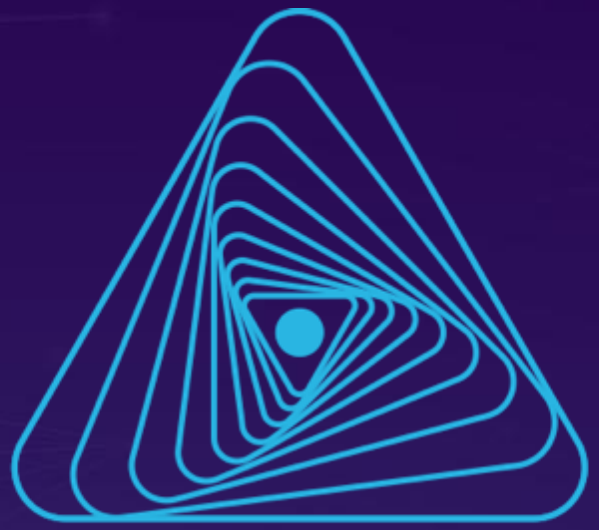


ARMADA

INNOVATION LABS

www.armadalabs.tech

Python Concepts

Python Programming Language

- General-purpose programming language.
- Used in machine learning, web development, desktop and mobile applications, and many more.
- Network and System Automation and Attack Payloads

Python 2.X	Python 3.X
No longer under active development, but supported by the Python community	Under active development
Better library support	Designed to be easier to learn
Default on Linux and Mac	Fixed major issues in 2.x
Supported by F5 BIG-IP, Cisco NX-OS and Arista	Not backwards-compatible

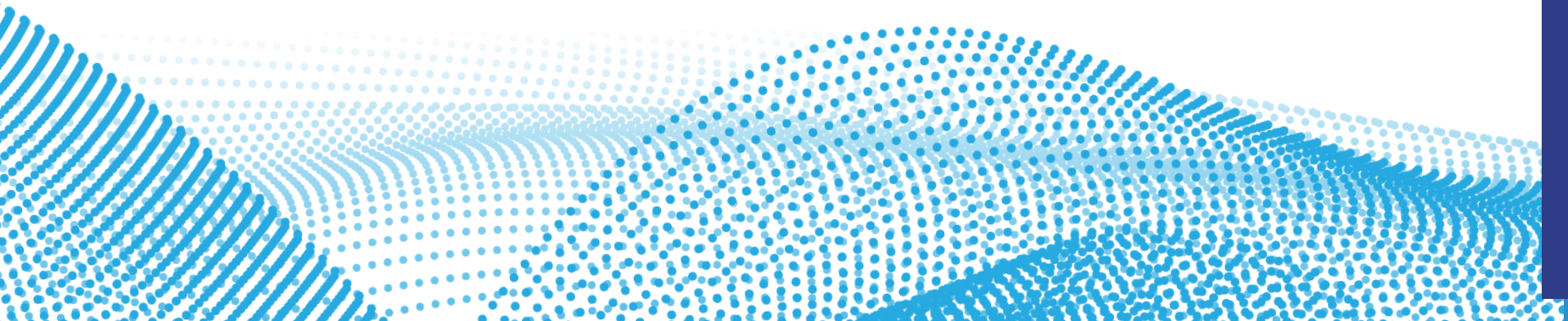
Executing Python

Python Code

- Using dynamic interpreter (shell)
- Writing Python Scripts

Open Python Shell

```
┌──(kali㉿kali)-[~]
└─$ python
Python 3.10.4 (main, Mar 24 2022, 13:07:27) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```



Executing Python

Python Scripts

- Script invoked by filename

Script locally written

```
#!/usr/bin/python

course = "Careers in Python – Web Hacking and Attacks"

print (course)
```

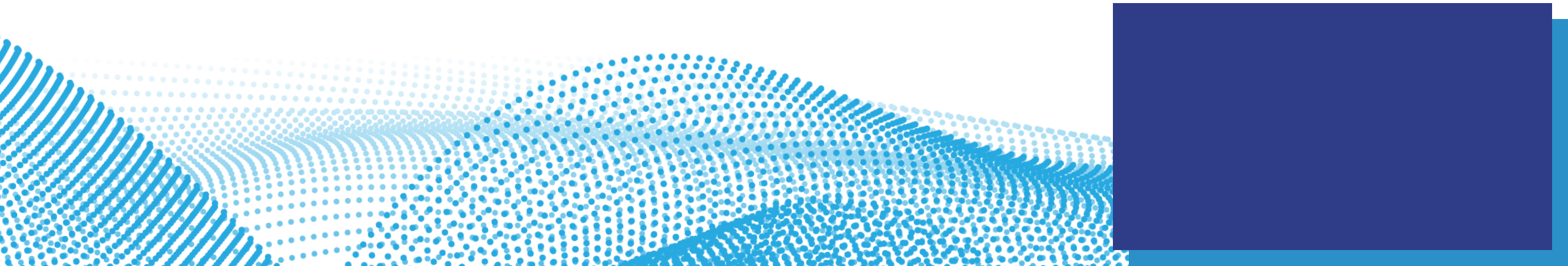
Script invoke by filename

```
┌──(kali㉿kali)-[~]
└─$ ./pysweb.py
Careers in Python – Web Hacking and Attacks

┌──(kali㉿kali)-[~]
└─$ python pysweb.py
Careers in Python – Web Hacking and Attacks
```

Executing Python

Command	Results
help()	Returns the python built-in documentation about the object.
dir()	Returns the attributes (and methods) of the object or module.
type()	Returns the type of the object.
exit()	Exit Python program



String

String Data Types

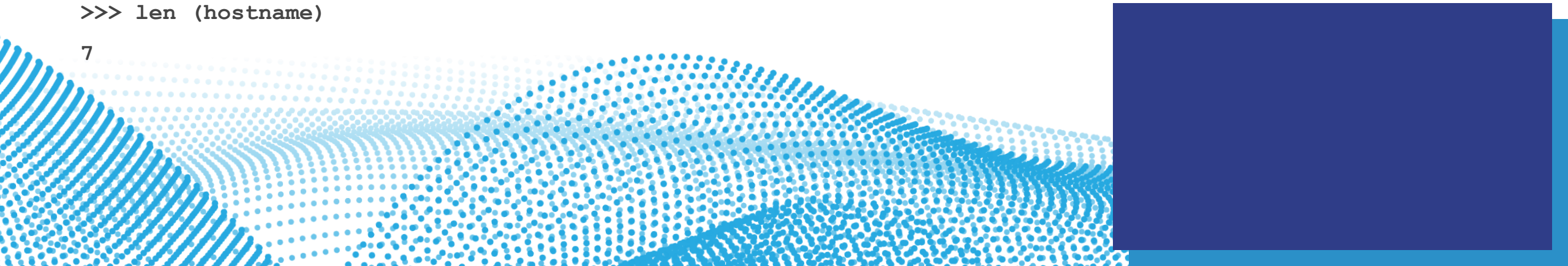
- Sequence of characters that surrounded by quotes

```
>>> hostname = "WebSrv1"
>>> mgmt = '10.0.0.1'
>>> hostname
'WebSrv1'
>>> print (hostname)
WebSrv1
>>> type (hostname)
<class 'str'>
>>>
>>> len (hostname)
7
```

Print values as string literal
Typing in the variable frrom interpreter

Print values as rendered string
Use print statement to print strings

Data tye is String



String

String Data Types

- Sequence of characters that surrounded by quotes

```
>>> hostname = "WebSrv1"
>>> mgmt = '10.0.0.1'
>>> hostname
'WebSrv1'
>>> print (hostname)
WebSrv1
>>> test[0] = "zero"
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'test' is not defined
>>> test = ("zero", "one", "two")
>>> print (test[0])
zero
```

Print values as string literal
Typing in the variable frrom interpreter

Print values as rendered string
Use print statement to print strings

Immutable individual can't be natively modified

String

Concatenating Strings

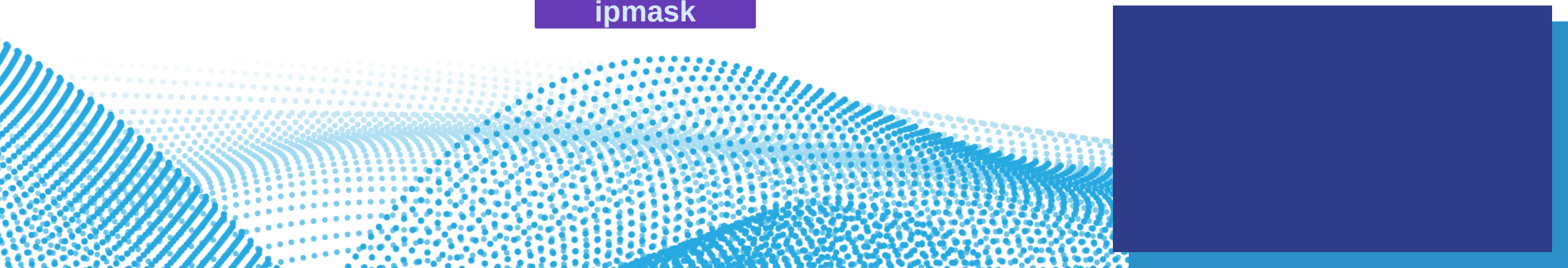
- combine two or more strings

```
>>> ipadd = "10.0.0.1"
>>> mask = '255.255.255.0'
>>>
>>> ipmask = ipadd + " " + mask
>>>
>>> print (ipmask)
10.0.0.1 255.255.255.0
```

Original Strings

Creating the string **ipmask** by adding combining variables with single space in between

New string **ipmask**



String

Built in Methods

```
>>> hostname = "websrv1"
>>> hostname.upper()
'WEBSRV1'
>>>
>>> macadd = "11:22:33:44:55:66"
>>> macadd.replace(":",".")
'11.22.33.44.55.66'
>>>
>>> ipadd = "10.0.0.101"
>>> ipadd.startswith('10')
True
>>> ipadd = "10.0.0.{}"
>>> ipadd.format("10")
'10.0.0.10'
>>>
>>> ipadd = "1.2.3.4"
>>> ipadd.split(".")
['1', '2', '3', '4']
```

returns string to all uppercase

":" replaced with "."

verified first octet of "ipadd"

formattable section where value 10 is added

showing list separated by dot(.) value in the string

Integers

Numeric Operators

- Math Basics – Addition, Subtraction, Multiplication, Division

```
>>> 7 * 8
56
>>> 89 - 6
83
>>> 283439284 + 93431
283532715
>>>
>>> 100 / 3
33.333333333333336
>>>
>>> round (100/3, 2)
33.33
```

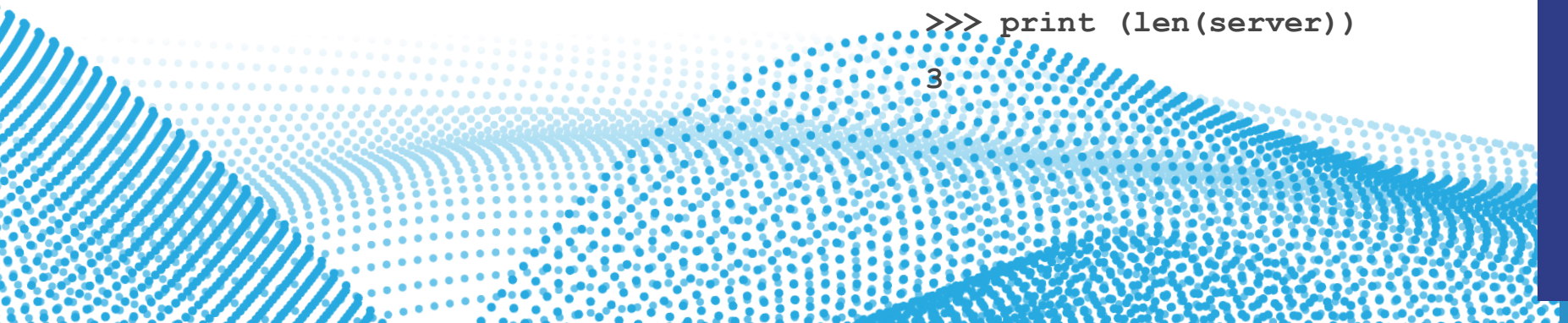
```
>>> num = 5
>>>
>>> num
5
>>>
>>> type(num)
<class 'int'>
```

List

Data in List / Array

- store multiple items in a single variable
- Separated by commas
- Uses Square Brackets
- Can have the same value
- Mutable

```
>>> server = ["srv1","srv2","srv3"]
>>> print (server)
['srv1', 'srv2', 'srv3']
>>>
>>> print (server[0])
srv1
>>> print (server[1])
srv2
>>> print (server[3])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: list index out of range
>>>
>>> print (len(server))
3
```

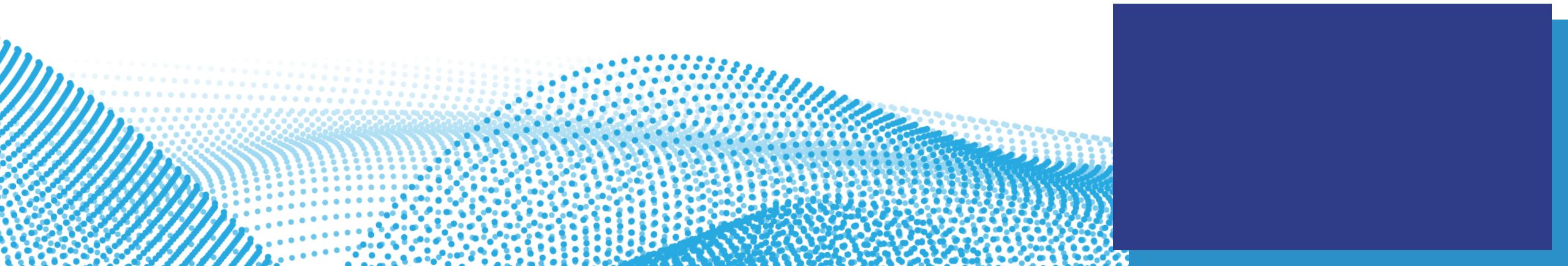


List vs Tuple

Data in Tuple

- store multiple items in a single variable
- Separated by commas
- Uses Open/Close Paranthesis
- Can have the same value
- Immutable

```
>>> server = ["srv1","srv2","srv3"]
>>> server
['srv1', 'srv2', 'srv3']
>>> server[0] = "srv0"
>>> server
['srv0', 'srv2', 'srv3’]
>>> web = ("tomcat", "nginx", "apache")
>>> web[0] = "iss"
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'tuple' object does not support item assignment
```



Dictionaries

Data in Dictionary

- store multiple items in a variable
- Key-Value pair
- Key-value uses colon
- Uses curly braces
- Pair Separated by commas

```
>>> server = ["srv1","srv2","srv3"]
>>>
>>> srvinfo = {"web1":"tomcat", "web2":"nginx", "web3":server}
>>>
>>> print (srvinfo['web1'])
tomcat
>>> print (srvinfo['web2'])
nginx
>>> print (srvinfo['web3'])
['srv1', 'srv2', 'srv3']
>>>
>>> print (srvinfo['web3'][2])
srv3
```

Boolean

Boolean Operators

- AND – All must match for output to be “True”.
- OR – Any must match for output to be “True”.
- NOT – Takes the inverse

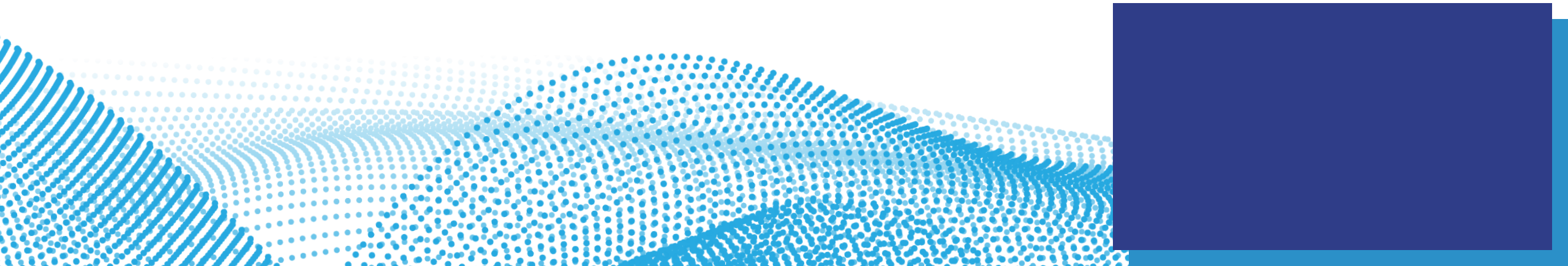
AND	Results
True and False	False
True and True	True
False and False	False
OR	**Results**
True and False	True
True and True	True
False and False	False
NOT	**Results**
not True	False
not False	True

Conditions

Operators

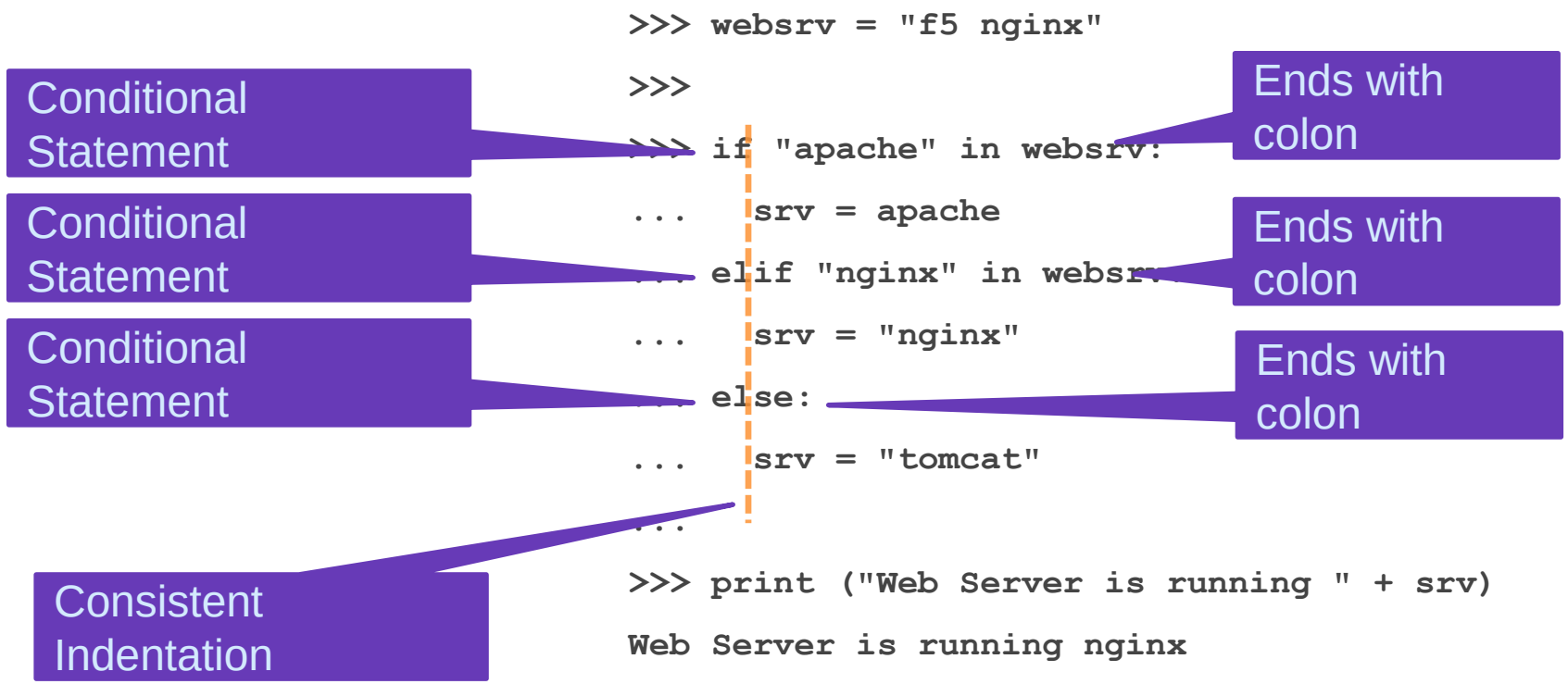
Type	Operators
Comparison	==,!=, >, <, >=, <=
Logical	and, or, not
Membership	in, not in
Identity	is, is not

```
>>> 89431 > 83111
True
>>> 87431 > 88715
False
>>> "nginx" != 'f5'
True
>>> "nginx" in "f5 nginx"
True
>>> "nginx" in "f5 big-ip"
False
>>> "nginx" not in "f5 big-ip"
True
```



Conditional Statement

If / elif / else Statement



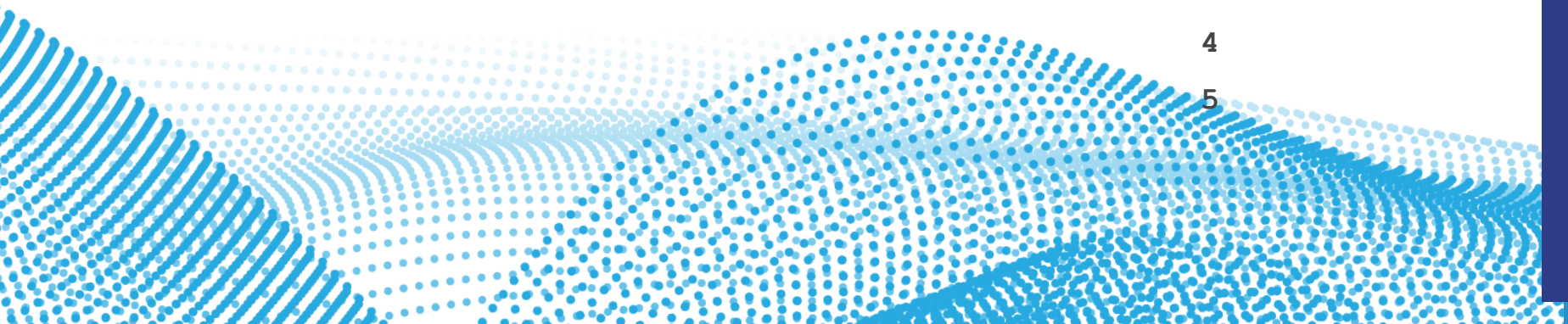

```
>>> websrv = "f5 nginx"
>>>
>>> if "apache" in websrv:
...     srv = apache
... elif "nginx" in websrv:
...     srv = "nginx"
... else:
...     srv = "tomcat"
...
>>> print ("Web Server is running " + srv)
Web Server is running nginx
```

Loops

For Loop

- Iterating over a sequence (most of the time numeric)
- Can be used for list, dictionary or string
- Print multiple values

```
>>> server = ["srv1","srv2","srv3"]
>>> for x in server:
...     print (x)
...
srv1
srv2
Srv3
>>>
>>> for x in range (2,6):
...     print (x)
...
2
3
4
5
```

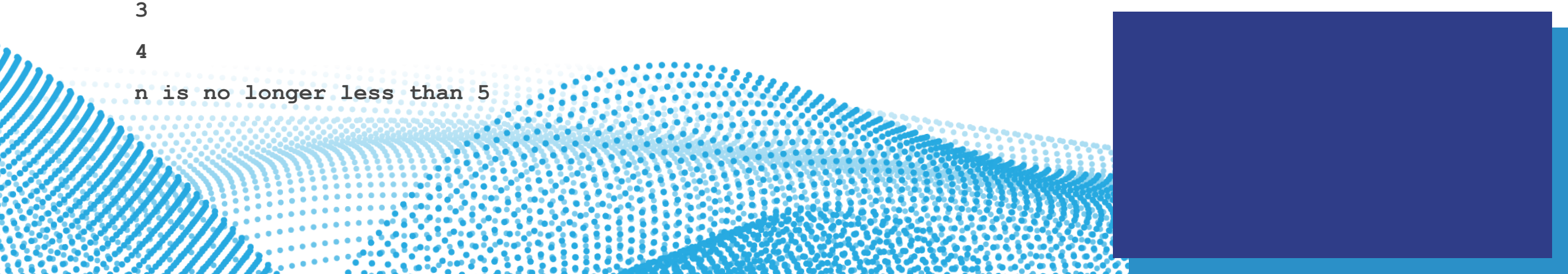


Loops

While Loop

- Requires relevant variables to be available

```
>>> n = 1
>>> while n < 5:
...     print (n)
...     n = n + 1
... else:
...     print ("n is no longer less than 5")
...
1
2
3
4
n is no longer less than 5
```



Input

Standard/Raw Input

- ask to input value before running specific code

```
#!/usr/bin/python


username = input ("Enter Username: ")
print ("\nYour username is " + username)


┌──(kali㉿kali)-[~]
└─$ python input.py
Enter Username: roborats

Your username is roborats
```

Input

Password Input

- ask to input value in a hidden format

```
#!/usr/bin/python
import getpass


username = input ("Enter Username: ")
password = getpass.getpass ("Enter Password: ")
print ("Your username is " + username + "\nYour Password is " + password)
```

```
┌──(kali㉿kali)-[~]
└─$ python input.py
Enter Username: roborats
Enter Password:
Your username is roborats
Your Password is supermouse
```

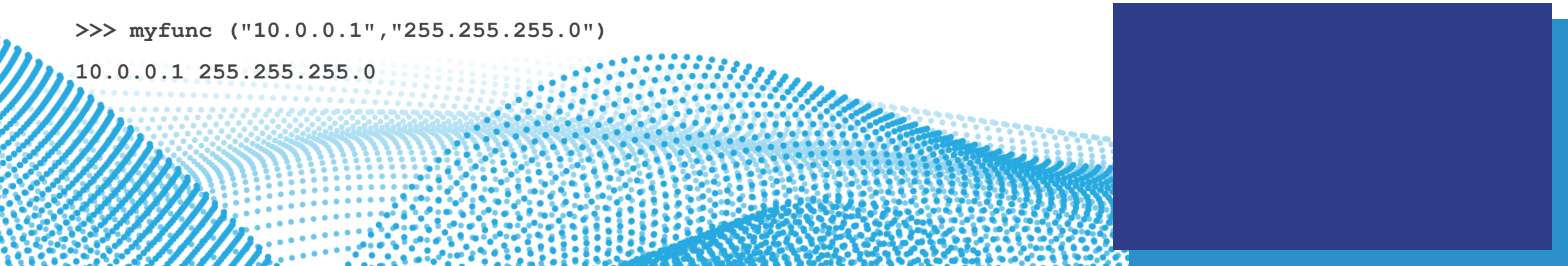
Functions

Functions

- Block of code only runs when it is called
- You can pass data (aka parameters) into function and it returns data as result

```
>>> def myfunc():
...   print ("Test from a function")
...
>>> myfunc()
Test from a function

>>> def myfunc(ipadd,mask):
...   print (ipadd + " " + mask)
...
>>> myfunc ("10.0.0.1","255.255.255.0")
10.0.0.1 255.255.255.0
```



Functions

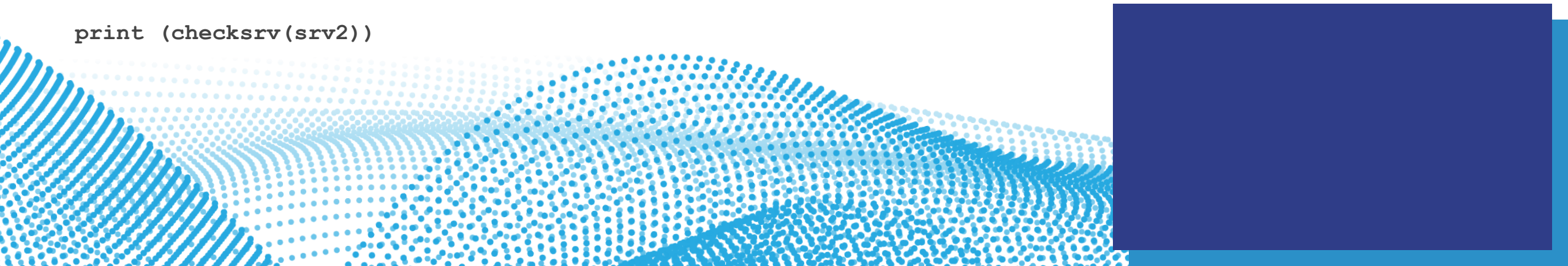
```
#!/usr/bin/python


srv1 = "apache"
srv2 = "nginx"


def checksrv(srv):
  if srv == srv1:
    return "The Web Server is " +  srv1
  elif srv == srv2:
    return "The Web Server is " +  srv2
  else:
    return "Its Tomcat"


print (checksrv(srv2))
```

```
┌──(kali㉿kali)-[~]
└─$ python mysrvs.py
The Web Server is nginx
```



Modules

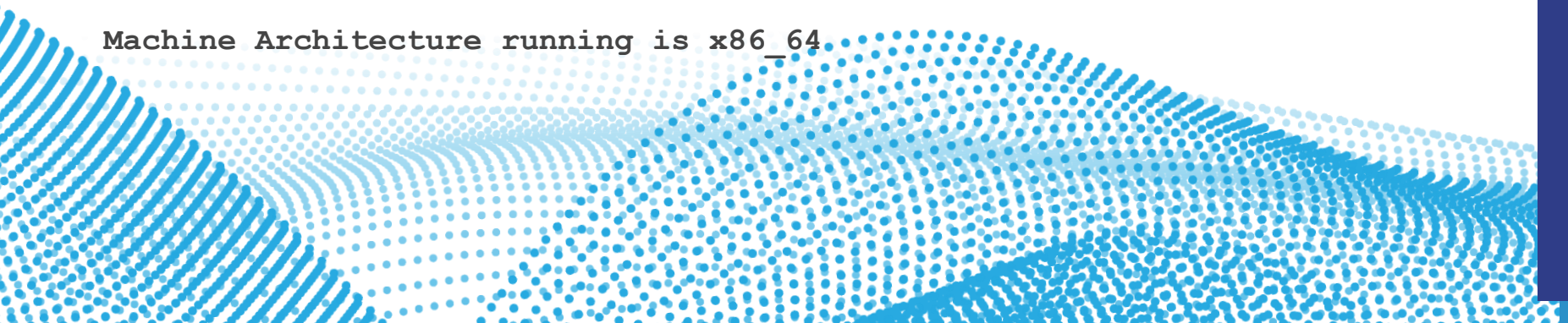
Python Modules

- Code library
- Built-in modules – import built-in modules per script requirements
- Custom modules – a separate file contains functions or variables of all types

```
#!/usr/bin/python
import platform


x = platform.machine()
print ("Machine Architecture running is " + x)
```

```
┌──(kali㉿kali)-[~]
└─$ python builtin.py
Machine Architecture running is x86_64
```



Modules

sys and os module

- provide numerous tools to deal with filenames, paths, directories.
- contains two sub-modules os.sys (same as sys) and os.path that are dedicated to the system and directories
- for network testing, file, directory, and path manipulations

```
#!/usr/bin/python


import sys,os


print (sys.version)
print (sys.api_version)
print (sys.platform)
```

```
┌──(kali㉿kali)-[~]
└─$ python ossys.py
3.10.4 (main, Mar 24 2022, 13:07:27) [GCC 11.2.0]
1013
linux
```

Modules

Custom Modules

custmod.py

```
#!/usr/bin/python
from srvvar import *


def checksrv(srv):
  if srv == srv1:
    return "The Web Server is " +  srv1
  elif srv == srv2:
    return "The Web Server is " +  srv2
  else:
    return "Its Tomcat"


print (checksrv(srv2))
```

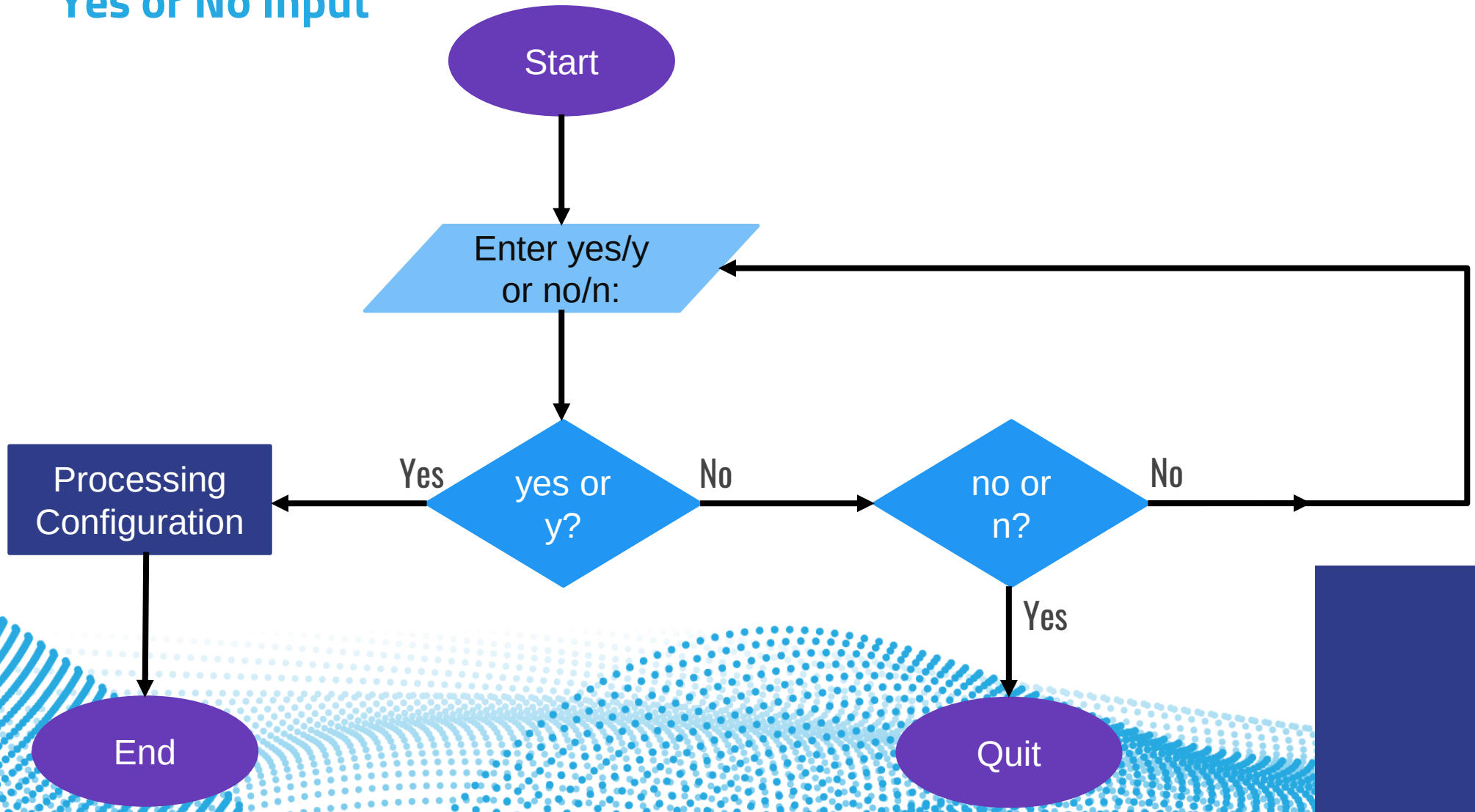
srvvar.py

```
srv1 = "apache"
srv2 = "nginx"
```

```
┌──(kali㉿kali)-[~]
└─$ python custmod.py
The Web Server is nginx
```

Project 1

Yes or No Input



Project 1

Yes or No User Input

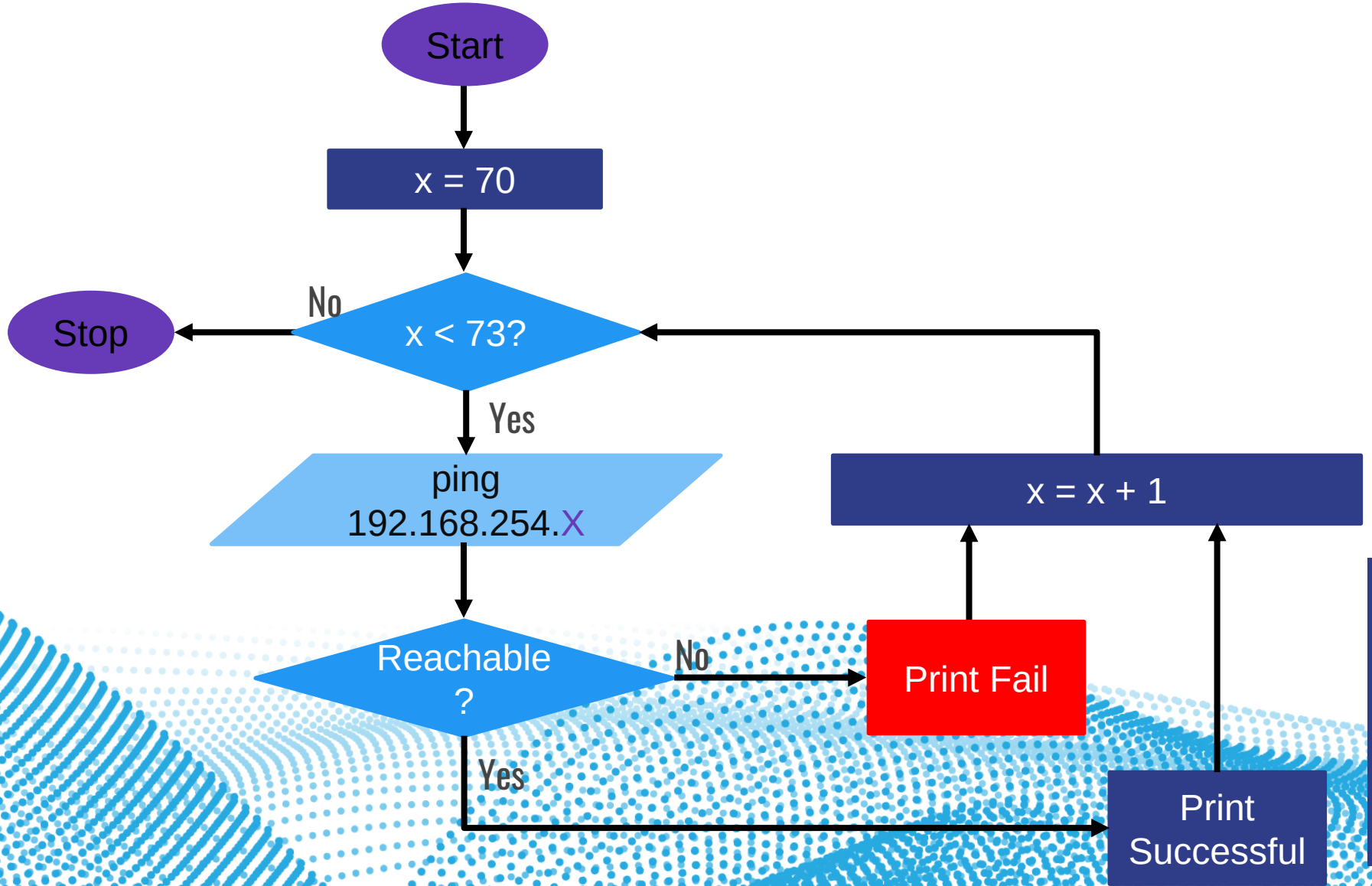
```
answ = ''
def yn(n):
  if answ == 'y':
    return "You entered - y"
  if answ == 'n':
    quit()
  else:
    return ("Uhhhh... please enter y or n only ")

while answ not in ['y', 'n']:
    answ = input('question  (y/n): ')
    print (yn(answ))

print ("Processing configuration")
```

Project 2

Host Reachability



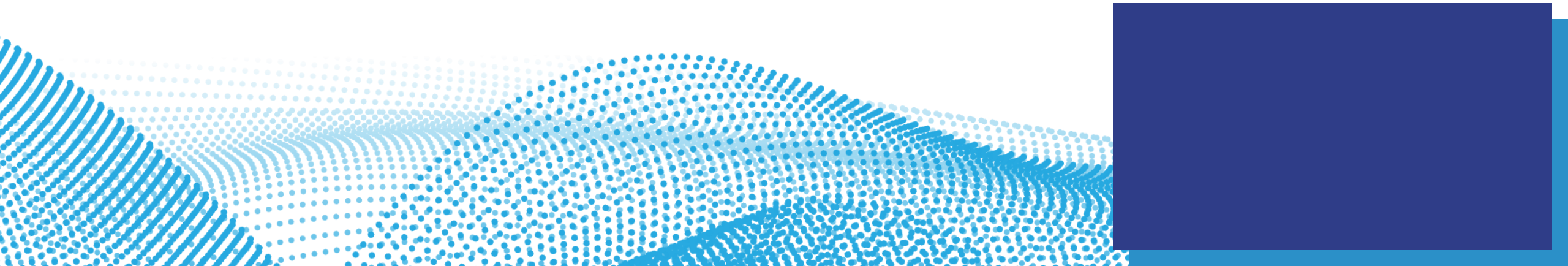
Project 2

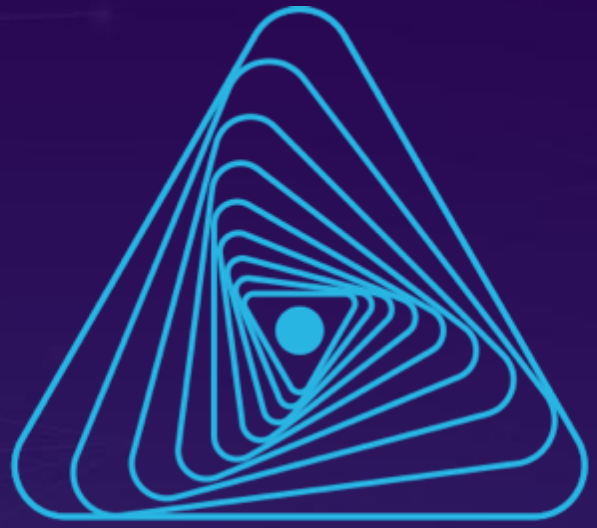
Test Reachability with Custom Module

```
import os,sys


for x in range (70,73):
  host = '192.168.254.' + str(x)
  conn = os.system('ping -c 1 ' + host + ' > /dev/null')

  if conn == 0 :
    print (host + " - Connection Successful")
  else :
    print (host + " - Connection Failed")
```





ARMADA

INNOVATION LABS

www.armadalabs.tech