Python methods

enumerate() → Enumerate function is used to iterate over a sequence while keeping track of the index of the current item. It returns pairs of index and corresponding element

Syntax:

enumerate(iterable, start=0)

Reduce() → It is used for cumulative computations on a list or any iterable. The f^n is repeatedly applied to the items of the iterable, reducing the iterable to a single cumulative result.

Syntax:

functools.reduce(function, iterable, initializer = None)

map() → It is used to apply a specified function to all the items in an iterable and return an iterable containing the results

Syntax:

map(function, iterable, ...)

filter() → This fn is used to construct an iterator from elements of an iterable for which a fn returns true.

Syntax:

filter(function, iterable)

zip() → zip() fn is used to combine elements from multiple iterables into tuples.

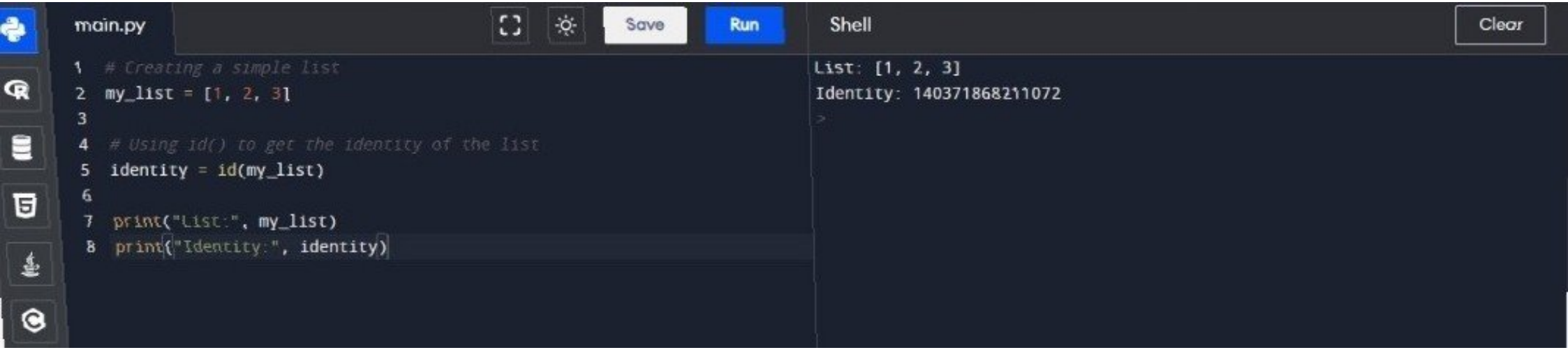
Syntax:

zip(iterable1, iterable2, ...)

id() → id() is used to get the identity of an object. this identity is a unique integer that is ~~guaranteed~~ guaranteed to be unique and constant

Syntax:

id(object)


main.py
Save
Run
Shell
Clear
1 # Creating a simple list
2 my_list = [1, 2, 3]
3
4 # Using id() to get the identity of the list
5 identity = id(my_list)
6
7 print("List:", my_list)
8 print("Identity:", identity)
List: [1, 2, 3]
Identity: 140371868211072
>

main.py

Save Run

```python
# Sample lists of names and ages
names = ["Alice", "Bob", "Charlie"]
ages = [25, 30, 22]

# Use zip to combine elements from both lists into tuples
zipped_data = zip(names, ages)

# Convert the zip object to a list for easier display
zipped_data_list = list(zipped_data)

print("Original names:", names)
print("Original ages:", ages)
print("Zipped data:", zipped_data_list)
```

Shell Clear

```
Original names: ['Alice', 'Bob', 'Charlie']
Original ages: [25, 30, 22]
Zipped data: [('Alice', 25), ('Bob', 30), ('Charlie', 22)]
>
```

main.py

Save Run

```python
# Sample list of numbers
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

# Define a function to filter out odd numbers
def is_even(x):
    return x % 2 == 0

# Use filter to get only the even numbers from the list
even_numbers = filter(is_even, numbers)

# Convert the filter object to a list for easier display
even_numbers_list = list(even_numbers)

print("Original numbers:", numbers)
print("Even numbers:", even_numbers_list)
```

Shell

Clear

```
Original numbers: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Even numbers: [2, 4, 6, 8, 10]
>
```

main.py

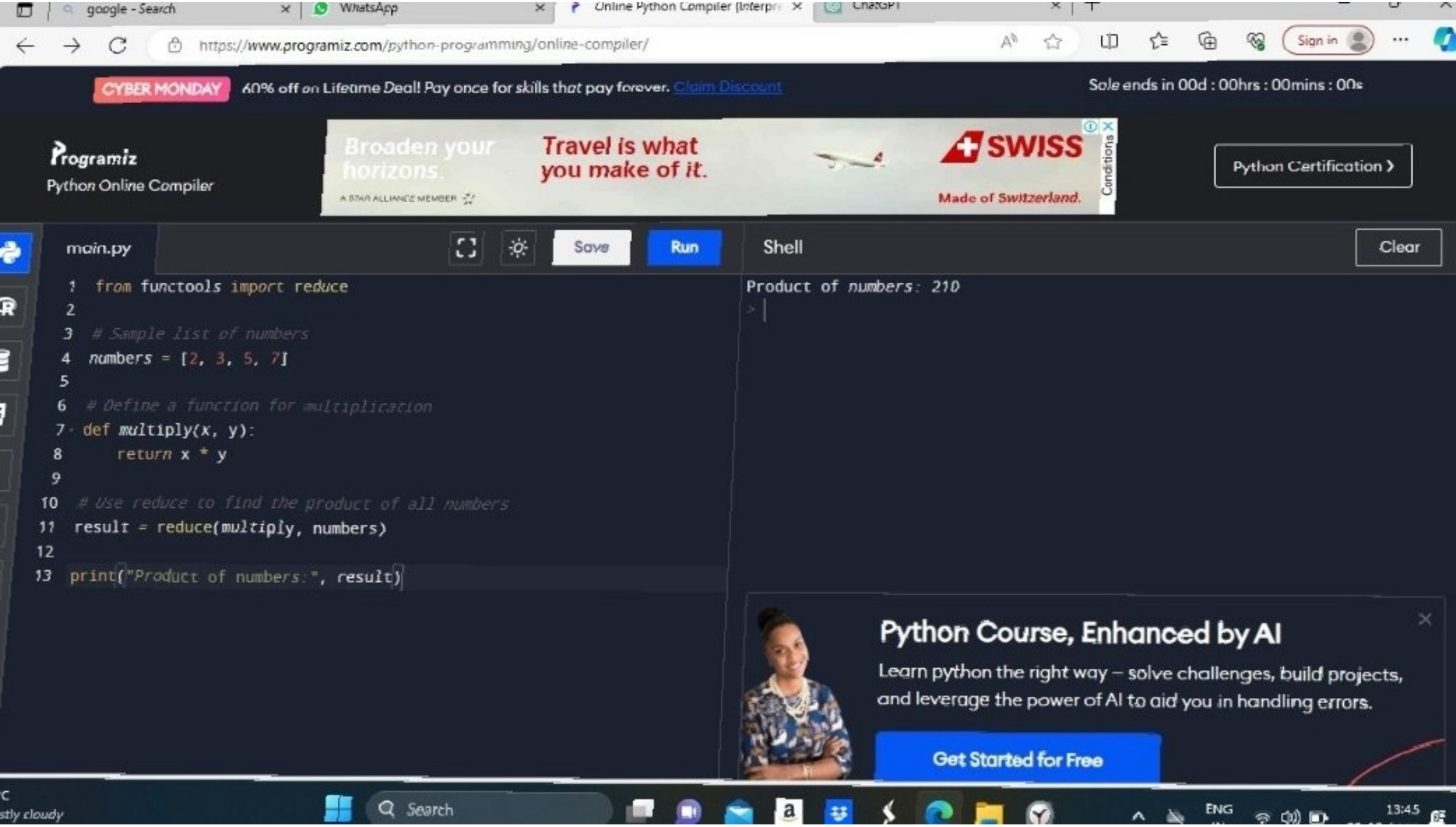
Save Run

```python
# Sample list of numbers
numbers = [1, 2, 3, 4, 5]

# Define a function to square a number
def square(x):
    return x ** 2

# Use map to apply the square function to each element in the list
squared_numbers = map(square, numbers)

# Convert the map object to a list for easier display
squared_numbers_list = list(squared_numbers)

print("Original numbers:", numbers)
print("Squared numbers:", squared_numbers_list)
```

Shell

Clear

```
Original numbers: [1, 2, 3, 4, 5]
Squared numbers: [1, 4, 9, 16, 25]
>
```


google - Search
WhatsApp
Online Python Compiler (Interpr
ChatGPT
https://www.programiz.com/python-programming/online-compiler/
Sign in
CYBER MONDAY
60% off on Lifetime Deal! Pay once for skills that pay forever. Claim Discount
Sale ends in 00d : 00hrs : 00mins : 00s
Programiz
Python Online Compiler
Broaden your horizons.
Travel is what you make of it.
A STAR ALLIANCE MEMBER
SWISS
Made of Switzerland.
Conditions
Python Certification
main.py
Save
Run
Shell
Clear
1 from functools import reduce
2
3 # Sample list of numbers
4 numbers = [2, 3, 5, 7]
5
6 # Define a function for multiplication
7 def multiply(x, y):
8 return x * y
9
10 # Use reduce to find the product of all numbers
11 result = reduce(multiply, numbers)
12
13 print("Product of numbers:", result)
Product of numbers: 210
Python Course, Enhanced by AI
Learn python the right way – solve challenges, build projects, and leverage the power of AI to aid you in handling errors.
Get Started for Free
Search
ENG
13:45

main.py

Save Run

```python
colors = ['red', 'green', 'blue', 'yellow']

# Using enumerate to iterate over the list with indices
for index, color in enumerate(colors, start=1):
    print(f"Color {index}: {color}")
```

Shell Clear

```
Color 1: red
Color 2: green
Color 3: blue
Color 4: yellow
>
```