Python I Installation, Data Structure, and Data Types

Sabtu, 18 Maret 2023

Arif Romadhan
Sr. Data Scientist

Program Zenius Studi Independen Bersertifikat
Zenius Bersama Kampus Merdeka

PT Cakra Syntesis Indonesia
Android Developer (2016)

Nawatech
Jr. ML Engineer (2017)

Bukalapak
Data Scientist (2018 - 2020)

Data Scientist Instructor
(Sept 2020 - Nov2020)

Data Scientist Instructor
(2021 - 2021)

Evermos
Sr. Data Scientist - Data Lead
(2021 - present)

Data Scientist Instructor
(2021- present)

Arif Romadhan

https://www.linkedin.com/in/arif-romadhan19/

1. Python Introduction
2. IDE Introduction
3. Python Variables & Data Types
4. Basic Operation on Data Types

Python Introduction

Python Introduction

Python

is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation

The founder of Python is Guido Van Rossum in the year of 1991

The most popular programming language is python ? (2022)

A. True

B. False

The most popular programming language is python ? (2022)

A. True

B. False

Python

Python is the most popular programming language.

Pos. Jan 2022	Pos. Jan 2021	Programming Language	Ratings	Chart Ratings	Variations
1	3	Python	13.58%		+1.86%
2	1	C	12.44%		-4.94%
3	2	Java	10.66%		-1.30%
4	4	C++	8.29%		+0.73%
5	5	C#	5.68%		+1.73%

A Flourish data visualization

As of 1 January 2022, the most popular programming language is Python. Python has a rating of 13.58%. This is an increase of +1.86 compared to January 2021. Thanks to this change, Python has moved from third position to first.

Why Python?

1. Data Analytics
2. Data Visualisation
3. Automation
4. Web Development & Testing
5. Machine Learning
6. Application

IDE Introduction

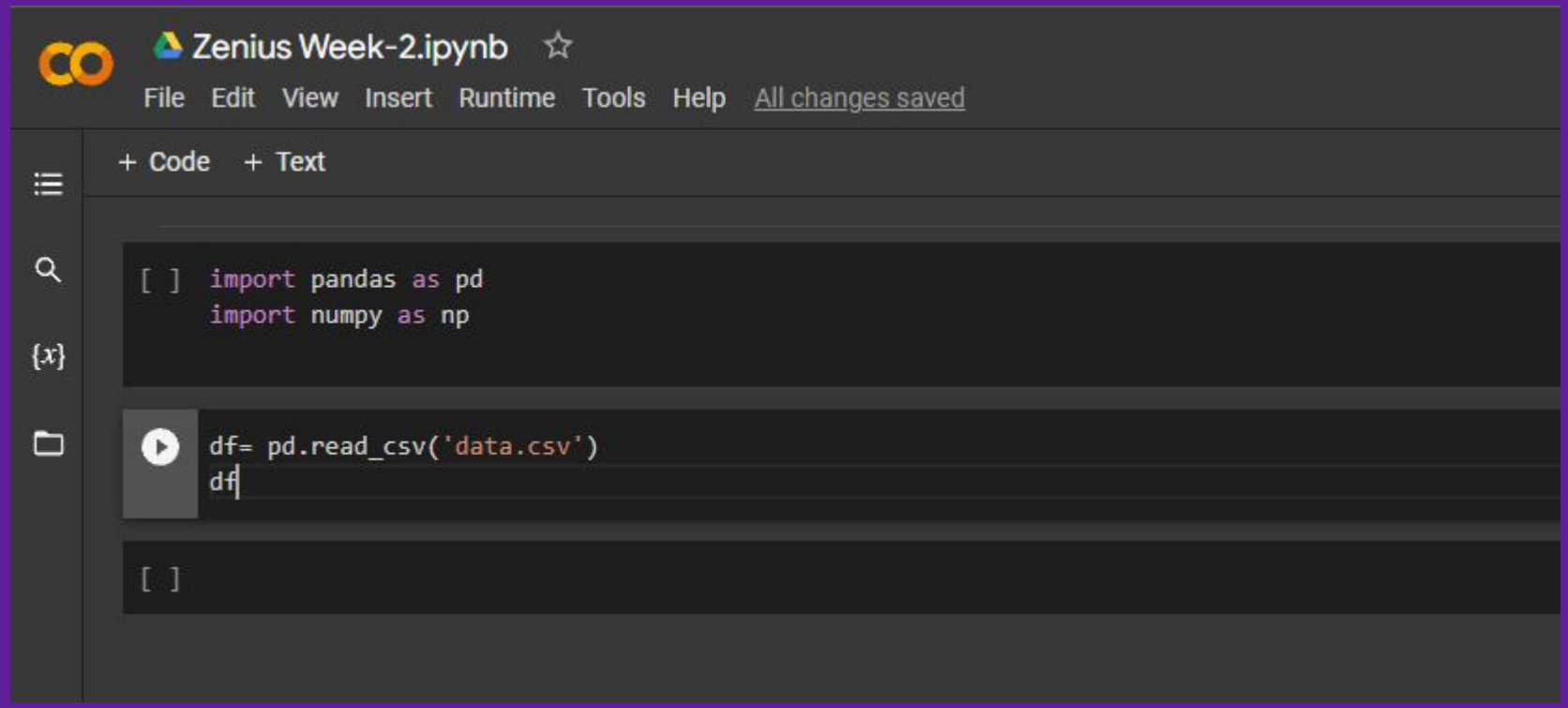
IDE Introduction

An integrated development environment (IDE) is software for building applications that combines common developer tools into a single GUI.

IDE for Python :

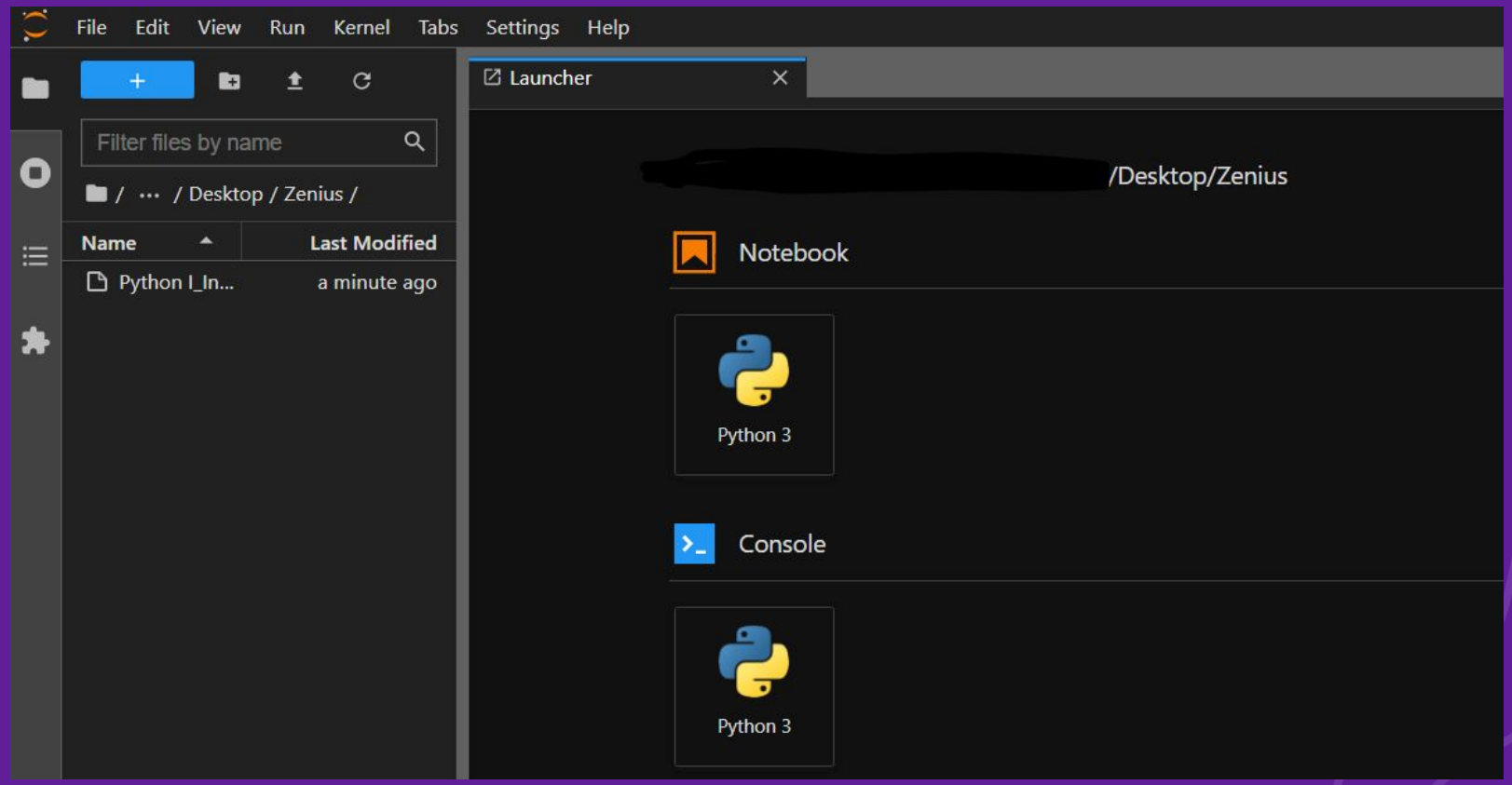
Jupyter Lab .
Visual Code Studio
Google Colab
Spyder

Google Colab



- Zero configuration required
- Access to GPUs free of charge
- Easy sharing

Jupyter Lab



- Web-based interactive
- Flexible interface
- Modular design

Live Coding

Let's get your hand dirty

Python Variables

Variable

Variables are containers for storing data values

```
In [ ]: umur=24
        nama="Budi"
```

Rules of creating a variable :

a. No space
b. No digit as prefix
c. No symbol or special characters as prefix

Python Data Types

Data Types

Types of data in Python :

1. Text
2. Numeric
   a. int
   b. float
3. Sequence
   a. list
   b. tuple
   c. range
4. Set
   d. set
   e. frozenset

Category	Type	Example
Text	str	"Zenius"
Numeric	int	12
	float	10.5
Sequence	list	["Vegemite","Burger","Gado-Gado"]
	tuple	("Vegemite","Burger","Gado-Gado")
	range	range(10)
Set	set	{"Name":""Steve","Hobby" :"Swimming"}
	frozen set	frozenset({"Name":""Steve","Hobby" :"Swimming"})

What is this datatype ?

```
hame = ['Yusup','Eko','Dindo']
```

A. List

B. Dictionary

C. Tuple

D. Set

What is this datatype ?

```
name = ['Yusup','Eko','Dindo']
```

A. List

B. Dictionary

C. Tuple

D. Set

What is this datatype ?

```
student['name']="Yusup"
student['age']="19"
```

A. List

B. Dictionary

C. Tuple

D. Set

What is this datatype ?

```
student['name']="Yusup"
student['age']="19"
```

A. List

B. Dictionary

C. Tuple

D. Set

Basic Operation in Python

Basic Operation– Numerical Data

Operation		Example	Output
+	Addition	1+5	6
-	Substraction	3 - 2	1
*	Multiplication	5 * 5	25
/	Division	14/2	7
**	Exponent	2**3	8
==	Equal	5 == 6	False
!=	Not Equal	6 != 7	True
>= ; <= ; > ; <	Camparison	6 >= 4	True

Basic Operation in Python

Challenge 1

Calculate in python..

```python
price=500
quantity=10
# insert formula here

print(total)
```

Basic Operation on String (Text Data)

```
[3] f_name="Yusup"
    l_name="Iskandar"

    f_name+l_name

    'YusupIskandar'
```

Concatenate both string

```
[4] f_name="Yusup"
    f_name.upper()

    'YUSUP'
```

Uppercase string

```
    f_name="YUSUP"
    f_name.lower()

    'yusup'
```

Lowercase string

Basic Operation on String (Text Data)

```
name="YUSUP"
name.replace("YU","WAS")
```

```
'WASSUP'
```

Replace String

```
[7] name="YUSUP"
    name=="YUSUP"
```

```
True
```

```
name="YUSUP"
name=="DINDA"
```

```
False
```

Conditional String
(Boolean Result)

Basic Operation in Python

Challenge 2

Given this phone number:

phone_number= "+62801-0029-9000"

replace "+628" into "0" and omit "-" symbols from the phone number

Basic Operation on Boolean Data

Operation	Description	Symbol
and	Both condition fulfilled	&
or	One of the condition is fulfilled	|
not	Not fulfilled	

Basic Operation on Boolean Data

```
soto="15000"
ketoprak="12000"

soto>ketoprak
```

```
True
```

```
soto<ketoprak
```

```
False
```

Basic Operation on Boolean Data

```
a=2
b=5
c=7

(a<b) and (c<a)
# True and False
```

```
False
```

```
[17] a=2
     b=5
     c=7

     (a<b) and (c>a)
     # True and True
```

```
True
```

Data Type Conversion

```
name="Wandy"
age= 12

name+age
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-1-46a15c01be75> in <module>()
      2 age= 12
      3
----> 4 name+age

TypeError: can only concatenate str (not "int") to str
```

The operation between different data types will always result in an error.

In this case, the addition of name(string) and age(int) will result in error (refer to the figure)

Data Type Conversion

```
name="Wandy"
age= 12

name+age
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-1-46a15c01be75> in <module>()
      2 age= 12
      3
----> 4 name+age

TypeError: can only concatenate str (not "int") to str
```

Solution?

Data Type Conversion

```
name="wandy"
age= 12
name+str(age)
```

```
'wandy12'
```

Data Type Conversion

```
# convert to string
a=str(123)

# convert to integer
b=int("123")

# convert to float
c=float("125.2")
```

Data Type Conversion

```
# convert to string
a=str(123)

# convert to integer
b=int("123")

# convert to float
c=float("125.2")
```

Python Data Type - List

What is List?

Lists are used to store multiple items in a single variable.

```
[ ] cities=['Jakarta','Medan','Surabaya','Bogor']
```

Declaring List

Many ways to declare list :

```
[ ] new_list=[]

    new_list=["Soto","Bakso","Ketoprak"]

    new_list=[14,15,"Jakarta"]

    new_list=[["Jakarta","Semarang"],"Medan"]
```

Empty list

List of strings

List of various data types (str & int)

List in List

List Structure

```
food=["Soto","Bakso","Ketoprak","Gado-Gado","Sate","Nasi Goreng"]
```

Index	0	1	2	3	4	5

Index of list always starts from 0

```
food=["Soto","Bakso","Ketoprak","Gado-Gado","Sate","Nasi Goreng"]
food[1]
```

```
'Bakso'
```

Food[1] will grab the second index from the list . ⇒ “Bakso”.

List Structure

```
food=["Soto","Bakso","Ketoprak","Gado-Gado","Sate","Nasi Goreng"]
```

Food[1:3] will grab value from index 1 until 2

```
food[1:3]
['Bakso', 'Ketoprak']
```

Food[3:] will grab all value after 3

```
[5] food[3:]
    ['Gado-Gado', 'Sate', 'Nasi Goreng']
```

Appending Value into List

```
food=["Soto","Bakso","Ketoprak","Gado-Gado","Sate","Nasi Goreng"]
```

Append() will let you add/append value into your list

```
food.append('Ayam Goreng')

food

['Soto',
 'Bakso',
 'Ketoprak',
 'Gado-Gado',
 'Sate',
 'Nasi Goreng',
 'Ayam Goreng']
```

Deleting Value from List

```
food=["Soto","Bakso","Ketoprak","Gado-Gado","Sate","Nasi Goreng","Ayam Goreng"]
```

Use remove() to delete value from list

```
food.remove("Sate")
```

```
food
```

```
['Soto', 'Bakso', 'Ketoprak', 'Gado-Gado', 'Nasi Goreng', 'Ayam Goreng']
```

Remove("Sate") will delete it from the list

Getting List Length

len() will return you the size/length of the list

```
vehicles=['Bus','Plane','Car']
len(vehicles)
```

```
3
```

Multi-dimensional List

List can be multidimensional

```
[1] students=[['Kevin','Michael'],['25','23','30']]
    students

    [['Kevin', 'Michael'], ['25', '23']]
```

students[0][0] will return Kevin
students[0][1] will return Michael
students[1][0] will return 25
Students[1][2] will return 30

Python Data Type - List

Challenge 1

Given this list of number:

num_list=['12','14',[16,20,30]]

From the list :

1. Grab 14
2.Grab 16,20,30
3.Grab 20

Basic Function - List

There are many functions for python List. These are some of the function:

Operation	Description
min()	Get lowest value from list
max()	Get highest value from list
sum()	Sum all value

```
num_list=[20,30,25,50]
print(min(num_list))
print(max(num_list))
print(sum(num_list))
```

```
20
50
125
```

Python Data Type - List

Challenge 2

Basic Function - List

Given this list of number:

numbers=[2,3,6,20,22]

From the list :
1. Find the differences between the highest number and lowest number
2. Get the total value of numbers in list
3. Grab the 2nd element until the 4th element
4. Insert new number in the list => number 50

Python Data Type Dictionary

Dictionary

Unordered collections that contains pairs of *key* and *value*

```
students=
{
'name':'Cindy',
'age':'21',
'hobby':'Racing'
}
```

Dictionary

Unordered collections that contains pairs of *key* and *value*

Key

```
students=
{
'name':'Cindy',
'age':'21',
'hobby':'Racing'
}
```

Value

Dictionary

Dictionary can be declared as empty dictionary.

```
[2] students={}
```

Dictionary

Dictionary can have different type of values.

```
fruits_dic = {
    "Fruit":["Mango","Banana"],
    "Color":["Blue", "Red"],
    "Quantity":[10,25]
    }
```

Creating Dictionary

To create dictionary, follow the format below:

```
Dict_name = {
Key:Value,
Key:value,
Key:value,
}
```

```
fruits_dic = {
    "Fruit":["Mango","Banana"],
    "Color":["Blue", "Red"],
    "Quantity":[10,25]
    }
```

There are no limit of how many keys we can put on dictionary.

Python Data Type Dictionary

Challenge 1

Create a dictionary containing these key-value:

‘brand’=[‘Samsung’,’Dell’,’Apple’]
‘stock’=[‘27’,’20’.’10’]
‘type’=[‘PC’,’Laptop’,’Tablet’]

Getting All Keys from Dictionary

To get all keys from a dictionary, follow the format below:

dict.keys*(dict_name)*

```
[1] fruits_dict = {
        "Fruit":["Mango","Banana"],
        "Color":["Blue", "Red"],
        "Quantity":[10,25]
        }

dict.keys(fruits_dict)

dict_keys(['Fruit', 'Color', 'Quantity'])
```

In the code sample, it will return all keys in *fruits_dict*

Getting Value from Dictionary

To get all value from specific key in a dictionary, follow the format below:

dict_name[key]

```
[1] fruits_dict = {
        "Fruit":["Mango","Banana"],
        "Color":["Blue", "Red"],
        "Quantity":[10,25]
        }

[3] fruits_dict['Fruit']

    ['Mango', 'Banana']
```

In the code sample, it will return value from key “fruit”

Getting Value from Dictionary

You can drill down to the value in dictionary

```
[1] fruits_dict = {
        "Fruit":["Mango","Banana"],
        "Color":["Blue", "Red"],
        "Quantity":[10,25]
        }

[3] fruits_dict['Fruit']

    ['Mango', 'Banana']
```

This is now a list data type. Hence, you can apply list data operation.

```
[4] fruits_dict['Fruit'][0]

    'Mango'
```

This will grab the 1st index from list `['Mango', 'Banana']`

Output

Python Data Type Dictionary

Challenge 2

From our existing dictionary (fruits_dict):

```
[1] fruits_dict = {
        "Fruit":["Mango","Banana"],
        "Color":["Blue", "Red"],
        "Quantity":[10,25]
        }

[3] fruits_dict['Fruit']

    ['Mango', 'Banana']
```

1. Get all of the Quantity value
2. Get the 2nd element from key "Color"

Updating Value from Dictionary

To update value from a dictionary, follow the format below

dict_name[key]=[new value]

or

dict_name[key][index]=[new value]

```
fruits_dict = {
    "Fruit":["Mango","Banana"],
    "Color":["Blue", "Red"],
    "Quantity":[10,25]
    }
```

```
[6] fruits_dict['Color']=['Green','Red','Blue']
    fruits_dict
```

```
{'Color': ['Green', 'Red', 'Blue'],
 'Fruit': ['Mango', 'Banana'],
 'Quantity': [10, 25]}
```

```
fruits_dict = {
    "Fruit":["Mango","Banana"],
    "Color":["Blue", "Red"],
    "Quantity":[10,25]
    }
```

```
fruits_dict['Color'][0]='Black'
fruits_dict
```

```
{'Color': ['Black', 'Red'], 'Fruit': ['Mango', 'Banana'], 'Quantity': [10, 25]}
```

Python Data Type Dictionary

Challenge 3

Given this dictionary:

```
Students={
`name':[`Budi','Jessica']
`age':[19,24]
}
```

1.Update the value of name into [`Arya','Yusup']
2.Update age value 19=>22

Adding Key and Value to Dictionary

To add key & value to a dictionary, follow the format below

Adding both key & value

dict_name.update({new_key:new_value})

```
[10] fruits_dict = {
         "Fruit":["Mango","Banana"],
         "Color":["Blue", "Red"],
         "Quantity":[10,25]
         }
```

```
fruits_dict.update({'Price':[3000,2000]})
fruits_dict
```

```
{'Color': ['Blue', 'Red'],
 'Fruit': ['Mango', 'Banana'],
 'Price': [3000, 2000],
 'Quantity': [10, 25]}
```

Adding Value to Specific Key Dictionary

Let's say we need to add one more fruit in => key = "Fruit"
We can just apply the append method as in list

```
[1] fruits_dict = {
        "Fruit":["Mango","Banana"],
        "Color":["Blue", "Red"],
        "Quantity":[10,25]
        }

[3] fruits_dict['Fruit']

    ['Mango', 'Banana']
```

List

List

```
fruits_dict['Fruit'].append('Melon')
fruits_dict

{'Color': ['Blue', 'Red'],
 'Fruit': ['Mango', 'Banana', 'Melon'],
 'Quantity': [10, 25]}
```

Python Data Type Dictionary

Challenge 4

Given this dictionary:

```
Students={
`name':[`Budi','Jessica']
`age':[19,24]
}
```

1. Add `Amar' to the key => name
2. Add new key and value:
   `Gender':[`Male','Female']

Deleting Key from Dictionary

To delete key from a dictionary, follow the format below:

del dict_name[‘keyname’]

```
fruits_dict = {
    "Fruit":["Mango","Banana"],
    "Color":["Blue", "Red"],
    "Quantity":[10,25]
    }
```

```
del fruits_dict['Quantity']
fruits_dict
```

```
{'Color': ['Blue', 'Red'], 'Fruit': ['Mango', 'Banana']}
```

Deleting a key will remove all associated value with it

Deleting Value from Dictionary

To delete value from a dictionary, follow the format below:

dict_name[‘key name’].remove(‘value’)

```
fruits_dict = {
    "Fruit":["Mango","Banana"],
    "Color":["Blue", "Red"],
    "Quantity":[10,25]
    }

fruits_dict['Color'].remove('Blue')
fruits_dict
```

```
{'Color': ['Red'], 'Fruit': ['Mango', 'Banana'], 'Quantity': [10, 25]}
```

Removed blue from the “Color”

Python Data Type Dictionary

Challenge 5

Given this dictionary:

```
Students={
`name':[`Budi','Jessica']
`age':[19,24]
}
```

1.Delete "Budi" from key=> name
2.Delete age from the dictionary

Thank you!

Any Questions?