environment variables (lesson 18)

* what are environment variables?

1. each user has its own environment.
2. each user can configure its own environment by setting preferences
3. These os configurations should be isolated from other user environments.

font	shell	editor	Browser
big	Bash	vim	Chrome
small	sh	Nano	Firefox

Where does OS store all these configurations?

→ environment variables.

→ pieces of info defined by key value pair.

uppercase ↖ SHELL = /bin/bash

variable name — variable value

→ available for the whole environment.

USER = yash

HOME = /home/yash.

→ user can change environment values

* list all environment variables.

```
printenv
```

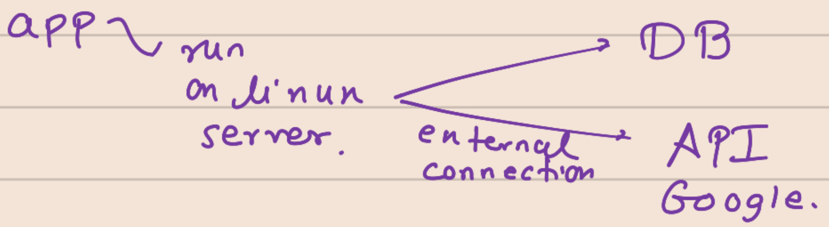
* print specific env variable.

```
printenv USER
echo $USER
```

* application env variable

- OS stores the info about the environment.
- we can create own environment variables

Use case: Sensitive data for application.



some authentication is needed
usr :
password :

How to tell the app the secret token to connect to the api ?

We cant write the value in the code ✗
not secure ✗

Set this data as env var on server

DB_PWD = secretpassword
DB_USER = mysqluser
↓
app

→ app can read those env var.

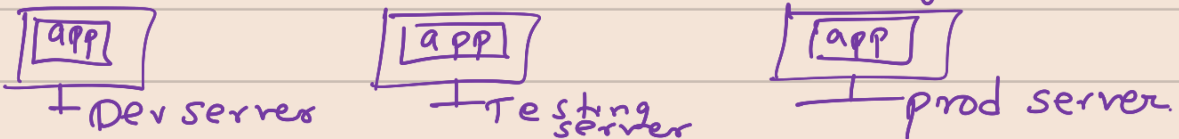
→ every programming language allows you to access environment variables

python example

```
import  mysql.connector
DB_USER=  os.getenv('DB_USER')
DB_PWD  =  os.getenv('DB_PWD)
mydb  =  mysql.connector.connect(
        host = "                    ",
        user = DB_USER,
        password = DB_PWD
)
```

→ it makes application more flexible!

app can run in diff server without changing code



↓ Dev DB   ↓ Testing DB   ↓ PROD DB.

each database has its own credentials.

→ no need to change application code

* Creating env variables

```
export DB_USERNAME = dbuser
export DB_PASSWORD = secretpwdvalue
export DB_NAME = mydb
printenv | grep DB
echo $DB_NAME
```

* Deleting env variable.

```
unset DB_NAME
```

* Persisting env variable.

when we close the terminal and open it again and search printenv | grep DB it is gone (user, pswd)

export → not permanent
goes away when session closes

To store permanently

```
vim .bashrc
```

press i (insert)
at last write

```
export DB_USERNAME = dbuser
export DB_PASSWORD = secret vl
export DB_NAME = mydb
:wq
source .bashrc
```

load the new env var into the current shell session

* persisting env variable system wide

.bashrc → user specific

```
vim /etc/environment
```

* add a custom command/program.

.bashrc file → `PATH = $PATH:/home/yash`