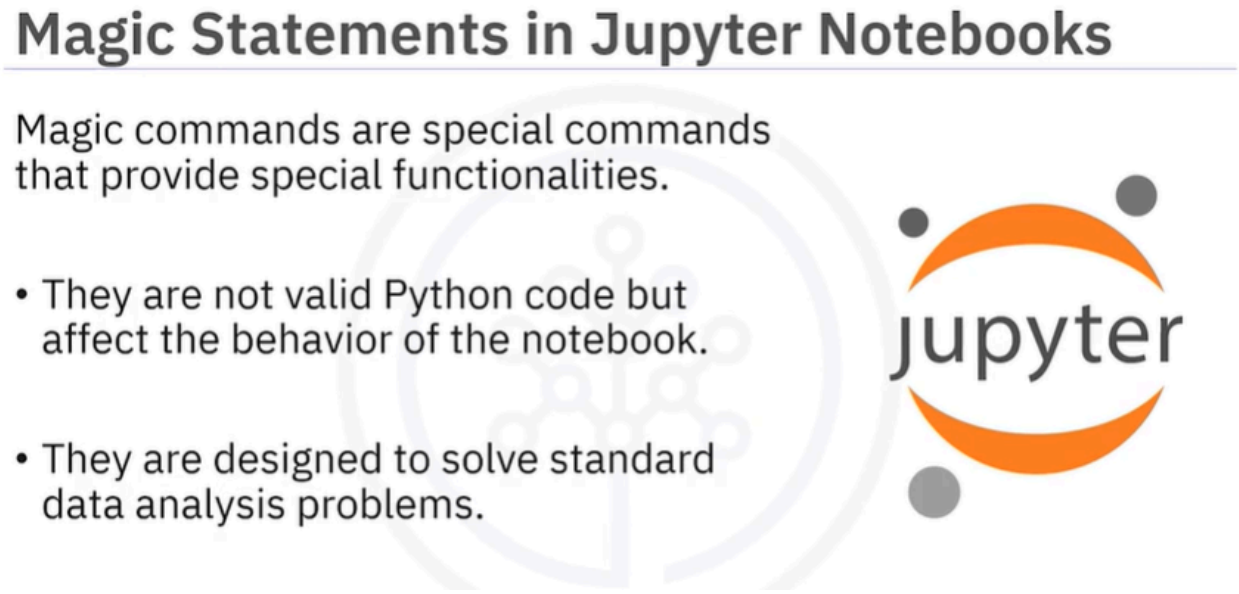
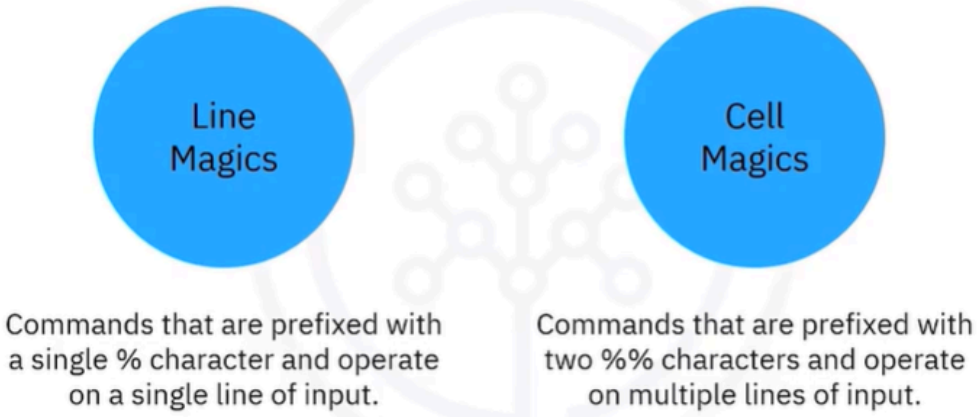
Accessing Databases with SQL Magic



- What are magic statements in Jupyter Notebooks? Magic commands or magic functions are special commands in Jupyter Notebooks that provide special functionalities. They are not valid Python code but affect the behavior of the notebook. They are designed to solve various common problems in standard data analysis.

Types of Cell Magics

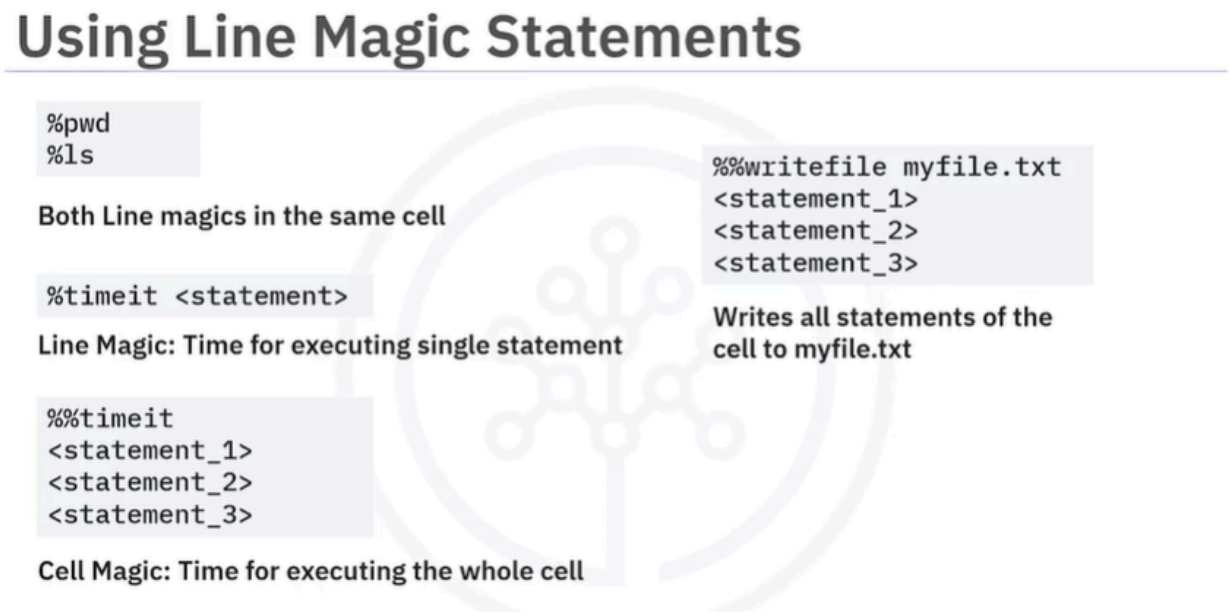


- There are two types of magic statements in Jupyter notebooks, line magics and cell magics. Line magics in Jupyter notebooks are commands that are prefixed with a single percentage character and operate on a single line of input. They are like command line calls in a terminal shell. On the other hand, cell magics are prefixed with two percentage characters and operate on multiple lines of input. They can even transform the entire cell or execute the cell in a different programming language.

Using Line Magic Statements

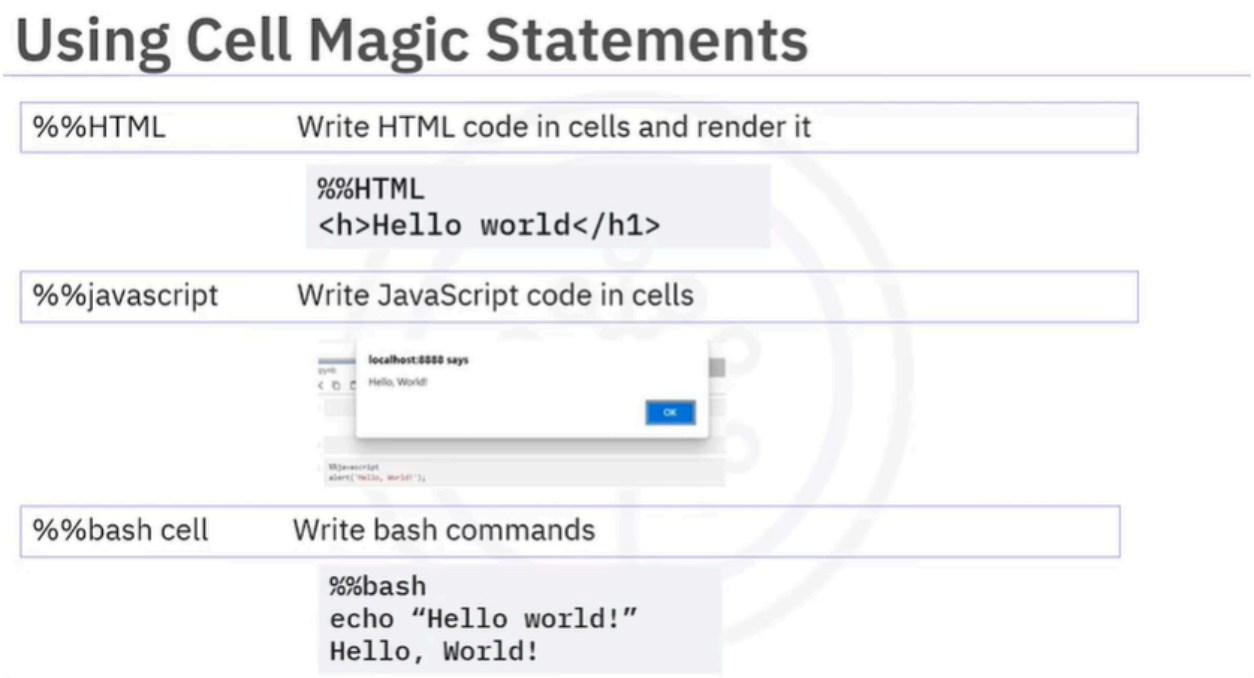
Line Magics	Uses
%pwd	prints the current working directory
%ls	lists all files in the current directory
%history	shows the command history
%reset	resets the namespace by removing all names defined by the user
%who	lists all variables in the namespace
%whos	provides more detailed information about all variables in the namespace
%matplotlib inline	makes matplotlib plots appear within the notebook
%timeit	times the execution of a single statement
%lsmagic	lists all available line magics

- Some popularly used Line Magics in Jupyter notebooks, %pwd. This command prints the current working directory. Percentage LS, this command lists all files in the current directory. Percentage history. This command shows the command history. Percentage reset. This command resets the namespace by removing all names defined by the user. Percentage who. This command lists all variables in the namespace. Percentage, whos. This command provides more detailed information about all variables in the namespace. Percentage matplotlib inline. This command makes matplotlib plots appear within the notebook. Percentage timeit, which times the execution of a single statement. To get a list of all available Line Magics you can use the percentage LS magic command.



- You can use multiple line magics in a single cell in a Jupyter Notebook. Each line magic should be on its own line. For example, percentage pwd, percentage is. This will print the current working directory and then list all files in the current directory. Remember, line magics operate on a single line of input. You can use as many as you need in a single cell if each is on its own line. Some line magic statements can also be used as cell magic statements, which would change the way they respond. For instance, percentage timeit

would generate response with the time required to execute a single statement. Whereas percentage percentage timeit times the execution of the entire cell. Another example where cell magics are uniquely applicable is writefile. Since percentage percentage writefile, my file.text writes the contents of the cell to myfile text.



- Cell magics are not just limited to Python, they allow you to run code in other languages too. Let's see some examples. Percentage, percentage HTML cell magic allows you to write HTML code in your cell, and it will be rendered accordingly. For example, the use of HTML code displays the text, Hello World as a heading, percentage percentage Javascript, or percentage percentage JS Cell magic allows you to write Javascript code in your cell. For example, the use of Javascript code generates a web page pop-up with the display message, Hello World. Percentage percentage bash cell magic allows you to write bash commands. For example, the use of bash script statement echo generates the text, Hello world, as will be received in a terminal.

Using SQL Magic

- Install ipython-sql by running the following statement:

```
!pip install --user ipython-sql
```

- Enable the SQL magic in Juypter notebook using this statement:

```
%load_ext sql
```

Using SQL Magic with SQLite Database

```
import sqlite3
conn = sqlite3.connect('HR.db')
```

```
%load_ext sql
```

```
%sql sqlite:///HR.db
```

```
%sql SELECT * FROM Employee
```

- Before running the queries however, you need to establish a connection between the SQL server and the SQL Magic module of the notebook.
- As an example, let us explore the use of SQL Magic with the SQL Lite3 database server. We first need to establish a connection to a database, say HR.db using SQL Lite3. Next, you load the external SQL module using %load_extsql.

  Now you need to establish a connection for the SQL magic module to the SQL server being accessed. This is done using the statement %sql sqlite:\\\HR.db.

  Now you can use SQL magic for executing any kind of queries on the table. A sample is shown here for displaying the contents of employee table in the HR database.

Let's talk about Magic Statements in Jupyter Notebooks.

Magic statements are special commands that you can use in Jupyter Notebooks to make your work easier. Think of them like shortcuts or tools in a toolbox that help you perform specific tasks without writing a lot of code. There are two main types of magic statements: **line magics** and **cell magics**. Line magics start with a single percentage sign (%) and work on one line of code, while cell magics start with two percentage signs (%%) and can handle multiple lines of code or even run code in different programming languages.

For example, if you want to see what files are in your current directory, you can use the line magic command `%ls`. If you want to write a block of HTML code and see it rendered, you can use the cell magic command `%%html`. This way, you can easily access different functionalities without getting bogged down in complex coding.

How can you use line magics in practice?

You can use line magics in Jupyter Notebooks to perform various tasks quickly. Here are a couple of practical examples:

1. **Check Current Directory**:
    - You can use the command `%pwd` to print the current working directory. This helps you know where you are in your file system.

```
%pwd
```

2. **List Files**:
    - To see all files in the current directory, you can use `%ls`. This is useful for checking what files are available for your work.

```
%ls
```

3. **View Command History**:
   - If you want to see the history of commands you've run, you can use `%history`. This can help you recall previous commands.

```
%history
```

4. **Reset Namespace**:
   - To clear all variables and reset your workspace, you can use `%reset`. This is helpful if you want to start fresh without any previous variables.

```
%reset
```

You can use multiple line magics in a single cell, just make sure each command is on its own line.

What would happen if you used a line magic incorrectly?

If you use a line magic command incorrectly in a Jupyter Notebook, you will typically encounter an error message. Here are a few common scenarios:

1. **Invalid Command**:
   - If you type a line magic that doesn't exist, you'll see an error message indicating that the command is not recognized. For example:

```
%invalid_command
```

   - This will result in an error like: `NameError: name 'invalid_command' is not defined`.

2. **Syntax Errors**:
    - If you don't follow the correct syntax for a line magic, such as missing the percentage sign, you will also get an error. For example:

```
pwd  # Missing the %
```

    - This will result in an error indicating that `pwd` is not defined.
3. **Incorrect Usage**:
    - If you try to use a line magic in a way that it doesn't support, you may get an error. For example, using a line magic that requires a specific argument without providing it:

```
%timeit  # Missing the statement to time
```

    - This will result in an error indicating that a statement is required.

In general, Jupyter Notebooks will provide feedback on what went wrong, allowing you to correct your command.

How can you correct an error in line magic?

To correct an error in line magic commands in Jupyter Notebooks, you can follow these steps:

1. **Read the Error Message**:
    - Carefully check the error message provided by Jupyter. It often gives clues about what went wrong, such as a typo or missing arguments.
2. **Check Command Syntax**:
    - Ensure that you are using the correct syntax for the line magic. For example, make sure you include the percentage sign (%) at the beginning of the command.

3. **Verify Command Existence**:
    - Confirm that the line magic command you are using is valid. You can list all available line magics by using the command `%lsmagic`. This will show you which commands you can use.
4. **Provide Required Arguments**:
    - If the line magic requires specific arguments, make sure to include them. For example, if using `%timeit`, you need to specify the statement you want to time:

```
%timeit sum(range(1000))
```

5. **Re-run the Corrected Command**:
    - After making the necessary corrections, re-run the command to see if it executes successfully.

By following these steps, you can effectively troubleshoot and correct errors in your line magic commands.