LESSON 5

main.py

```
def average(nums):
    return sum(nums)
        / len(nums)
```

Py

Functions

Learn it all @MyWebUniversity



A FREE MYWEBUNIVERSITY COURSE

PYTHON SCRIPTING & OOP

LESSON 5

Functions

def, parameters, defaults, *args, **kwargs, scope, lambda and docstrings

Defining Functions

functions.py

```
def average(numbers):
    """Return the average of a list of numbers."""
    if not numbers:
        return 0
    return sum(numbers) / len(numbers)

print(average([88, 95, 72]))      # 85.0
```

- def, a name, parameters in parentheses, a colon, and an indented body. return sends a value …
- A docstring in triple quotes explains what the function does, and help shows it.

01

Defaults and Keyword Arguments

defaults.py

```
def greet(name, greeting="Salaam", punct="!"):
    return f"{greeting}, {name}{punct}"

greet("Mariam")                          # Salaam, Mariam!
greet("Ahmad", punct=".")                # Salaam, Ahmad.
def add_item(item, items=None):             # not items=[]
    items = items or []; items.append(item); return items
```

- Parameters can have default values, and callers can pass arguments by name, in any order. …
- One warning: never use a list as a default value, because the same list is shared between …

02

*args and **kwargs

args.py

```python
def total(*numbers):
    return sum(numbers)
total(1, 2, 3, 4)                  # 10

def make_user(name, **info):
    return {"name": name, **info}
make_user("Zahra", city="Herat", year=3)
```

- Star args collects any number of extra positional arguments into a tuple. Double star …
- They let you write flexible functions, and pass arguments through to other functions.

03

Scope

Avoid

- count = 0
- def add(): global count; count += 1
- hidden changes

Prefer

- def add(count):
- return count + 1
- clear in, clear out

- Variables created inside a function are local and disappear when it returns. Code inside a …
- Good design avoids global variables: pass values in as parameters and return results.

04

lambda and Functions as Values

lambda.py

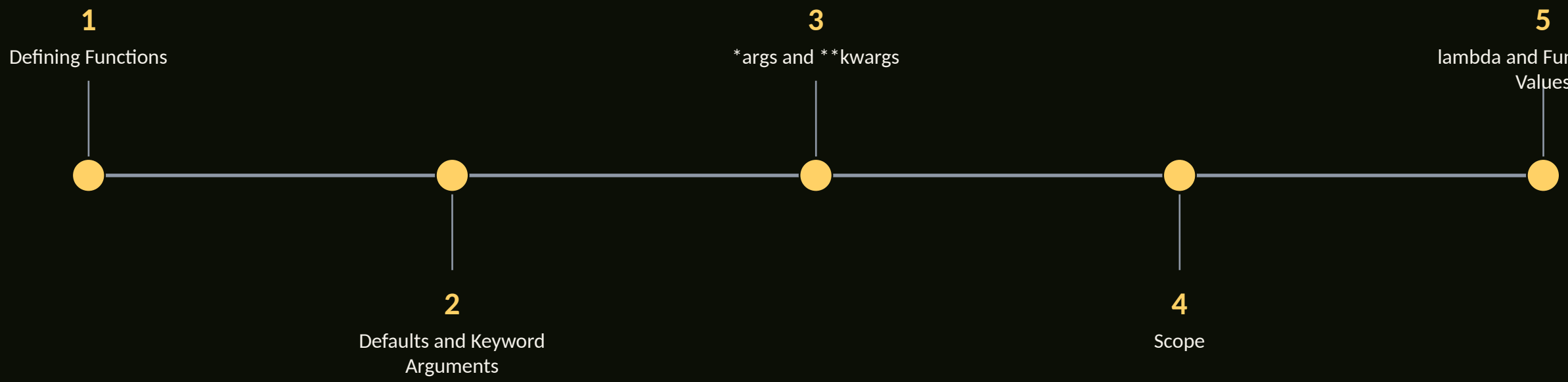
```
students = [("Mariam", 95), ("Ahmad", 88), ("Zahra", 91)]
students.sort(key=lambda s: s[1], reverse=True)
# [('Mariam', 95), ('Zahra', 91), ('Ahmad', 88)]

square = lambda x: x * x
list(map(square, [1, 2, 3]))        # [1, 4, 9]
```

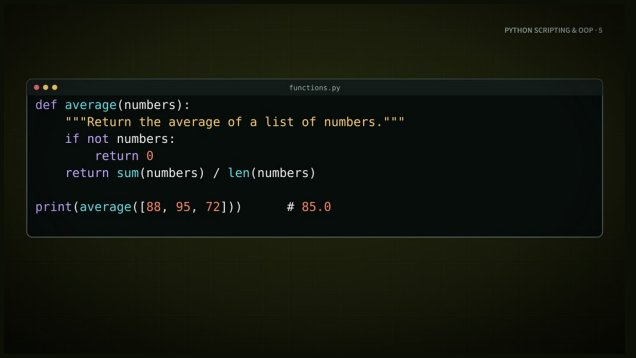
- Functions are values: you can pass them to other functions. lambda creates a tiny unnamed …
- The most common use is as a key for sorting, like sorting students by their score.

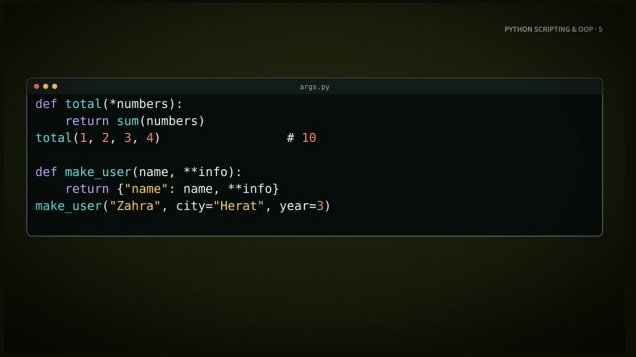
05

Lesson Map


1
Defining Functions
2
Defaults and Keyword Arguments
3
*args and **kwargs
4
Scope
5
lambda and Functions as Values

Sources and Further Reading





Thank you for watching

MyWebUniversity.com

Free technical education since 2000 — English and Dari

@MyWebUniversity · @MyWebUniversity-Dari