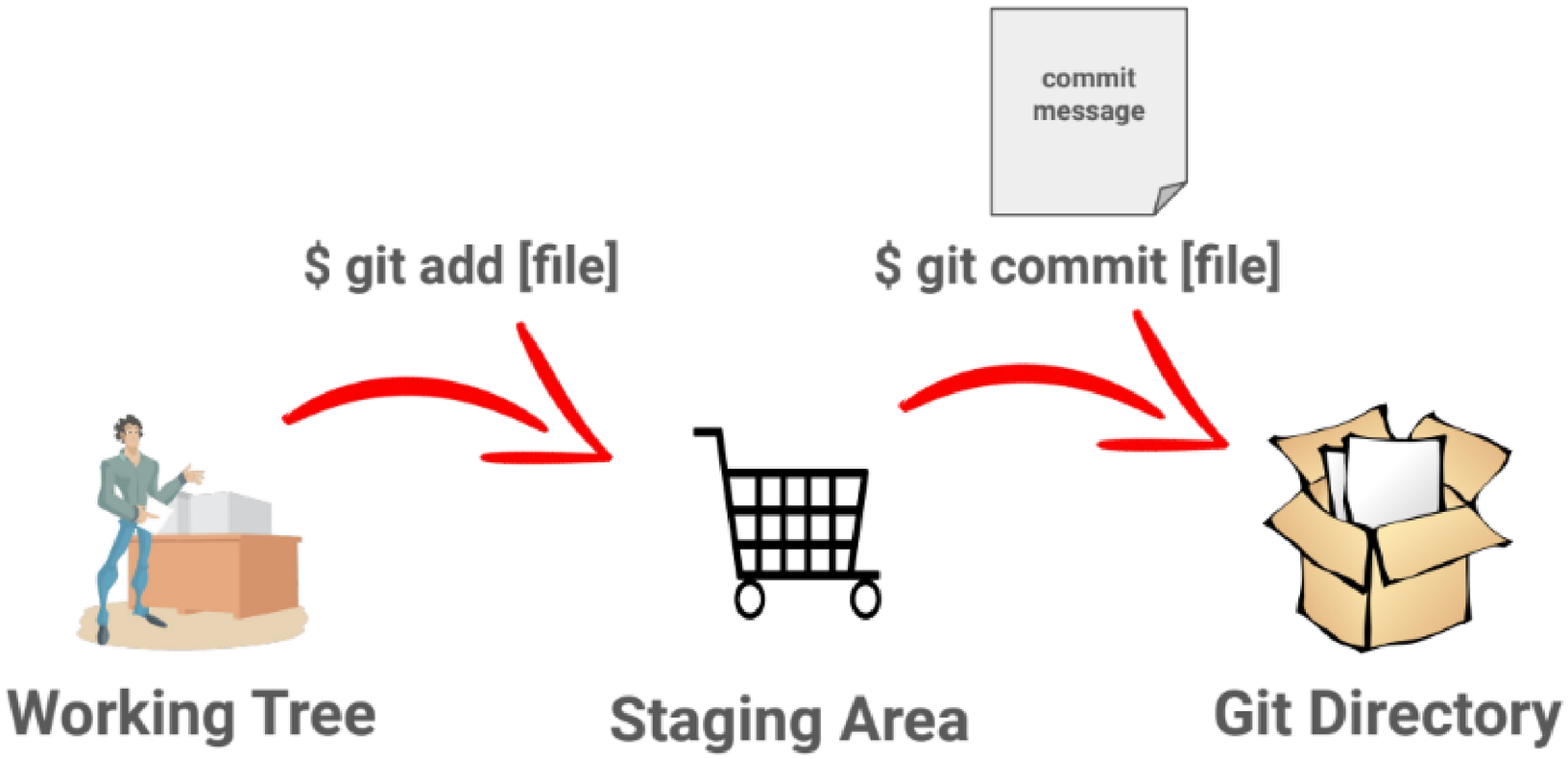
Can help us to:

Keep Track

Rollback

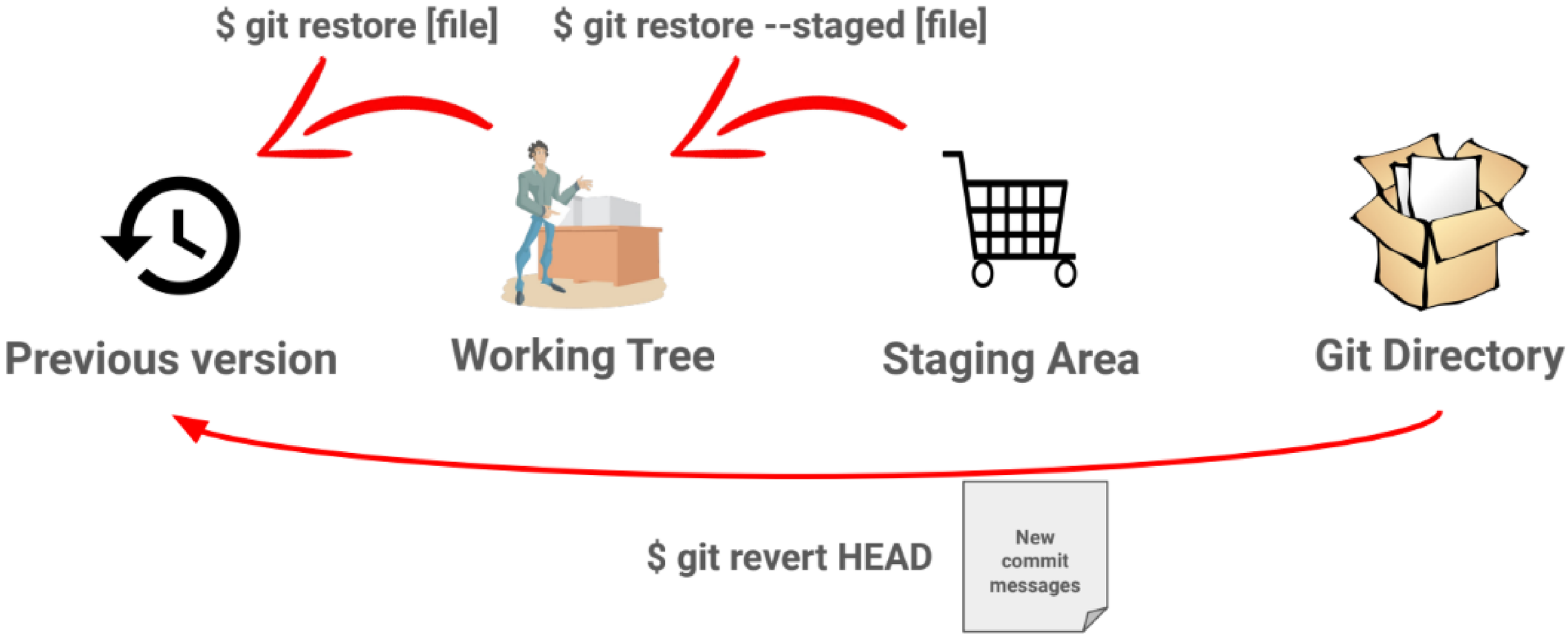
Collaborate

Workflow in Git



bangkit

Undoing Things on Git



bangkit

Branching

- In Git, a **branch** at the most basic level **is just a pointer** to a particular commit.
- git branch command will show a list of all the branches in your repository
- git branch [branch-name] command will create a new branch.
- git checkout [branch-name] command will switch to another branch.

```
git branch
* master

git branch new-feature
git branch
* master
  new-feature

The current branch

git checkout new-feature
git branch
  master
* new-feature
```

Merging

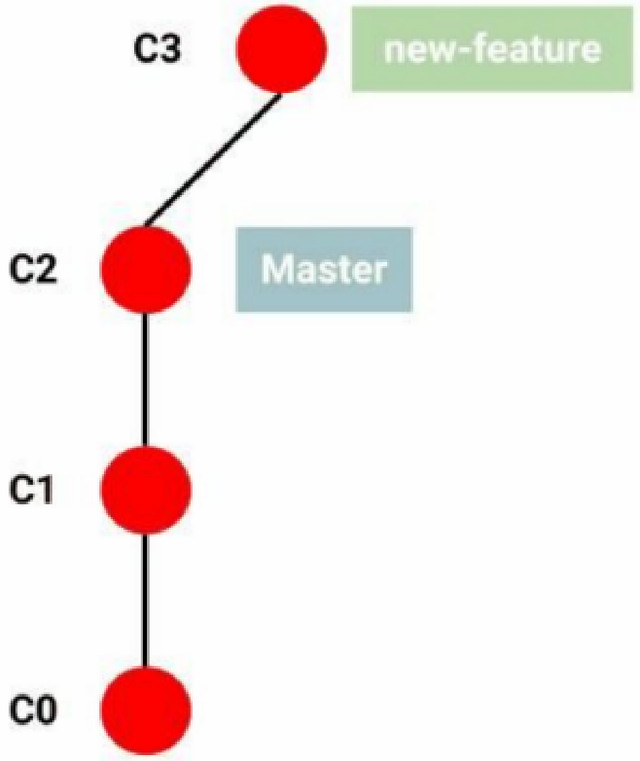
- Merging is the term employed by Git to combine branched data and history together.
- Use git merge [branch-name] command to merge the specified branch's history into the current one.

```
git merge new-feature

git branch -d new-feature
```

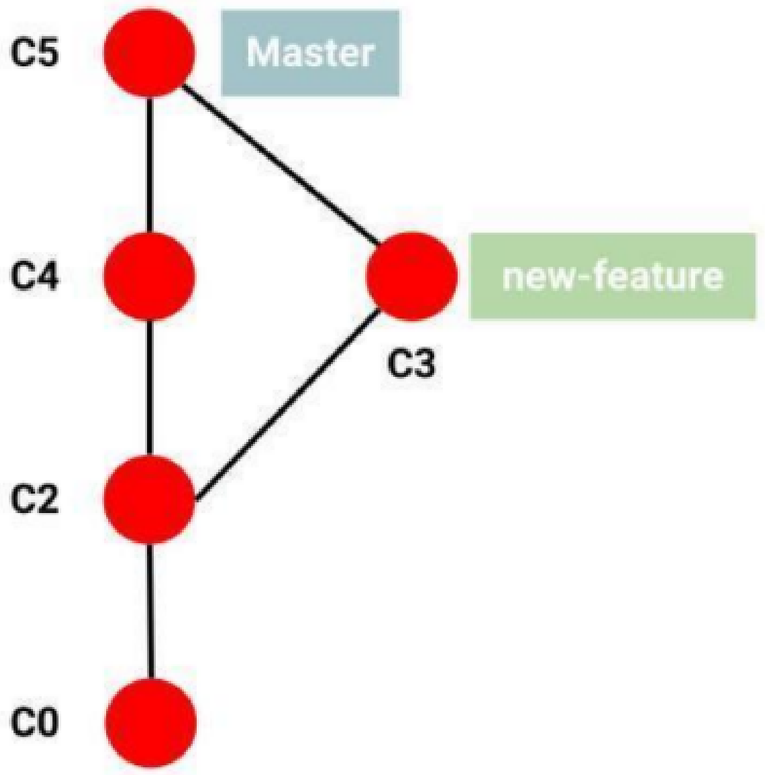
Fast-Forward Merge

This kind of merge occurs when all the commits in the checked-out branch are also in the branch that's being merged.



Three-Way Merge

A three-way merge is performed when the history of the merging branches has diverged in some way, and there isn't a nice linear path to combine them via fast-forwarding.



Collaboration using GitHub

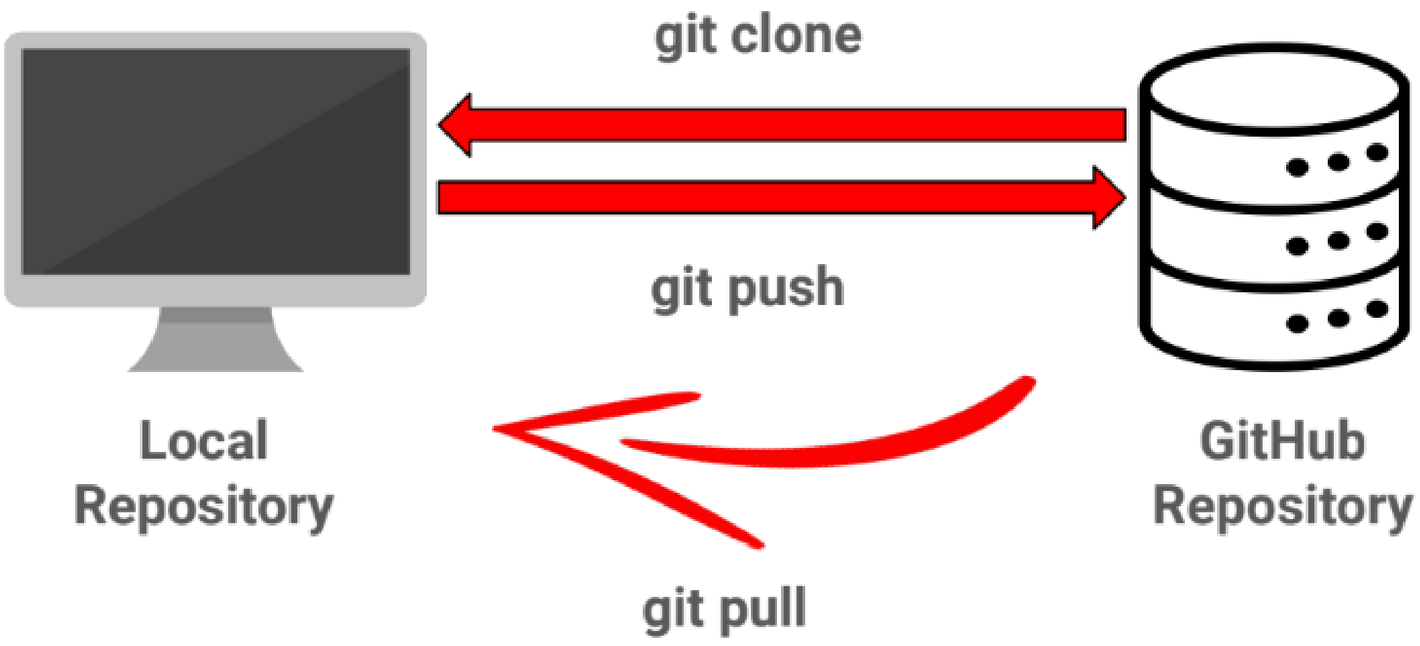
bangkit

Introduction to GitHub

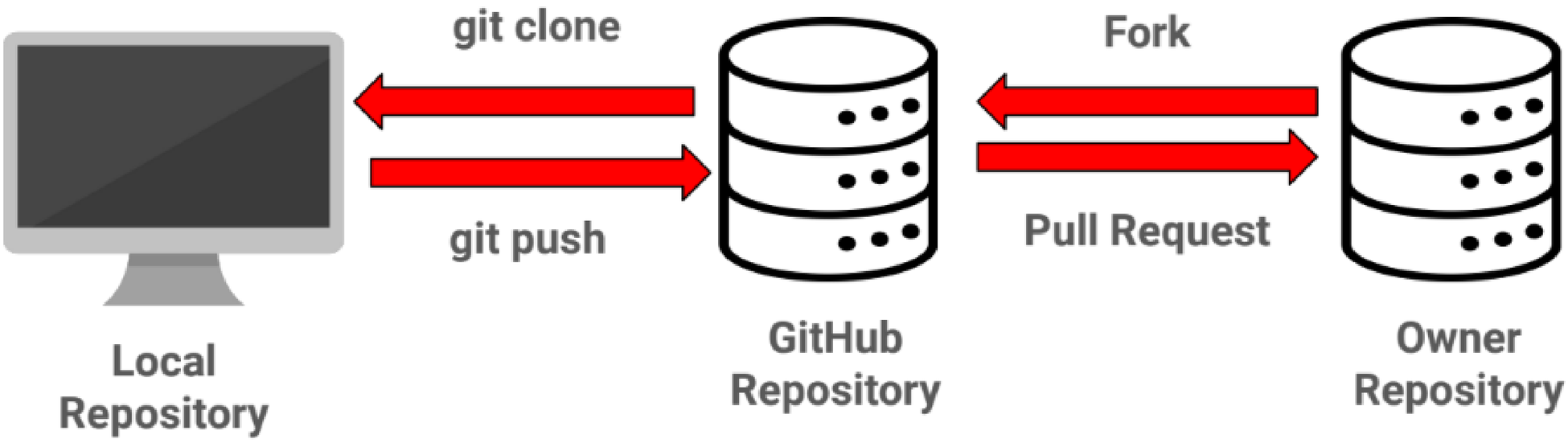
GitHub (https://github.com) is a web-based Git repository hosting service

Many open-source projects use GitHub for **Git hosting**, **issue tracking**, **code review**, and other things

Working with GitHub



Collaborate Using GitHub



Tips for Solving Conflicts

1. **Always synchronize** the branches before starting any work.
2. Try to make **changes as small as possible** as long as they're self-contained
3. **Regularly merge** changes made on the master branch back onto the feature branch
4. It's a common practice to have **the latest version** of the project in the master branch and a **stable version** of the project on a separate branch.

You have learned basic Python and understand how Git and GitHub can help developers to collaborate.

These knowledges will be helpful for your day-to-day work as a developer in the future. Now, you can write a simple Python program and be ready to collaborate using Git/GitHub!

bangkit

Thank You

bangkit