Coding Data Science

RONNY FAHRUDIN

OUTLINE

- The essential databases and SQL
- Roles data science with SQL
- SQL Component
- SQL Environments
- SQL Fundamentals & Practice

- The essential python
- Python Fundamentals
- Python Data Types
- Python Feature
- Practice

- Shell Script Fundamentals
- Basic Unix Command
- Feature in Shell Script
- Practice

DATABASES & SQL

What is Database?
What is SQL?

- Database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system **(DBMS)**.
- SQL is structured query language

What is Database?
What is SQL?

- Database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system **(DBMS)**.

- SQL is structured query language for writing and querying data, like accessed, managed, modified, updated, controlled, organized.

Why we use database?

- Databases can store very large numbers of records efficiently
- Very easy and quick to find information
- Easy to add new data and to edit or delete old data
- Data can be searched easily
- Data can be sorted easily
- Data can be imported into other applications
- More than one person can access the same database at the same time – multi access
- Security may better than in paper files

Databases in Data Science Role

Data Scientist

Use SQL to load data into their predictive models

Data Analyst

Use SQL to query tables of data and derive insights from it

Data Engineer

Will use SQL to ensure that everybody in their organization has access to the data they need.

SQL TOOLS

PostgreSQL

MySQL
Developement Services

MariaDB

talend

HS sql

Microsoft SQL Server

Microsoft Access

HUE

FUNDAMENTALS SQL SYNTAX

DDL (Data Definition Language)

This component to define the structure or schema of the database.

Standard command: CREATE TABLE table_name: create table
ALTER TABLE table_name : add/rmove column
DROP TABLE table_name: remove table

Example:

```
create table persons
(ID int not null, name varchar(255)
not null, age int);
```

```
DROP TABLE persons ;
```

https://www.w3schools.com/sql/trysql.asp?filename=trysql_select_all

FUNDAMENTALS SQL SYNTAX

DDL (Data Manipulation Language)

This component to manipulate data by the table.

Standard command:

- SELECT: select rows
- INSERT: insert rows
- UPDATE: change rows
- DELETE: remove rows

Example:

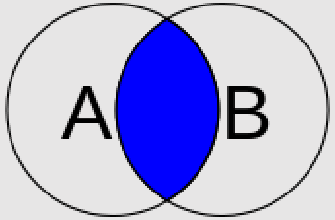
SELECT * from customers;

INSERT INTO customers(CustomerID, Country)
VALUES (30000,’Indonesia’);

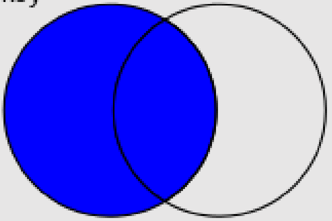
UPDATE customers SET Country ='Republic Indonesia'

DELETE FROM customers Where CustomerID = 12000;

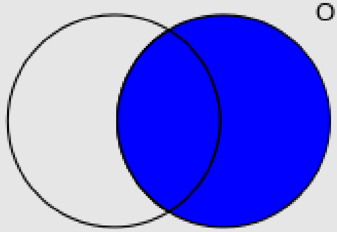
https://www.w3schools.com/sql/trysql.asp?filename=trysql_select_all



```sql
SELECT <fields>
FROM TableA  A
INNER JOIN TableB  B
ON A.key = B.key
```

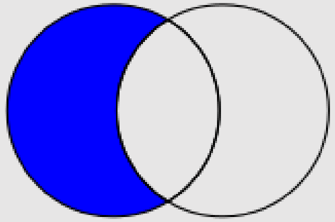


```sql
SELECT <fields>
FROM TableA  A
LEFT JOIN TableB  B
ON A.key = B.key
```

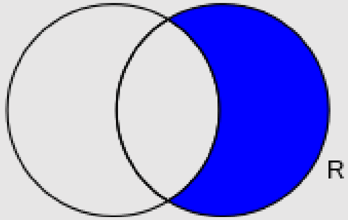


```sql
SELECT <fields>
FROM TableA  A
RIGHT JOIN TableB  B
ON A.key = B.key
```

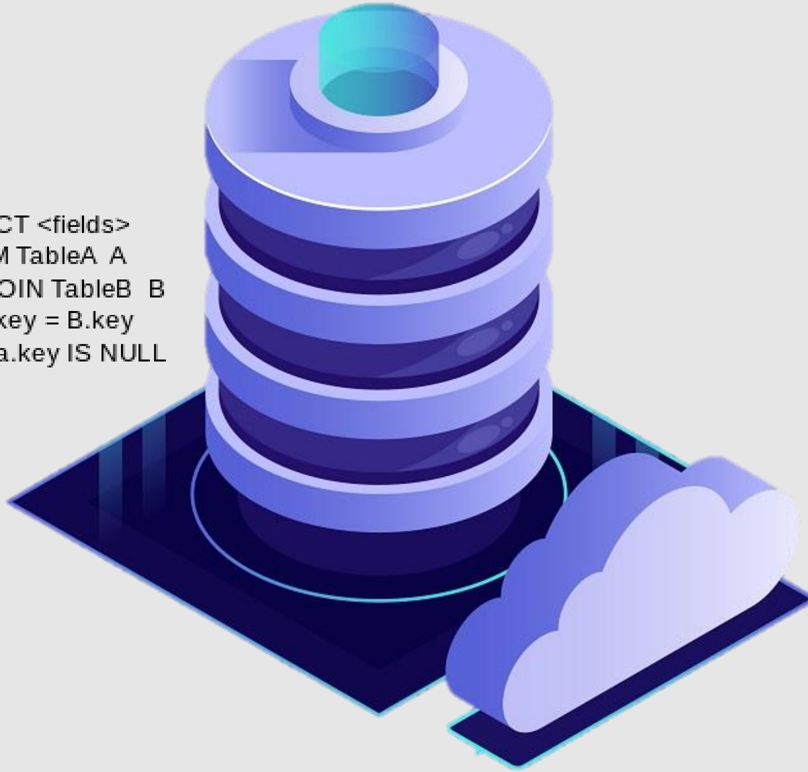
SQL JOINS

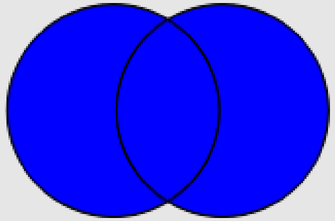


```sql
SELECT <fields>
FROM TableA  A
LEFT JOIN TableB  B
ON A.key = B.key
WHERE B.key IS NULL
```

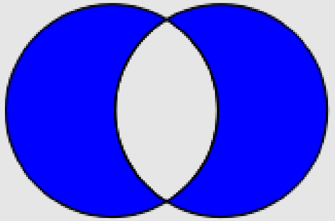


```sql
SELECT <fields>
FROM TableA  A
RIGHT JOIN TableB  B
ON A.key = B.key
WHERE a.key IS NULL
```





```sql
SELECT <fields>
FROM TableA  A
FULL OUTER JOIN TableB  B
ON A.key = B.key
```

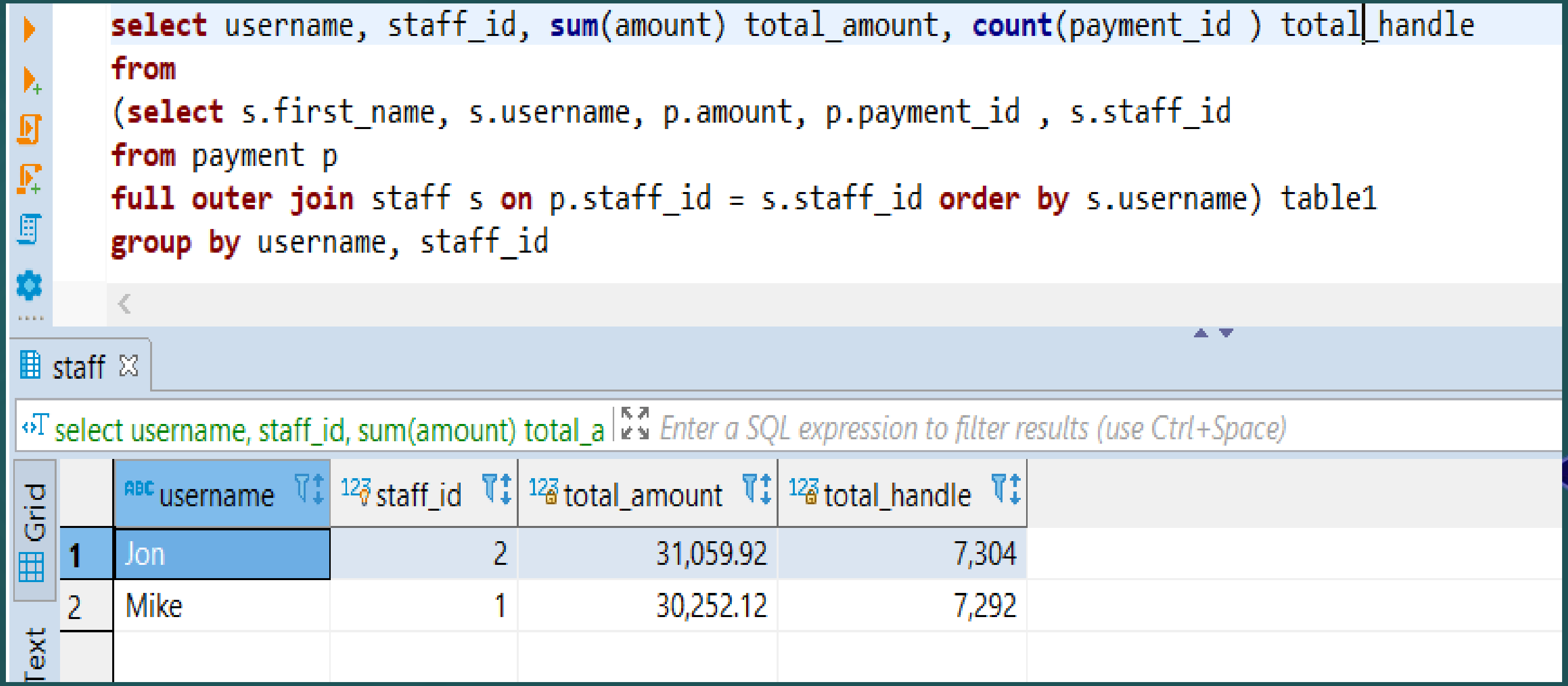


```sql
SELECT <fields>
FROM TableA  A
FULL OUTER JOIN TableB  B
ON A.key = B.key
WHERE A.key IS NULL
OR B.key iIS NULL
```

Practicum with study case

- Question 1:

We have a dataset dvdrentals. We have two staff members with staff IDs 1 and 2. We want to give a bonus to the staff member that handled the most payments. How many payments did each staff member handle? And how much was the total amount processed by each staff member?



```
select username, staff_id, sum(amount) total_amount, count(payment_id ) total_handle
from
(select s.first_name, s.username, p.amount, p.payment_id , s.staff_id
from payment p
full outer join staff s on p.staff_id = s.staff_id order by s.username) table1
group by username, staff_id
```

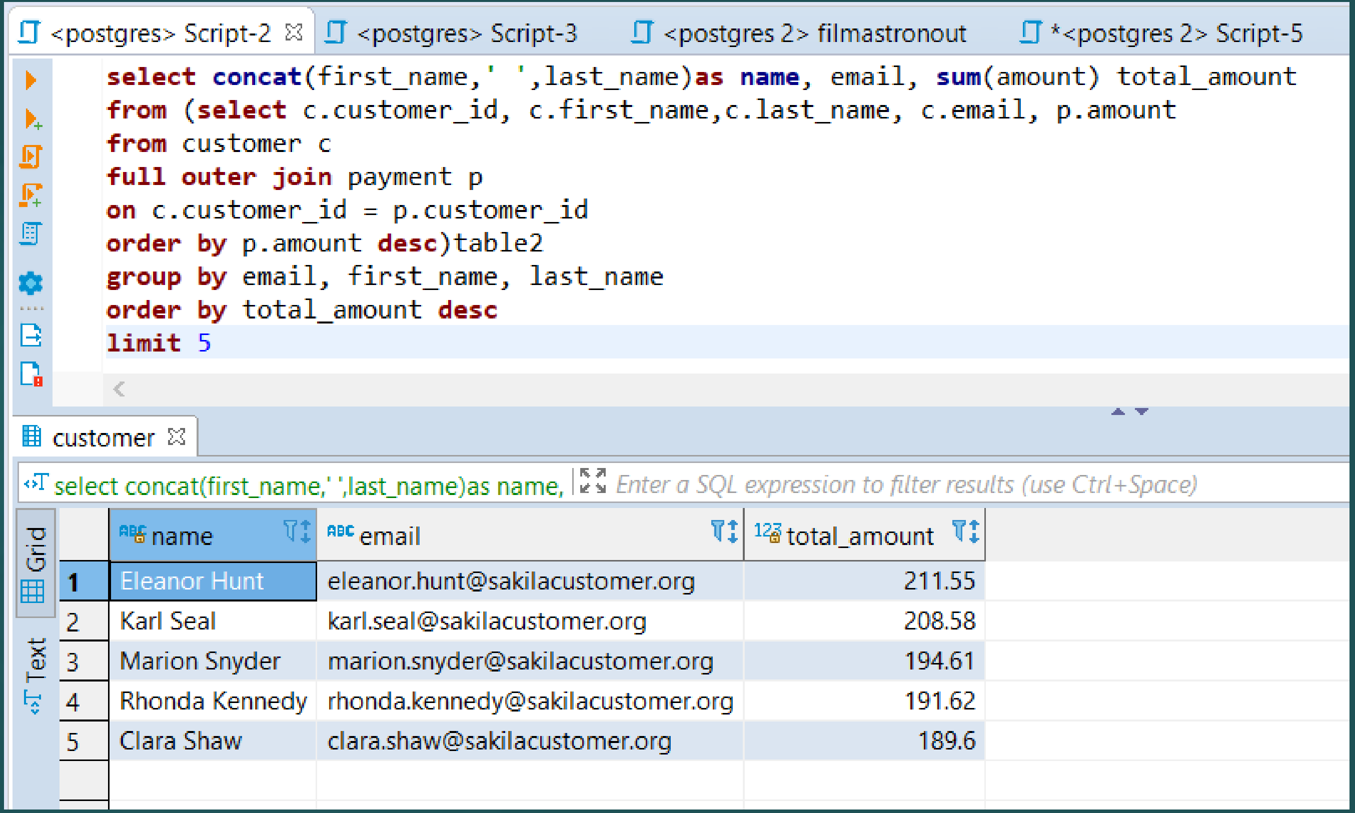
staff
select username, staff_id, sum(amount) total_a
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text

	username	staff_id	total_amount	total_handle
1	Jon	2	31,059.92	7,304
2	Mike	1	30,252.12	7,292

Practicum with study case

- Question 2:

We want to send coupons to the 5 customers who have spent the most amount of money. Get the customer name, email and their spent amount!



```
select concat(first_name,' ',last_name)as name, email, sum(amount) total_amount
from (select c.customer_id, c.first_name,c.last_name, c.email, p.amount
from customer c
full outer join payment p
on c.customer_id = p.customer_id
order by p.amount desc)table2
group by email, first_name, last_name
order by total_amount desc
limit 5
```

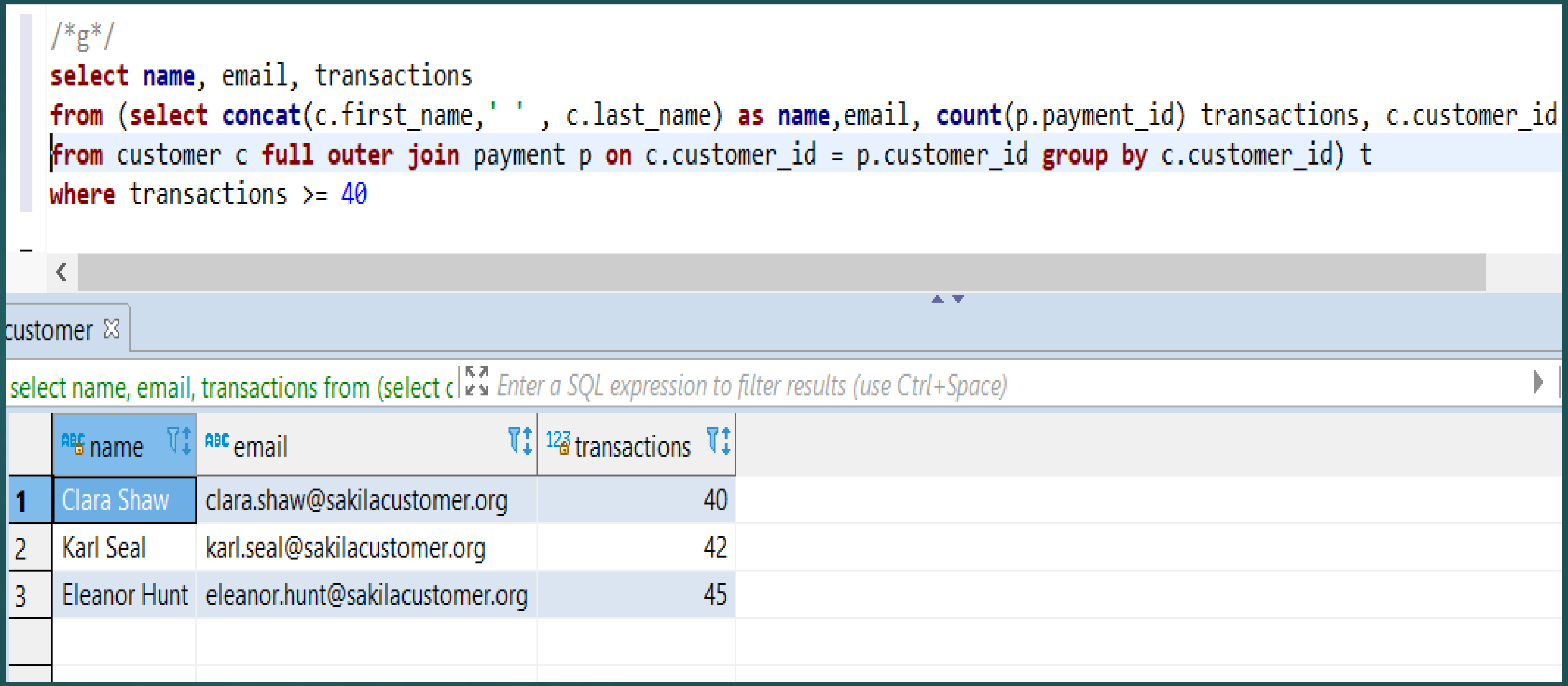
select concat(first_name,' ',last_name)as name, | Enter a SQL expression to filter results (use Ctrl+Space)

	name	email	total_amount
1	Eleanor Hunt	eleanor.hunt@sakilacustomer.org	211.55
2	Karl Seal	karl.seal@sakilacustomer.org	208.58
3	Marion Snyder	marion.snyder@sakilacustomer.org	194.61
4	Rhonda Kennedy	rhonda.kennedy@sakilacustomer.org	191.62
5	Clara Shaw	clara.shaw@sakilacustomer.org	189.6

Practicum with study case

- Question 2:

a. We want to know what customers are eligible for our platinum credit card. The requirements are that the customer has at least a total of 40 transaction payments. Get the customer name, email who are eligible for the credit card!



```sql
/*g*/
select name, email, transactions
from (select concat(c.first_name,' ' , c.last_name) as name,email, count(p.payment_id) transactions, c.customer_id
from customer c full outer join payment p on c.customer_id = p.customer_id group by c.customer_id) t
where transactions >= 40
```

customer
select name, email, transactions from (select c
Enter a SQL expression to filter results (use Ctrl+Space)

	name	email	transactions
1	Clara Shaw	clara.shaw@sakilacustomer.org	40
2	Karl Seal	karl.seal@sakilacustomer.org	42
3	Eleanor Hunt	eleanor.hunt@sakilacustomer.org	45

PYTHON

```
rror_mod         "MIRROR_X":
irror_mod.use_x = True
irror_mod.use_y = False
irror_mod.use_z = False
_operation == "MIRROR_Y":
irror_mod.use_x = False
irror_mod.use_y = True
irror_mod.use_z = False
_operation == "MIRROR_Z":
irror_mod.use_x = False
irror_mod.use_y = False
irror_mod.use_z = True

election at the end -add
_ob.select= 1
er_ob.select=1
ntext.scene.objects.activ
"Selected" + str(modifier
irror_ob.select = 0
bpy.context.selected_obj
ata.objects[one.name].sel

int("please select exactl

OPERATOR CLASSES ----

pes.Operator):
X mirror to the selecte
ject.mirror_mirror_x"
```

What is Python?

- Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Created by Guido van Rossum and first released in 1991.

Guido van Rossum

Benefit of use Python

- Can be used on a server to create web applications
- Can be used alongside software to create workflows.
- Can connect to database systems. It can also read and modify files.
- Can be used to handle big data and perform complex mathematics.
- Can be used for rapid prototyping, or for production-ready software development.
- Easy understanding because the language python like human language

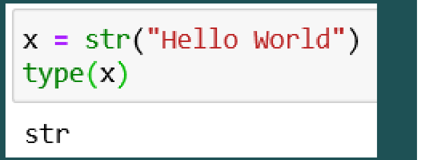
PYTHON DATA TYPES

Category	Type
Text Type	String/ str
Numeric Types	Integer, float, complex
Sequences Types	List, Tuple, Range
Mapping Type	dictionary
Set Types	Set, frozenset
Boolean Types	True/False, Yes/No
Binary Types	Bytes, bytearray, memoryview

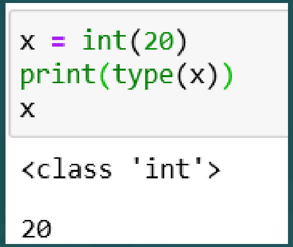
CHARACTERISTICS DATA TYPES

Type	Mutable	Indexing/Slicing
LIST	YES	YES
STRING	NO	YES
TUPLE	YES	YES
DICTIONARY	YES	NO
Set	YES	YES

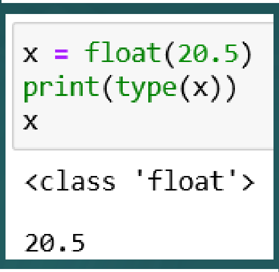
PYTHON DATA TYPES



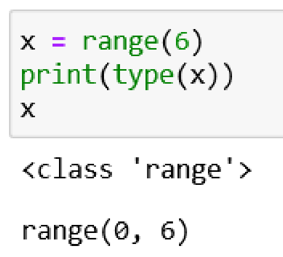
```
x = str("Hello World")
type(x)

str
```



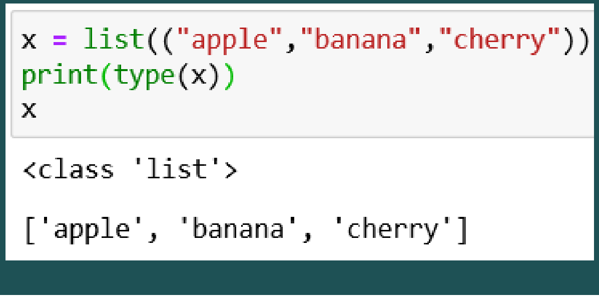
```
x = int(20)
print(type(x))
x

<class 'int'>

20
```



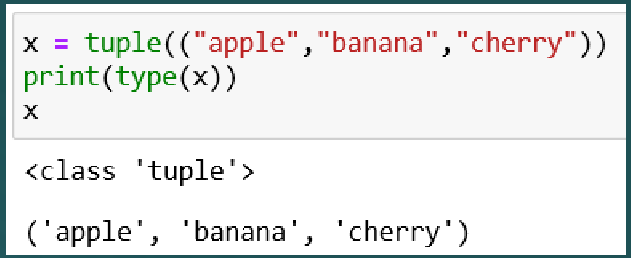
```
x = float(20.5)
print(type(x))
x

<class 'float'>

20.5
```



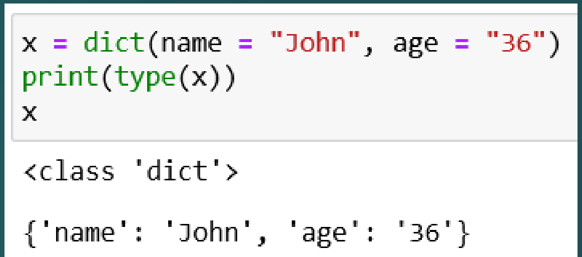
```
x = range(6)
print(type(x))
x

<class 'range'>

range(0, 6)
```



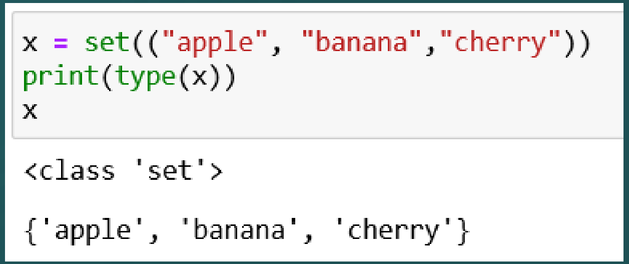
```
x = list(("apple","banana","cherry"))
print(type(x))
x

<class 'list'>

['apple', 'banana', 'cherry']
```



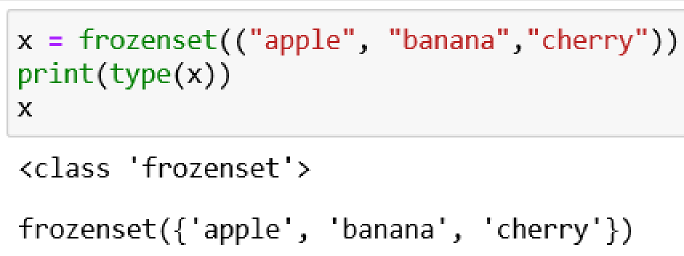
```
x = tuple(("apple","banana","cherry"))
print(type(x))
x

<class 'tuple'>

('apple', 'banana', 'cherry')
```



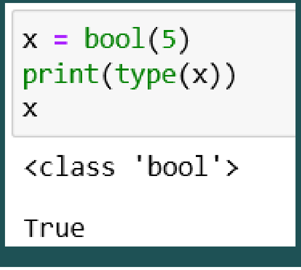
```
x = dict(name = "John", age = "36")
print(type(x))
x

<class 'dict'>

{'name': 'John', 'age': '36'}
```



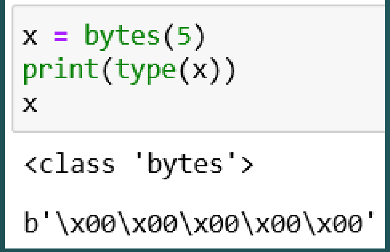
```
x = set(("apple", "banana","cherry"))
print(type(x))
x

<class 'set'>

{'apple', 'banana', 'cherry'}
```



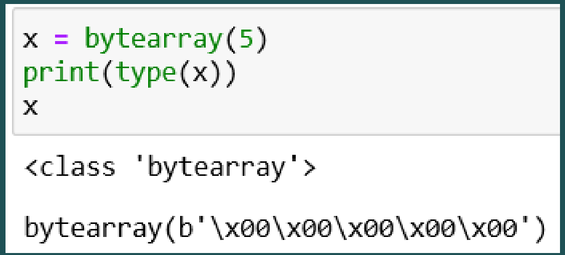
```
x = frozenset(("apple", "banana","cherry"))
print(type(x))
x

<class 'frozenset'>

frozenset({'apple', 'banana', 'cherry'})
```



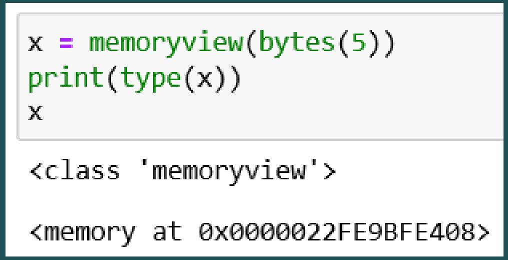
```
x = bool(5)
print(type(x))
x

<class 'bool'>

True
```



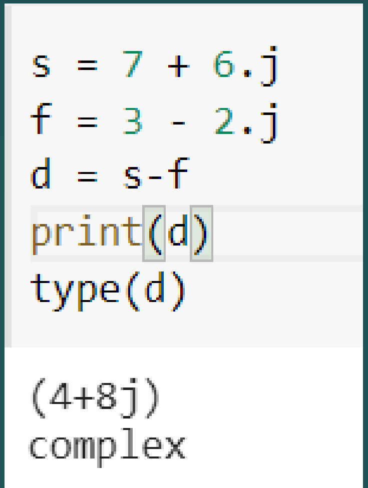
```
x = bytes(5)
print(type(x))
x

<class 'bytes'>

b'\x00\x00\x00\x00\x00'
```



```
x = bytearray(5)
print(type(x))
x

<class 'bytearray'>

bytearray(b'\x00\x00\x00\x00\x00')
```



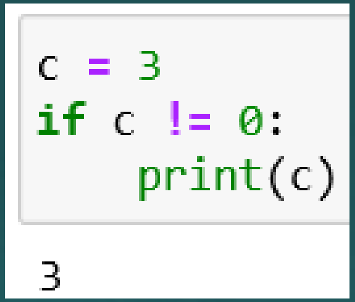
```
x = memoryview(bytes(5))
print(type(x))
x

<class 'memoryview'>

<memory at 0x0000022FE9BFE408>
```



```
s = 7 + 6.j
f = 3 - 2.j
d = s-f
print(d)
type(d)

(4+8j)
complex
```

Python Conditionals

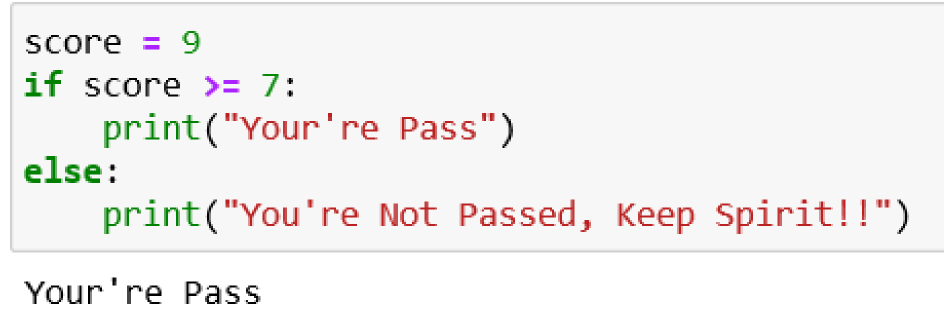
If



```
c = 3
if c != 0:
    print(c)
```

```
3
```

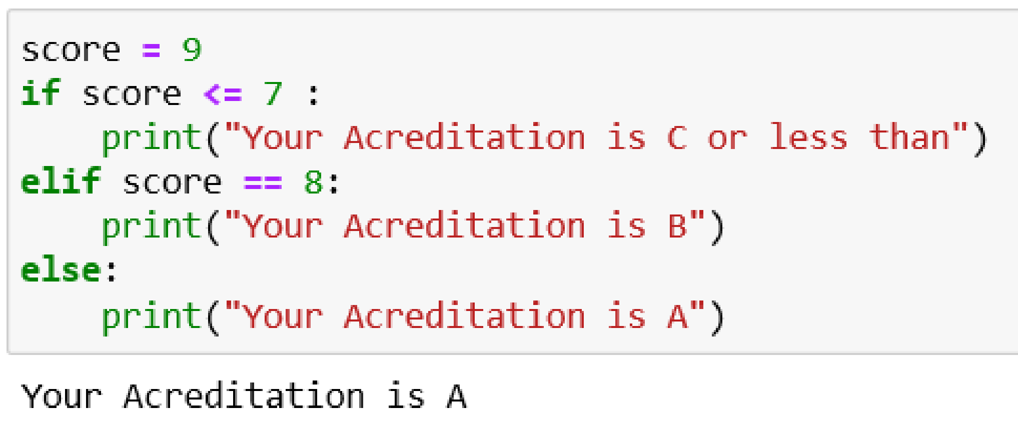
If else



```
score = 9
if score >= 7:
    print("Your're Pass")
else:
    print("You're Not Passed, Keep Spirit!!")
```

```
Your're Pass
```

If-elif-else

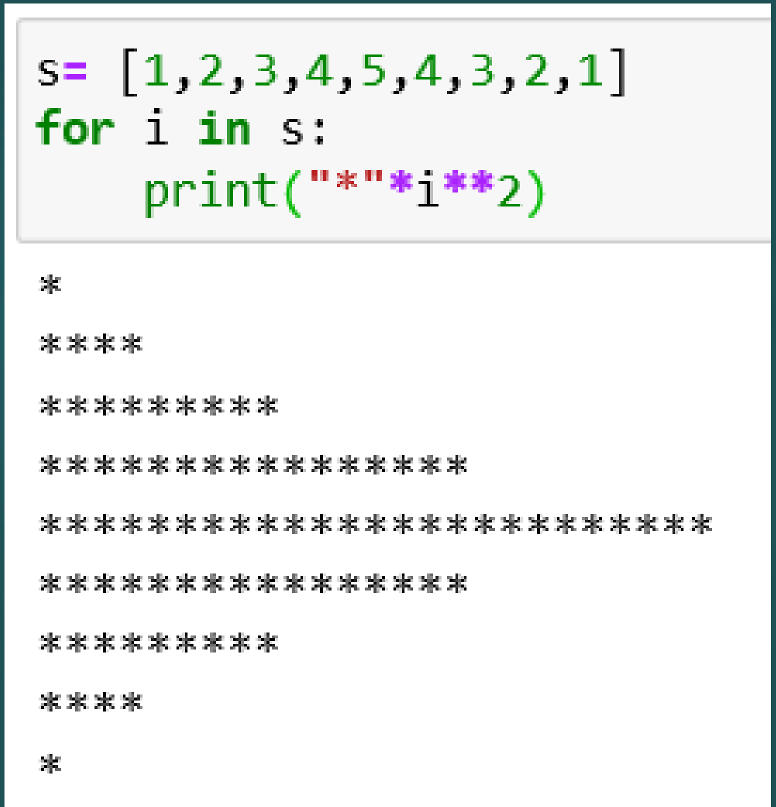


```
score = 9
if score <= 7 :
    print("Your Acreditation is C or less than")
elif score == 8:
    print("Your Acreditation is B")
else:
    print("Your Acreditation is A")
```

```
Your Acreditation is A
```

Python Loops

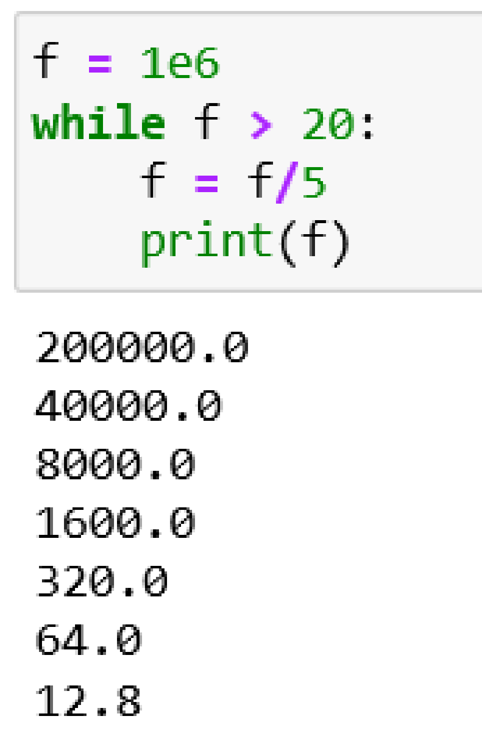
For



```
s= [1,2,3,4,5,4,3,2,1]
for i in s:
    print("*"*i**2)
```

```
*
****
*********
****************
*************************
****************
*********
****
*
```

While

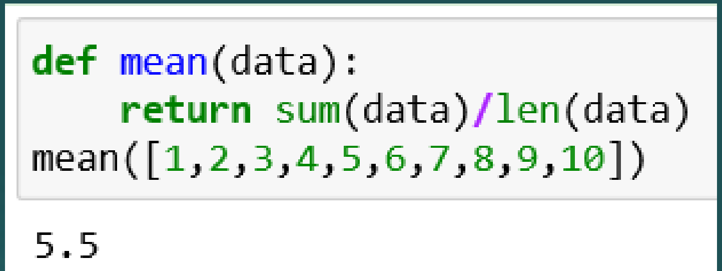


```
f = 1e6
while f > 20:
    f = f/5
    print(f)
```

```
200000.0
40000.0
8000.0
1600.0
320.0
64.0
12.8
```

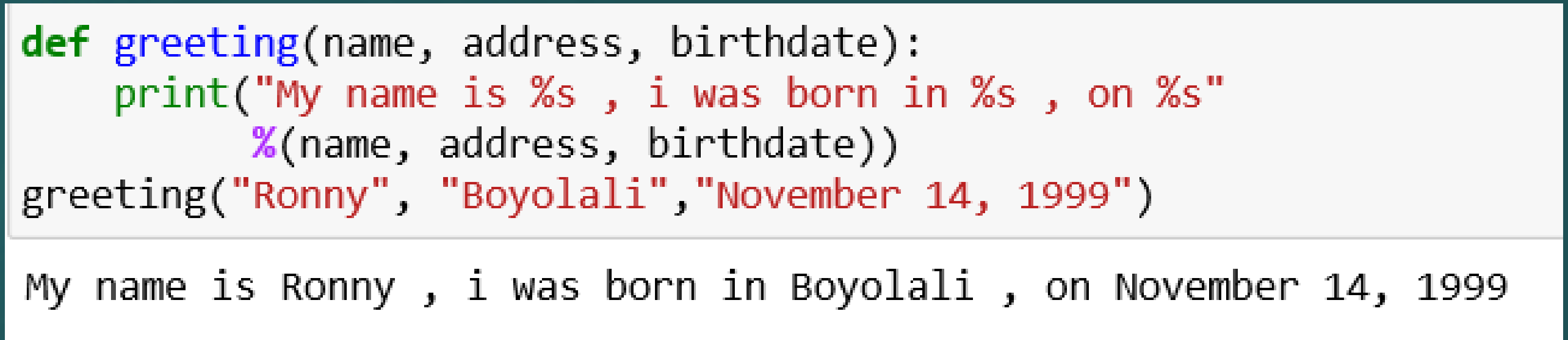
Python Functions

- To make function in python we use command “**def function_name():**”



```python
def mean(data):
    return sum(data)/len(data)
mean([1,2,3,4,5,6,7,8,9,10])
```

```
5.5
```



```python
def greeting(name, address, birthdate):
    print("My name is %s , i was born in %s , on %s"
          %(name, address, birthdate))
greeting("Ronny", "Boyolali","November 14, 1999")
```

```
My name is Ronny , i was born in Boyolali , on November 14, 1999
```

Where to learn python free

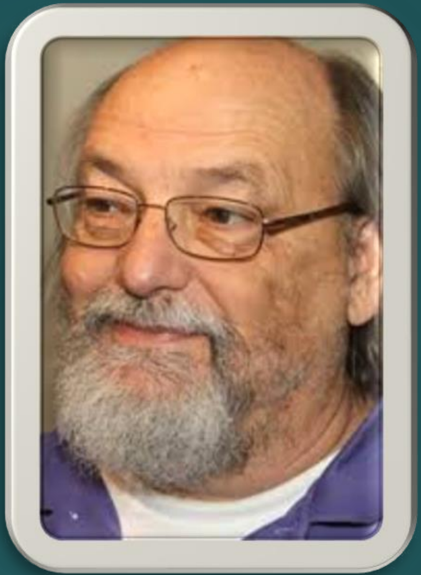
- https://colab.research.google.com/
- https://www.hackerrank.com/
- https://www.w3schools.com/python/
- https://www.codewars.com/
- https://codingbat.com/python
- etc

UNIX/LINUX

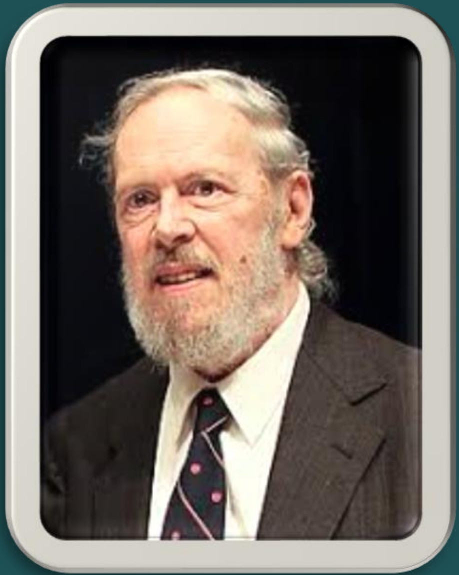
```
                 "MIRROR_X":
irror_mod.use_x = True
irror_mod.use_y = False
irror_mod.use_z = False
_operation == "MIRROR_Y":
irror_mod.use_x = False
irror_mod.use_y = True
irror_mod.use_z = False
_operation == "MIRROR_Z":
irror_mod.use_x = False
irror_mod.use_y = False
irror_mod.use_z = True

election at the end -add
_ob.select= 1
er_ob.select=1
ntext.scene.objects.activ
"Selected" + str(modifie
irror_ob.select = 0
 bpy.context.selected_obj
ata.objects[one.name].sel

int("please select exactl

  OPERATOR CLASSES ----

ypes.Operator):
 X mirror to the selecte
ject.mirror_mirror_x"
```

What is UNIX?

- Unix is a computer Operating System like Windows, iOS, MacOS. Unix capable of handling activities from multiple Users at the same time. It is developed by Ken Thompson and Dennis Ritchie at AT&T Bell Labs on around 1969.



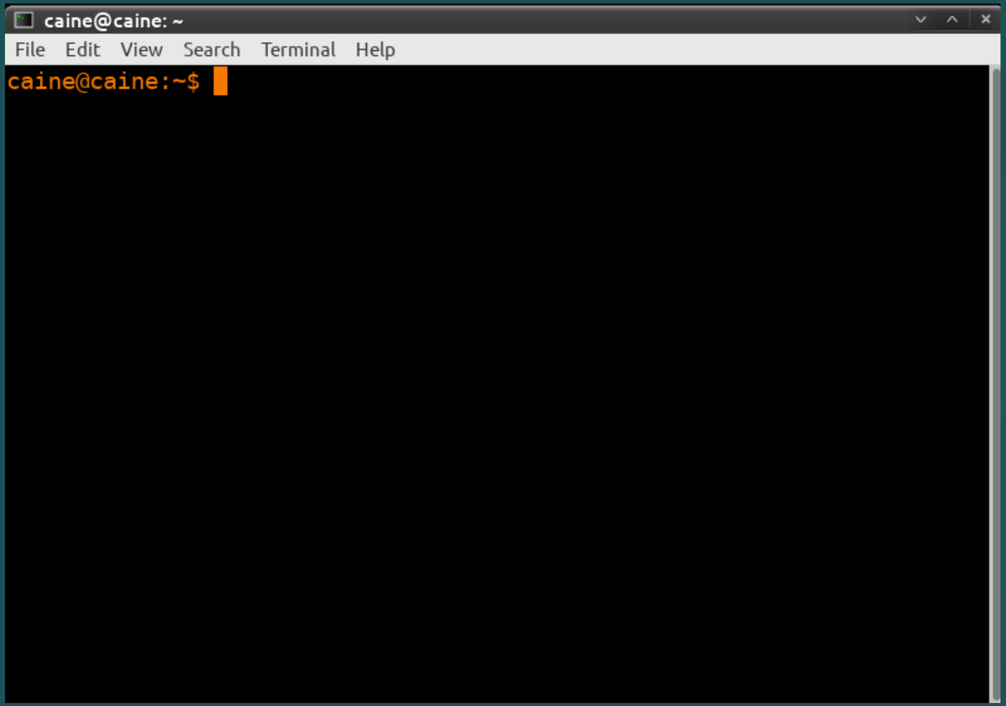
Ken Thompson



Dennis Ritchie

What is shell script?

- Shell script is a computer program designed to be run by the Unix/Linux like a command prompt in Windows. It is a command-line interpreter and typical operations performed by shell scripts include file manipulation, program execution, and printing text.



```
caine@caine:~$ 
```

Basic Command Shell Scripting

- To seeing list folder : “**ls**”
- To create file: “**touch file_neme.formatfile**”
- To see the content of file: “**cat file_name**”
- Editing content: “**vim file_name**”
- Save and quit from editing: “**:wq**”
- Copy file: “**cp name_file name_file_copy**”
- Rename file: “**mv file_neme renamed_file**”
- Delete file: “**rm name_file**”
- Make new directory: “**mkdir name_directory**”
- Copy directory: “**cp –r name_directory name_directory_copy**”
- Remove directory : “**rm –r name_directory**”
- Access folder : ”**cd name_folder**”

https://medium.com/@ronnyfahrudin/basic-tutorial-shell-scripting-on-unix-65a5051e2e8f

THANK YOU

Resource:

- https://medium.com/omarelgabrys-blog/database-introduction-part-1-4844fada1fb0
- https://www.w3schools.com/sql/
- https://www.w3schools.com/python/
- https://www.learnpython.org/en/Functions
- https://www.tutorialspoint.com/unix/shell_scripting.htm
- https://linux.die.net/abs-guide/why-shell.html
- https://www.tutorialspoint.com/python/index.htmhttps://www.geeksforgeeks.org/introduction-to-unix-system/
- https://en.wikipedia.org/wiki/Unix
- https://www.w3schools.com/python/python_datatypes.asp
- https://youtu.be/zWVV31NYi1U?list=PLS1QulWo1RIYmaxcEqw5JhK3b-6rgdWO_