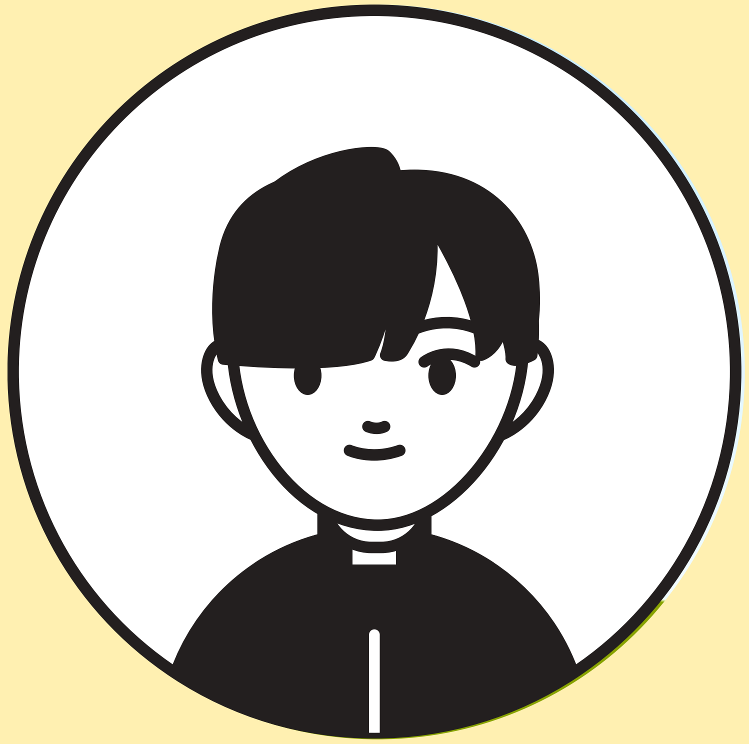
Python Programming

Bootcamp

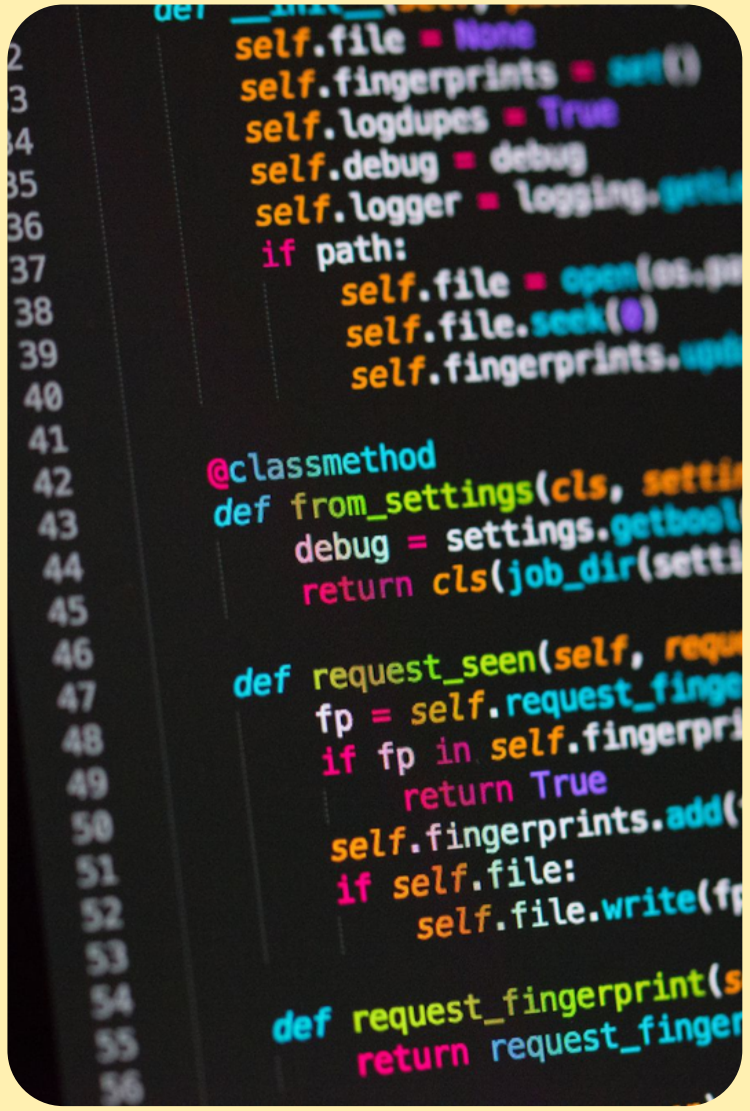
Presented by



Abdulziz

Introduction

Welcome to our Python Programming Bootcamp!
Explore the versatile Python language’s
fundamentals and applications in just a few weeks.
Get ready to code and unlock countless
possibilities with Python!



```
        self.file = None
        self.fingerprints = set()
        self.logdupes = True
        self.debug = debug
        self.logger = logging.getL
        if path:
            self.file = open(os.pa
            self.file.seek(0)
            self.fingerprints.upd

    @classmethod
    def from_settings(cls, settin
        debug = settings.getbool(
        return cls(job_dir(setti

    def request_seen(self, reque
        fp = self.request_finge
        if fp in self.fingerpri
            return True
        self.fingerprints.add(
        if self.file:
            self.file.write(fp

    def request_fingerprint(s
        return request_finger
```

Python Language

Background

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec blandit lectus sit amet urna finibus luctus. Vestibulum vitae ante et nisi dapibus tincidunt. Aenean dignissim nibh risus, vel consectetur tellus cursus nec.

The Basics

- =: Assignment operator, assigns a value to a variable.
- print(): Displays text or variables to the console.
- str(): Converts a value to a string data type.
- type(): Returns the data type of a variable.
- int(): Converts a value to an integer data type.
- float(): Converts a value to a floating-point number data type.

String Methods

- len(): Measures the length of a string.
- find(): Searches for the index of a substring in a string.
- upper(): Converts a string to uppercase letters.
- lower(): Converts a string to lowercase letters.
- isdigit(): Checks if a string consists of digits.
- isalpha(): Checks if a string consists of alphabetic characters.
- count(): Counts the occurrences of a substring in a string.
- replace(): Replaces specified substrings with another substring in a string.

Logical Operators

- and: Logical AND operator, returns true if both conditions are true.
- or: Logical OR operator, returns true if at least one condition is true.
- not: Logical NOT operator, negates the condition (T => F or F => T).
- if: executes code block if the condition is true.
- else: Executes an alternative code block if the if condition is false.
- elif: Short for "else if," adds additional conditions to check.
- if not: Checks the opposite of a condition and executes code block if it’s false.

Math Methods

- round(): Rounds a number to the nearest integer.
- math.ceil(): Rounds a number up to the nearest integer
- math.floor(): Rounds a number down to the nearest integer
- abs(): Returns the absolute value of a number.
- pow(): Raises a number to a specified power.
- math.sqrt(): Calculates the square root of a number
- max(): Returns the maximum value among multiple numbers.
- min(): Returns the minimum value among multiple numbers.
- math.sin(): Calculates the sine of an angle in radians
- math.cos(): Calculates the cosine of an angle in radians

Type cast

- bool(): Converts a value to a boolean (True or False) data type.
- list(): Converts an iterable (e.g., tuple or string) to a list.
- tuple(): Converts an iterable (e.g., list or string) to a tuple.
- set(): Converts an iterable (e.g., list or tuple) to a set.
- dict(): Converts a sequence of key-value pairs (e.g., a list of tuples) to a dictionary.
- chr(): Converts an integer Unicode code point to a character.
- ord(): Converts a character to its integer Unicode code point.

User Input

- input(): Reads user input as a string from the console.

Casting Is Important in User Input:

- int(): Converts a string to an integer data type.
- float(): Converts a string to a floating-point number data type.

String Slicing

- name[index]: Accesses a specific character in a string by its index.
- name[start:end]: Extracts a substring from the string, starting from the "start" index (inclusive) and ending at the "end" index (exclusive).
- name[:end]: Extracts a substring from the begin to the "end" index.
- name[start:]: Extracts a substring from the "start" index to the end of string.
- name[::-1]: Reverses the string.
- name[start:end:step]: Extracts a substring with a specified step value.
- index operator [] also gives access to list,tuples

If statemens

- if: Conditional control structure for decision-making.
- elif: Short for "else if," used for additional conditions.
- else: Executes when the if condition is false.
- >=: Greater than or equal to comparison operator.
- <: Less than comparison operator.
- <=: Less than or equal to comparison operator.

Note:

int(input()): Converts user input to an integer.

For loop

- for: Iterative control structure for executing code a limited number of times.
- range(stop): Generates numbers from 0 to (stop-1).
- range(start, stop): Generates numbers from start to (stop-1).
- range(start, stop, step): Generates numbers from start to (stop-1) with the specified step
- time.sleep(): Delays program execution for a specified number of seconds.

Nested Loop

- Nested Loops: Inner loop completes all its iterations before the outer loop.

While Loop

- while: Executes code as long as a specified condition remains true.
- Infinite Loop: A loop that runs indefinitely without a stop condition.
- += and -=: Increment and decrement operators for variables.
- not: Negates a condition (true becomes false and vice versa).

1D Lists

- append(): adds an item at the end of the list
- remove(): removes a specific item from the list
- pop(): removes the last item from the list
- insert(): adds an item at a specific index in the list
- sort(): sorts the items in the list in ascending order
- clear(): removes all items from the list
- extend(): to add elements from one list to another
- sorted(): to sort a list or tuple in ascending order

2D&3D Lists

- food[1]: retrieves the second sublist from the food list
- food[0][0]: retrieves the first item from the first sublist in the food list
- food[1][0]: retrieves the first item from the second sublist in the food list
- food[1][1]: retrieves the second item from the second sublist in the food list
- menu[1][0][1]: retrieves the second item from the first sublist in the second sublist in the menu list

Break-Continue-Pass

- break: Used to terminate a loop or a statement.
- continue: Skips the rest of the code in the loop and goes to the next iteration.
- pass: Acts as a placeholder, does nothing.

Dictionary

- [‘key’] : Retrieves the value associated with the specified key.
- get(): Returns the value associated with the specified key, or a default value if the key does not exist.
- keys(): Returns a list of all the keys in the dictionary.
- values(): Returns a list of all the values in the dictionary.
- items(): Returns a list of tuples containing all the key-value pairs in the dictionary.
- update(): Updates the dictionary with the specified key-value pairs.
- pop(): Removes the element with the specified key from the dictionary.
- clear()Removes all the elements from the dictionary.

Functions

- def: Declares and defines a custom function.
- def function_name(parameters): Defines a function with a given name and parameters.
- function_name: The name of the function you’re defining.
- parameters: Input values that can be passed to the function.

Sets

- add(): adds an element to a set
- remove(): removes an element from a set
- clear(): clears all elements from a set
- update(): adds all elements from another set to a set
- union(): returns a set containing all elements from two or more sets
- difference(): returns a set containing the difference between two or more sets
- intersection(): returns a set containing the intersection of two or more sets

Nestedfunctions

- function1(function2(function3(input(’enter a whole positive num:’))))

it is a way to call multiple functions sequentially, Starting by the Inner Function.

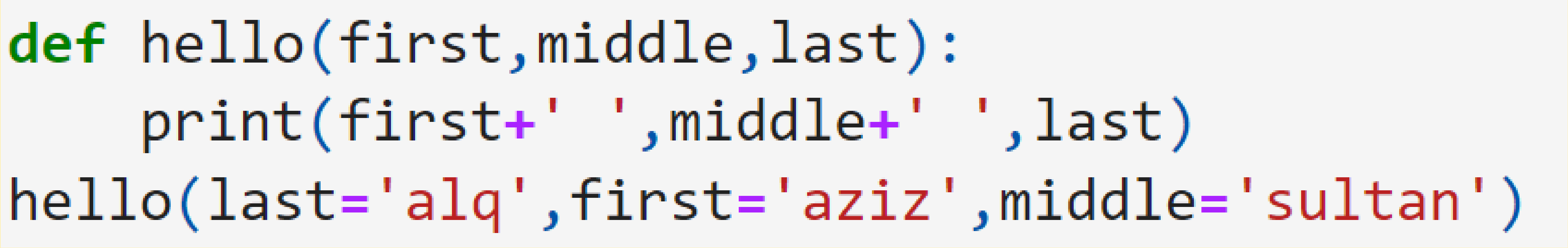
Args

- *args: The *args parameter in a function definition allows you to pass a variable number of arguments to the function. It packs all the arguments into a tuple.
- *numbers: The asterisk * before the parameter name numbers indicates that any number of arguments can be passed, and they will be collected into the tuple.

Tuples

- Tuple: Ordered, unchangeable collection for grouping related data.
- count(): Counts occurrences of an element.
- index(): Returns the index of an element.
- in Operator: Checks if an element is in the tuple.

Keyword Arguments



```python
def hello(first,middle,last):
    print(first+' ',middle+' ',last)
hello(last='alq',first='aziz',middle='sultan')
```

- hello(): A function that prints the first, middle, and last name.
- Function Parameters: Accepts three named parameters: first, middle, and last.
- Function Call: Calls the hello function with specified parameters.
- Named Arguments: Parameters are passed by name in any order.

Exception handling

- try: Executes a block of code and catches any exceptions
- except: Specifies the code to be executed if an exception occurs
- ZeroDivisionError: Handles the specific exception of dividing by zero
- Exception: Handles any other exceptions that are not specifically handled
- ValueError: Handles the specific exception of invalid input
- else: Specifies the code to be executed if no exceptions occur
- finally: Specifies the code to be executed regardless of whether an exception occurs or not

Random

- random.randint(): generates a random integer within a specified range.
- random.random(): generates a random float between 0 and 1.
- random.choice(): returns a random element from a given list.
- random.shuffle(): randomly shuffles the elements in a list.

String format

- format(): to format a string with variables
- len(): to get the length of a string, list, tuple, or dictionary
- lower(): to make all characters in a string lower case
- upper(): to make all characters in a string upper case
- strip(): to remove leading and trailing whitespace from a string
- split(): to split a string into a list
- join(): to join elements of a list into a string
- replace(): to replace a specific substring in a string
- round(): to round a number to a specified number of decimal places

Copy file

- copyfile(): copies the contents of a file.
- copy(): copies a file with permission mode and destination can be a directory.
- copy2(): copies a file with metadata.

File detection

- shutil: provides functions to work with files and directories
- os.path.exists(): checks if a path or file location exists.
- os.path.isfile(): checks if a given path is a file.

Delete file

- os.remove(): deletes a file from the file system
- os.path.exists(): checks if a file or directory exists in the file system

Write file

- open(): opens a file and returns a file object
- write(): writes text to a file
- read(): reads the contents of a file

Modules

- module: a file containing python code.
- May contain functions, classes.
- used with modular programming
- FileNotFoundError: raises an error when a file is not found

Move file

- os.path.exists(): checks if a file or directory exists
- os.replace(): replaces a file or directory
- FileNotFoundError: raises an exception when a file or directory is not found

Thank You!

Any Question?