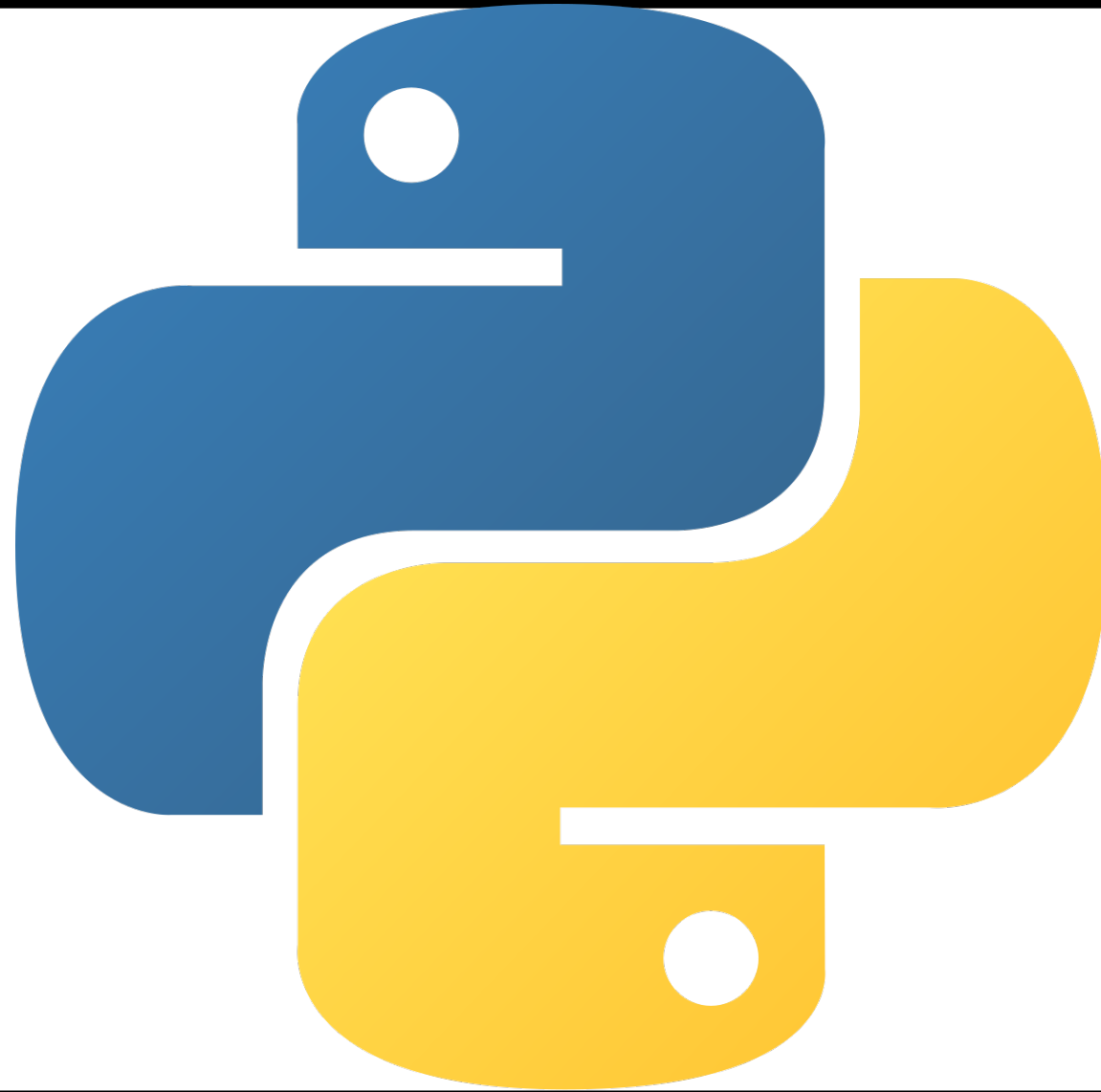

PYTHON PROGRAMMERING



Föreläsning 14

DAGENS FRÅGA

- Vilken hemsida besöker du mest?

DAGENS AGENDA

- Repetition

FÖRRA FÖRELÄSNING

- Vad är ett API?
 - API på internet – HTTP
 - CRUD för HTTP
 - HTTP request
 - Localhost
 - RESTful API
 - Ännu ett objektorienterad programmerings exempel, `pokemonhandler.py`
-

VAD HAR NI LÄRT?

1. Utveckla pythonskript
 2. Köra kod både via terminal och i Jupyter Notebooks
 3. Följa konventioner inom språket Python
 4. Modifiera filer och filstrukturer med Python
 5. Använda Python på ett objektorienterat sätt
 6. Koppa Python ihop med SQL databas
 7. Visualisera och analysera data
 8. Testa och debugga er kod
 9. Interagera med API:er
-

README

JUPYTER NOTEBOOK KOMMANDON

- Enter – redigera läge
- Ecs – kommando läge
- Shift+enter – run cell
- Alt+enter – run cell och infoga ny cell under
- Gömma output - dubbelklicka vänster sida av output eller använd ;

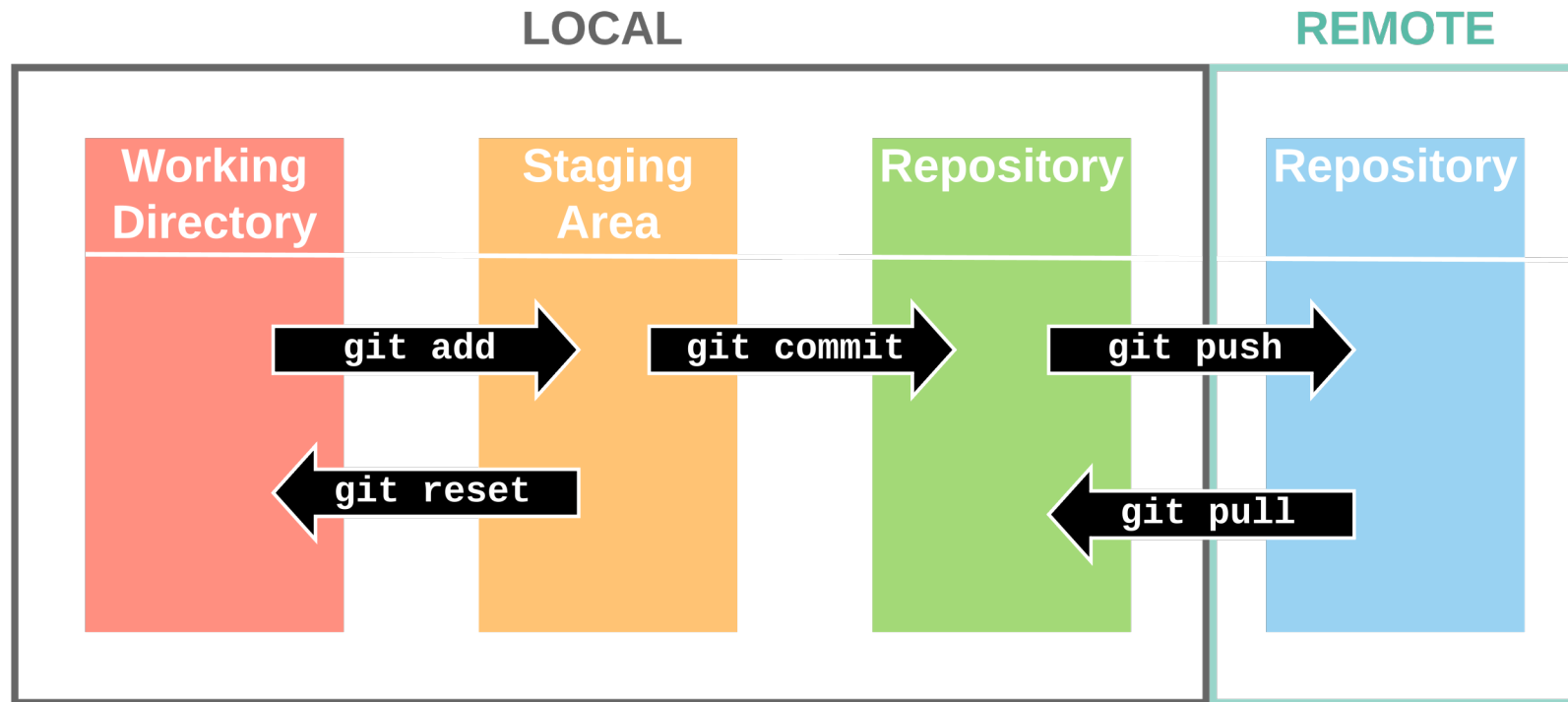
När man är i kommando läge (Esc):

- a – ny cell över
- b – ny cell under
- x – klipp ut vald cell
- c – kopiera vald cell
- v - klistra in cell
- d+d – radera cell
- z – ångra raderad cell
- s – spara
- m – ändra cell till Markdown
- y – ändra cell till kod
- o – göm output/vis output

MARKDOWN

- <https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>

VERSIONSHANTERING - GIT



GENERELLT

- Variabler: behållare som har en typ och ett namn
 - Tilldelningsoperatoren: =
 - Matematiska operatorer: +, -, *, /, %, ()
 - Tilldelning och matematisk operator: +=, -=, *=, /=, %=
 - Datatyper: int, float, string, boolean
 - Casting: konvertera datatyper. str(), int(), float()
 - Print: print()
 - f-string: print(f"Hello {var}!")
 - Läs från konsol: input()
 - Slicing sequences: var[start : stop]
-

DATASTRUCTURES

- List
 - indexerade startar på 0
 - `list = [1, 2, 3]`
 - `list[0] = 1`
 - Tuples
 - Indexerade, men kan inte ändra ordning
 - `tuple = (1, 2, 3)`
 - Dictionaries
 - Key-value par
 - `dict = {'key': 'value'}`
 - Inte duplicate keys men duplicate values
 - `dict['key'] = value`
 - Sets
 - Samling unika, oindexerade element
 - `set = {'one', 'two', 'three'}`
-

CONDITIONS & LOOPS

- Conditional statement:
 - if True:
else:
 - Operators: ==, !=, <, >, <=, >=, and, or, not
 - elif
 - For-loops
 - Iterera genom en sekvens
 - for i in range(x):
 - While-loops
 - Utföra så länge given condition är True
 - while cond:
-

METODER

- Functions
 - Ett block kod som bara körs när den kallas på
 - `def function_name(param1, param2 = value):`
 - function code*
 - `return`
 - Main-metod
 - Startpunkt för att köra ett Python-script
 - `if __name__ == "__main__"`
 - Module/library
 - Samling av funktioner i en egen fil som kan inkluderas i ett script
 - Importeras: `import module`
 - Kommentera med docstring
 - `""" [summary]`
Args:
 - `param ([type]): [description]`
-

PANDAS

- Series
 - Endimensionell datavektor med index
 - `pd.Series(data)`
 - Komma åt data: `series['index']`
 - DataFrame
 - Tvådimensionell datastruktur med columns och rows
 - Varje kolumn är en Series
 - `pd.DataFrame(data)`
 - Komma åt data: `df['column']`
 - Index
 - `iloc` – interger location
 - `loc` – location
 - `[]` returns Series
 - `[[]]` reurns DataFrame
-

OBJEKTORIENTERAD PROGRAMMERING

- OOP grundpelar
 - **Abstraktion:** att simplificera användning genom att gömma implementering → interface
 - **Enkapsulering:** att samla funktionalitet och data på ett ställe → classes
 - **Arv (Inheritance):** att kunna återanvända gemensam funktionalitet → parent classes
 - **Polymorfism:** att kunna ändra beteendet av gemensam funktionalitet → overriding
 - Klasser
 - Gruppera data och functioner
 - `class ClassName():`
 - Object
 - När man kallar på klassen till en variabel
 - Metoder
 - Funktioner som skapas i en klass
 - `def method(self):`
-

OBJEKTORIENTERAD PROGRAMMERING

- Konstruktor
 - Metod som anropas automatisk när ett objekt av klassen skapas
 - Konfigurerar allt som behövs för att klassen ska fungera
 - `__init__(self)`
 - Instans variabler
 - Unika default värden för varje instans av en klass som skapas i konstruktorn
 - `self.var_name = var_name`
 - Klassvariabel
 - Skapas direkt under klassen och över konstruktorn
 - Alla objekt av klassen kan nå variabeln
 - Self
 - För att komma åt variabler och metoder från klassen
 - Self är en referens till sig själv. Alle metoder i klassen måste ha self för att komma åt instansvariabler
 - Privat metod
 - Hjälpmetod
 - Användas bara inuti klassen
 - `def _privat_method(self):`
-

RELATIONSDATABAS

- Kopplade tabeller med rader och kolumner
 - Primary key – en eller flera kolumner som skapar unika värden
 - Foreign key – kopplar tabeller ihop
 - CRUD i SQL
 - Create → INSERT: INSERT INTO table
 - Read → SELECT: SELECT * FROM table
 - Update → UPDATE: UPDATE table SET column1 = y
 - Delete → DELETE:
 - Datatyper: integer, float, text, bool, null
 - SQLite i Python
-

ERRORS OCH EXCEPTIONS

- Errors
 - Syntax error: Syntaxfel som gör att koden inte kan köra för att den inte är skriven rätt
 - Runtime error: Fel som upptäcks medan kod körs
 - Semantisk (logiskt) error: Fel i logiken som gör att programmet inte gör rätt.
 - Exceptions
 - raise – skapa egna exception
 - Hantera med try: except: finally:
 - Stack trace
 - Meddelanden vi får i terminalen när vi får errors
 - Läsas från botten och upp!
 - Debugging – vsc för att felsöka kod mer smärtfritt
 - Pytest
 - Unit testing är små automatiska tester av funktioner i koden
-

MENTIMETER - KURSSLUT
