Module 01

Lambda Exp Tuples & Etc

Data Science Developer

Purwadhika
Startup and Coding School

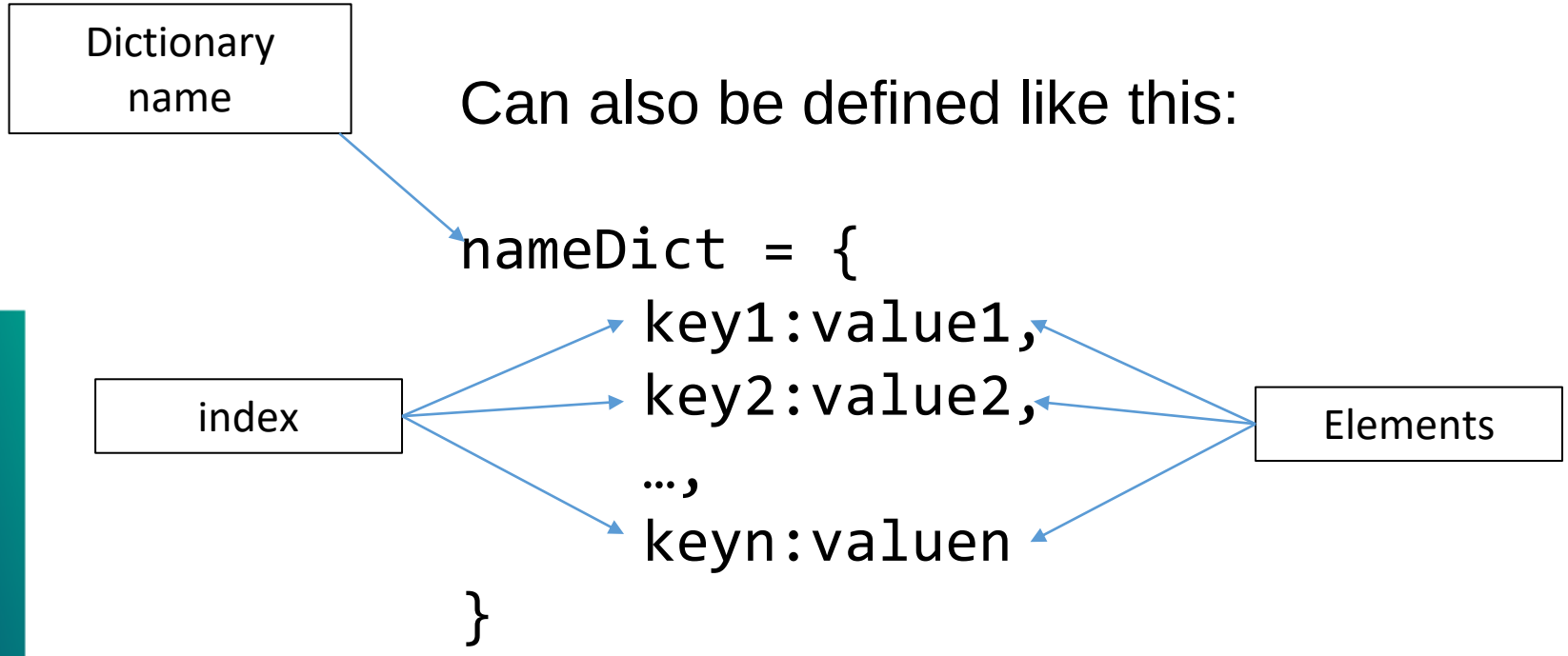
Dictionary

- Dictionary is data type that can hold multiple data with different type
- So far we know that another data type that can hold multiple data area list and string.
- **Dictionary is similar like lists but different in the index.**.
- We define index in dictionary by our self and the index is actually known as key.
- Order in dictionary doesn’t matter.
- Dictionary can be used for mapping

Purwadhika
Startup and Coding School

General Structure

```
nameDict = {key1:value1, key1:value2, …,}
```



Purwadhika
Startup and Coding School

Dictionaries

```
d = { "key1" : "item1", "key2" : "item2",
"kucing" : [3, "jerapah"] };

print(d["key1"]);
print(d["key2"]);
print(d["kucing"]);
print(d["kucing"][1]);
```

Purwadhika
Startup and Coding School

Dictionaries inside Dictionaries

```
d = { "key1" : { "key2" : "item2" },
"kucing" : [3, "jerapah"] };

print(d["key1"]);
print(d["key1"]["key2"]);
print(d["kucing"]);
print(d["kucing"][1]);
```

Purwadhika
Startup and Coding School

Methods in dictionary

Method	Parameters	Description
keys	none	Return all keys
values	none	Return all values
items	none	Return all pair value-key
get	key	Return a value associated with the key, None otherwise
pop	none	Removes the element with the specified key
popitem	none	Removes the last inserted key-value pair
clear	none	Removes all the elements from the dictionary
copy	none	Returns a copy of the dictionary

Purwadhika
Startup and Coding School

Methods in dictionary

```
dict_age1 = {
        "muhyi":24,
        "alfa":22,
        "baron":25
}


print(dict_age1.keys())
print(dict_age1.values())
print(dict_age1.items())
print(dict_age1.get("muhyi"))
```

```
dict_age1 = {
        "muhyi":24,
        "alfa":22,
        "baron":25,
        "lucas":23
}


print(dict_age1.pop("muhyi"))
print(dict_age1.pop("alfa"))
print(dict_age1.popitem())
```

Purwadhika
Startup and Coding School

Changing element inside dictionary

```
dict_age1 = {
      "muhyi":24,
      "alfa":22,
      "baron":25,
}

print(dict_age1)
```

```
dict_age2 = {}

dict_age2["muhyi"] = 24
dict_age2["alfa"] = 22
dict_age2["baron"] = 25

print(dict_age2)
```

Purwadhika
Startup and Coding School

Changing element inside dictionary

```
dict_rahmah = {
      "name":"Rahmah",
      "age":25,
      "live":"Bogor",
      "job":"Banker"
}

dict_rahmah["age"] = 26
dict_rahmah["live"] = "Jakarta"
print(dict_rahmah)
```

Purwadhika
Startup and Coding School

Changing element inside dictionary

See what happen in this code:

```
dict_rahmah = {
        "name":"Rahmah",
        "age":25,
        "live":"Bogor",
        "job":"Banker"
}
dict_rahmah2 = dict_rahmah

dict_rahmah2["age"] = 26
dict_rahmah2["live"] = "Jakarta"
print(dict_rahmah)
```

See what happen in this code too:

```
dict_rahmah = {
        "name":"Rahmah",
        "age":25,
        "live":"Bogor",
        "job":"Banker"
}
dict_rahmah2 = dict_rahmah.copy()

dict_rahmah2["age"] = 26
dict_rahmah2["live"] = "Jakarta"
print(dict_rahmah)
print(dict_rahmah2)
```

Purwadhika
Startup and Coding School

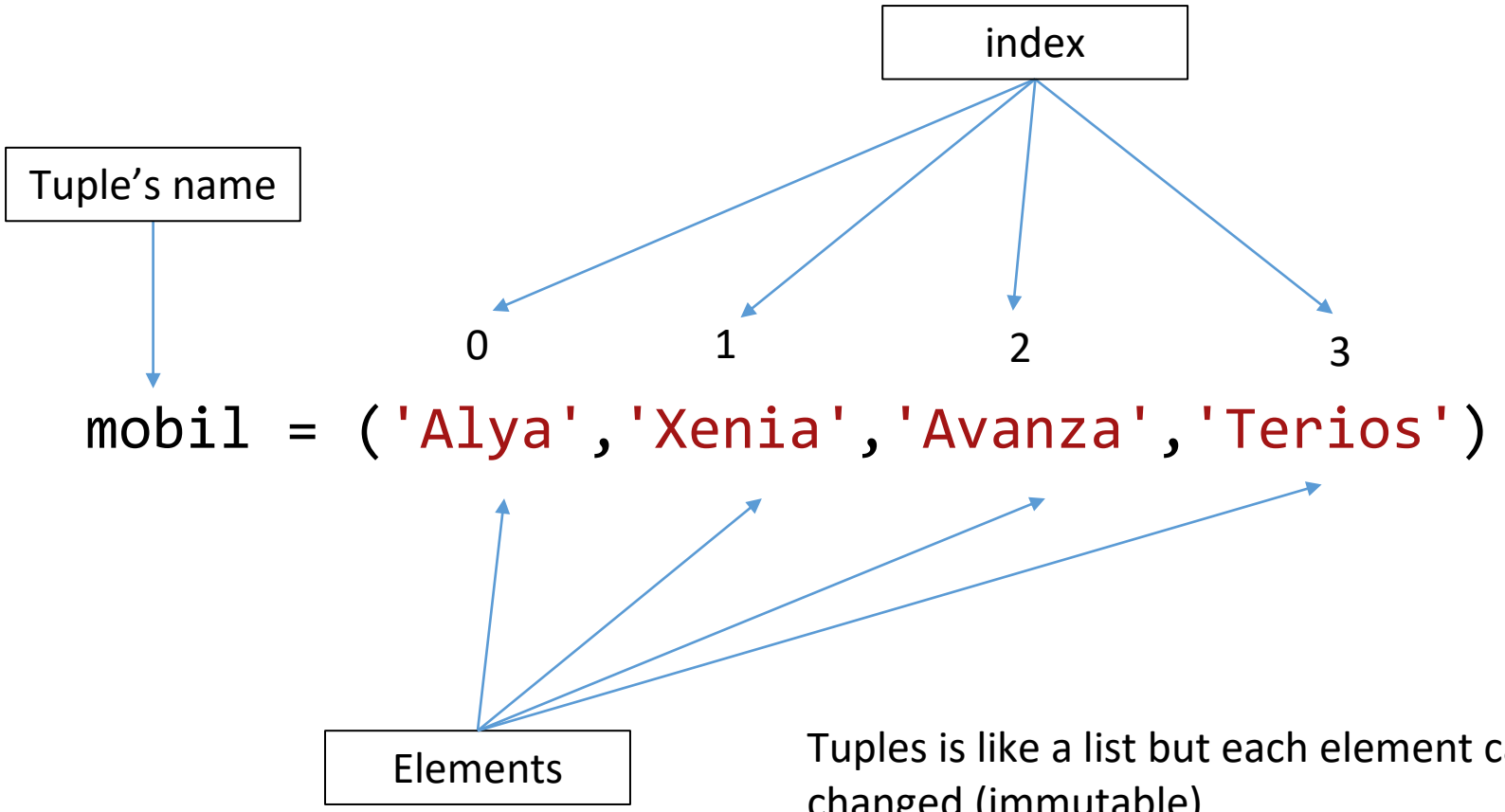
Operation in Dictionary

```
dict_age1 = {
      "muhyi":24,
      "alfa":22,
      "baron":25,
}

del dict_age1["muhyi"]
print(dict_age1)
dict_age1.clear()
print(dict_age1)
```

Purwadhika
Startup and Coding School

Tuple



Tuples is like a list but each element can't be changed (immutable)

Purwadhika
Startup and Coding School

Tuple

```
Tuple1 = 1,2,3
print(Tuple1)

Tuple2 = (1,2,)
print(Tuple2)

Tuple3 = ()
print(Tuple3)
print(type(Tuple3))
```

```
Tuple4 = tuple([1,2,3)
print(Tuple4)

Tuple5 = tuple('abc')
print(Tuple5)

Tuple6 = tuple((1,2,3))
print(Tuple6)
```

Purwadhika
Startup and Coding School

Tuples

```
t = (1, [0, "test"], { "a1" : True })

print(t[2]["a1"])
print(t[1][1])
t[1][1] = "akan"
print(t[1][1])
t[1] = "mark"
print(t[1])
```

Purwadhika
Startup and Coding School

Tuples inside Tuples

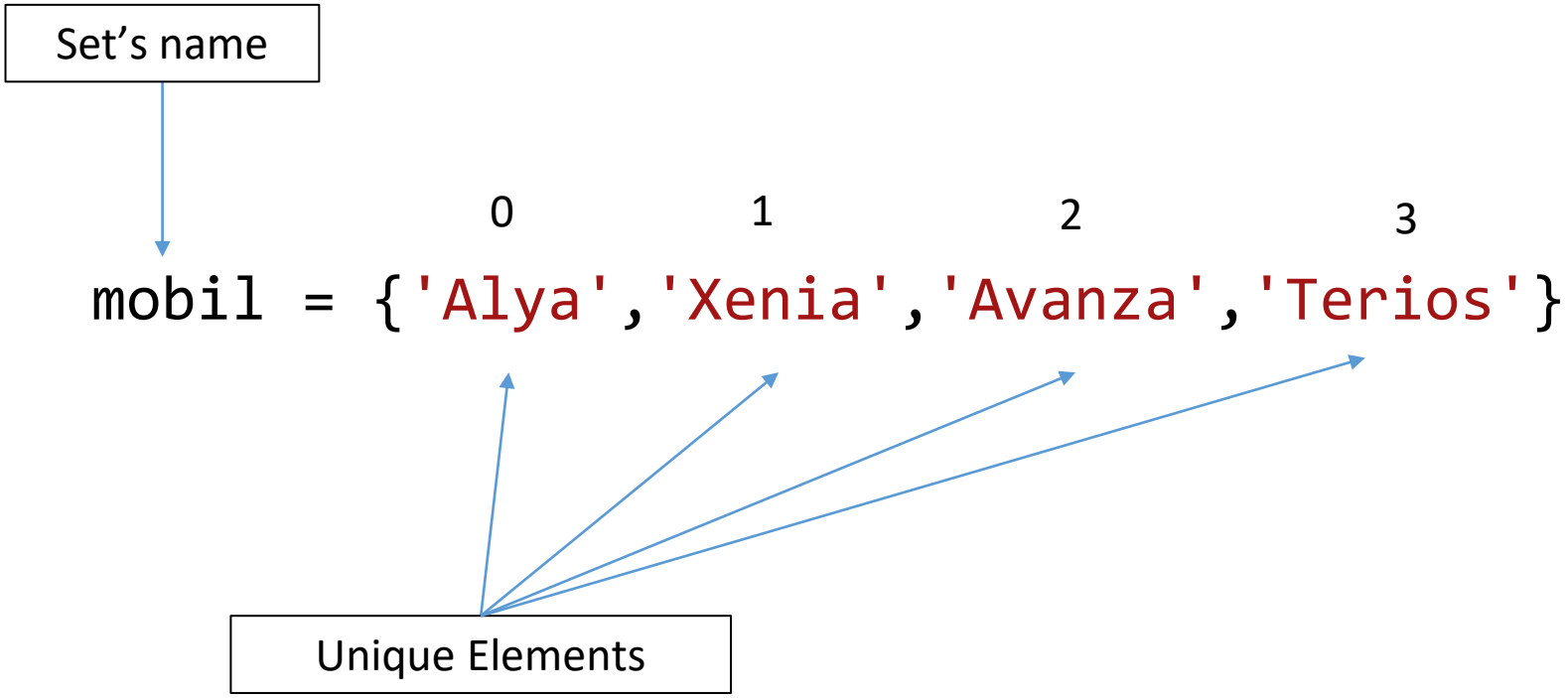
```
t = (1, [0, "test"], { "a1" : True },
(0,{ "test" : 5 },2));

print(t[3][1]["test"]);
```

Purwadhika
Startup and Coding School

Sets

Sets doesn't support indexing, there isn't duplicate items in set (every item unique).



Purwadhika
Startup and Coding School

Sets

Sets doesn't support indexing, there isn't duplicate items in set (every item unique).

```
s = { 1, 3, 1, 2, 2, 3 };

print(s);
print(list(s)[2]);
```

Purwadhika
Startup and Coding School

Filtering List using Set

```
newList = [ 1, 3, "test1", "test2" , 2, 3, "test1" ];
s = set(newList);

print(s);
print(list(s)[2]);
```

Purwadhika
Startup and Coding School

List Comprehension

```
listNum = [ 1, 2, 3, 4, 5];
listNum = [item * 2 for item in listNum];
print(listNum);
```

Purwadhika
Startup and Coding School

List Comprehension

```
def times2(num) :
    return num * 2;

listNum = [ 1, 2, 3, 4, 5];
listNum = [times2(item) for item in listNum];
print(listNum);
```

Purwadhika
Startup and Coding School

Lambda Expressions

```
def times2(num) :
    return num * 2;

lambda num: num * 2;
```

Purwadhika
Startup and Coding School

Map

Without Lambda (using function) :

```
def times2(num) :
    return num * 2;

listNum = [ 1, 2, 3, 4, 5];
listNum = list(map(times2, listNum));
print(listNum);
```

With Lambda :

```
listNum = [ 1, 2, 3, 4, 5];
listNum = list(map(lambda num: num * 2, listNum));
print(listNum);
```

Purwadhika
Startup and Coding School

Filter

Without Lambda (using function) :

```
def genap(num) :
    return num % 2 == 0;

listNum = [ 1, 2, 3, 4, 5];
listNum = list(filter(genap, listNum));
print(listNum);
```

With Lambda :

```
listNum = [ 1, 2, 3, 4, 5];
listNum = list(filter(lambda num: num % 2 == 0, listNum));
print(listNum);
```

Purwadhika
Startup and Coding School

Methods for Searching

```
numList = [1,2,3];
input = 'x';

check1 = input in numList;
check2 = 'x' in ['x','y','z'];
check3 = 'ka' in 'kurakas';

print(check1);
print(check2);
print(check3);
```

Purwadhika
Startup and Coding School

Solve It! #1

Buatlah aplikasi python sederhana untuk filtering list (searching) berdasarkan input user seperti dibawah ini.

```
PS D:\Purwadhika\Purwadhika\Python Fundamental> python sc
['Merdeka', 'Hello', 'Hellos', 'Sohib', 'Kari ayam']
Search : ka
['Merdeka', 'Kari ayam']
PS D:\Purwadhika\Purwadhika\Python Fundamental> python sc
['Merdeka', 'Hello', 'Hellos', 'Sohib', 'Kari ayam']
Search : hel
['Hello', 'Hellos']
```

Purwadhika
Startup and Coding School