CITY
UNIVERSITY OF LONDON
EST 1894

STEM Digital Academy

School of Science & Technology

www.city.ac.uk

Introduction to User Defined Functions

Programming and Algorithms

Lecture by

Dr Daniil Osudin

```
n = 3
for i in range(1,n+1):
    print("Hello World!")
```

```
Hello World!
Hello World!
Hello World!
```

What will we Cover?

- Introduction of user defined functions
- Purpose of user defined functions and their components
- Defining custom functions

What is a Function?

- Self-contained block of code
- A group of statements written to perform a specific task
- Fundamental program structure and decomposition tool

Why use Functions?

- Break larger or complex programs into smaller, more manageable parts
- Easier to develop, manage and debug larger programs
- Code reusability – once written, the function can be called within a program multiple times

Defining a Function

- User defined function syntax and formatting:

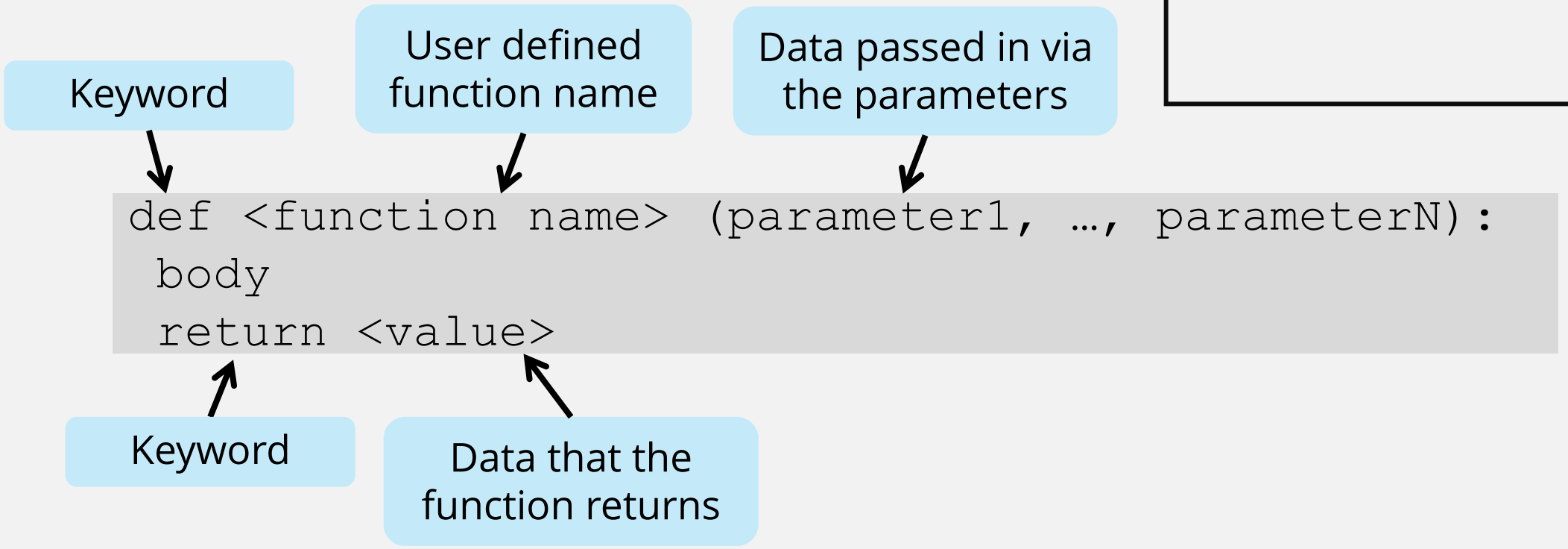
```
def <function name> (parameter1, …, parameterN):
 body
 return <value>
```

One or multiple statements in the body

All statements in the body and the return statement need to be indented the same

Colon separates the header from the body

Function Components



Function Header

- Contains the keyword `def`, which indicates the start of the function
- Function name – similar naming conventions to the variables
- Function parameters – none, one or more separated by comma and placed within parenthesis. Parameters allow data to be passed to the function
- Colon – to mark the end of the function header

Function Body

- One or more statements
- Need to be indented at the same level similarly to other blocks of code
  - For example, IF statements or FOR loops
- Statements are executed when the function is called
- Function body may perform some calculation and then the return statement is used to pass it back to the program

Return Statement

- Return transfers the control back to the program where the function was called
- Contains a keyword `return` followed by one or multiple values to be returned
- Return statement and the value are optional
  - If a function does not need to return a result (for example it prints something instead), the return keyword is not used
  - A `return` without a value is used to exit the function's body sooner

Function Call

- A function call is a statement that shifts the control from the program to the function
- A function call consists of
  - The name of the function being called
  - None, one or more arguments separated by a comma
    - Arguments are the data passed to the called function (via the function parameters)
    - The number of arguments and the number of parameters must match