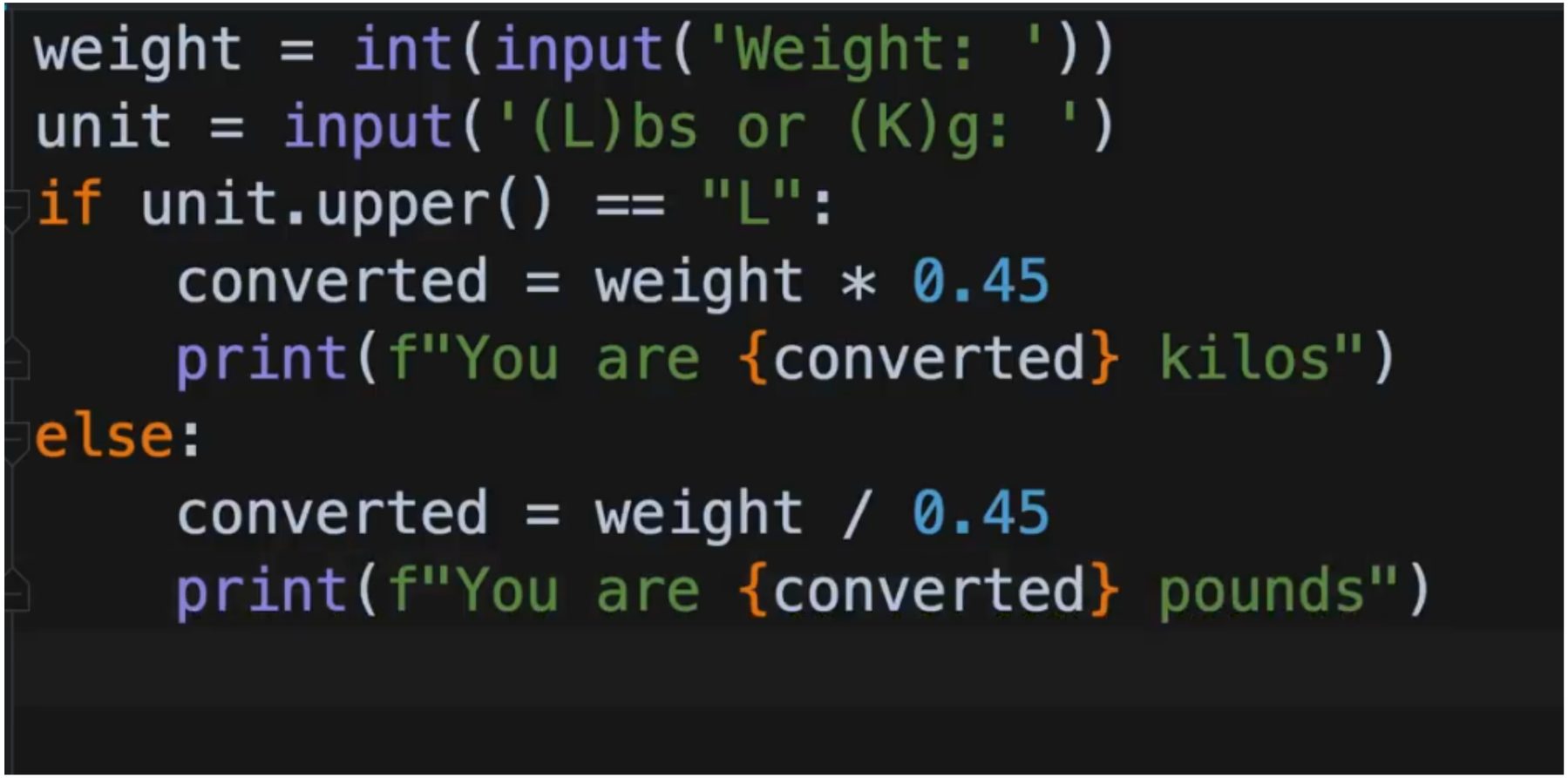
WEIGHT CONVERTER PROGRAM

```
Weight:
```

```
Weight: 72
(L)bs or (K)g: k
You are 160.0 pounds
```

```
Weight: 160
(L)bs or (K)g: l
You are 72.0 kilos
```

Solution



```
weight = int(input('Weight: '))
unit = input('(L)bs or (K)g: ')
if unit.upper() == "L":
    converted = weight * 0.45
    print(f"You are {converted} kilos")
else:
    converted = weight / 0.45
    print(f"You are {converted} pounds")
```

WHILE LOOPS

```python
i = 1
while i <= 5:
    print(i)
    i = i + 1
print("Done")
```

```python
i = 1
while i <= 5:
    print('*' * i)
    i = i + 1
print("Done")
```

BUILDING A GUESSING GAME

```
Guess:
```

```
Guess: 1
Guess: 2
Guess: 3
Sorry you failed!
```

```
Guess: 9
You win!
```

Solution

```
secret_number = 9
guess_count = 0
guess_limit = 3
while guess_count < guess_limit:
    guess = int(input('Guess: '))
    guess_count += 1
    if guess == secret_number:
        print('You won!')
        break
else:
    print('Sorry, you failed!')
```

BUILDING THE CAR GAME

```
>help
start - to start the car
stop - to stop the car
quit - to exit
```

```
>asd
I don't understand that...
>start
Car started...Ready to go!
>stop
Car stopped.
>quit
```

Solution

```
command = ""

while True:
    command = input("> ").lower()
    if command == "start":
        print("Car started ..."   )
    elif command == "stop":
        print("Car stopped.")
    elif command == "help":
        print("""
start - to start the car
stop - to stop the car
quit - to quit
                """)
    elif command == "quit":
        break
    else:
        print("Sorry, I don't understand that !")
```

FOR LOOPS

```python
for item in 'Python':
    print(item)
```

```python
for item in ['Mosh', 'John', 'Sarah']:
    print(item)
```

```python
for item in [1, 2, 3, 4]:
    print(item)
```

```
for item in [1, 2, 3, 4, 5, 6, 7, ]:
    print(item)
```

```
for item in range(10):
    print(item)
```

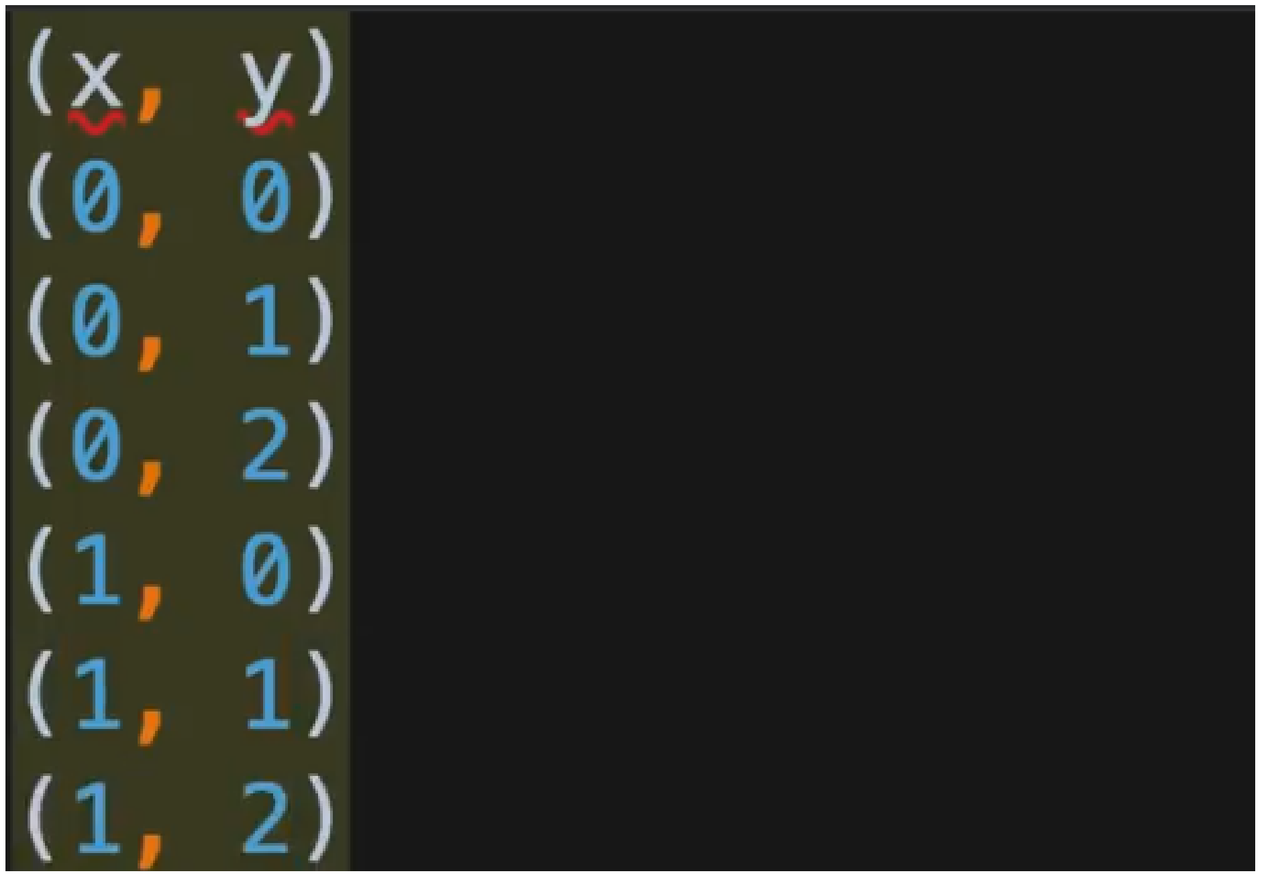
```
for item in range(5, 10):
    print(item)
```

```
for item in range(5, 10, 2):
    print(item)
```

```
prices = [10, 20, 30]
```

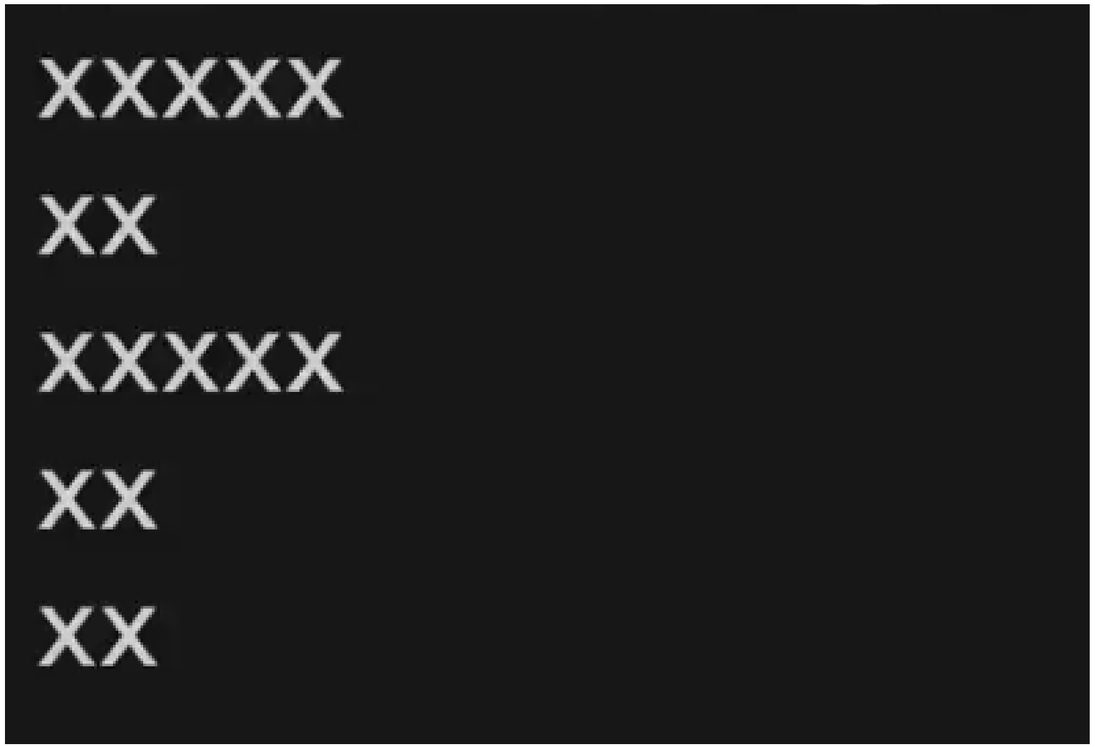
```python
prices = [10, 20, 30]

total = 0
for price in prices:
    total += price
print(f"Total: {total}")
```

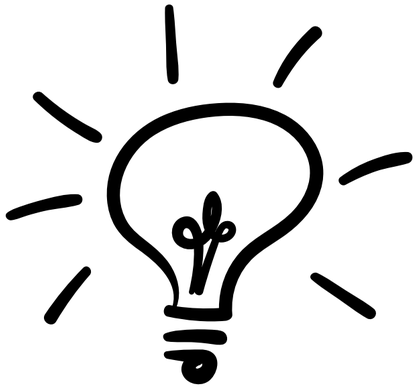
NESTED LOOPS


(x, y)
(0, 0)
(0, 1)
(0, 2)
(1, 0)
(1, 1)
(1, 2)

```
for x in range(4):
    for y in range(3):
        print(f'({x}, {y})')
```

Challenge


XXXXX
XX
XXXXX
XX
XX



```
numbers = [5, 2, 5, 2, 2]
```

```
numbers = [5, 2, 5, 2, 2]
for x_count in numbers:
    print('x' * x_count)
```

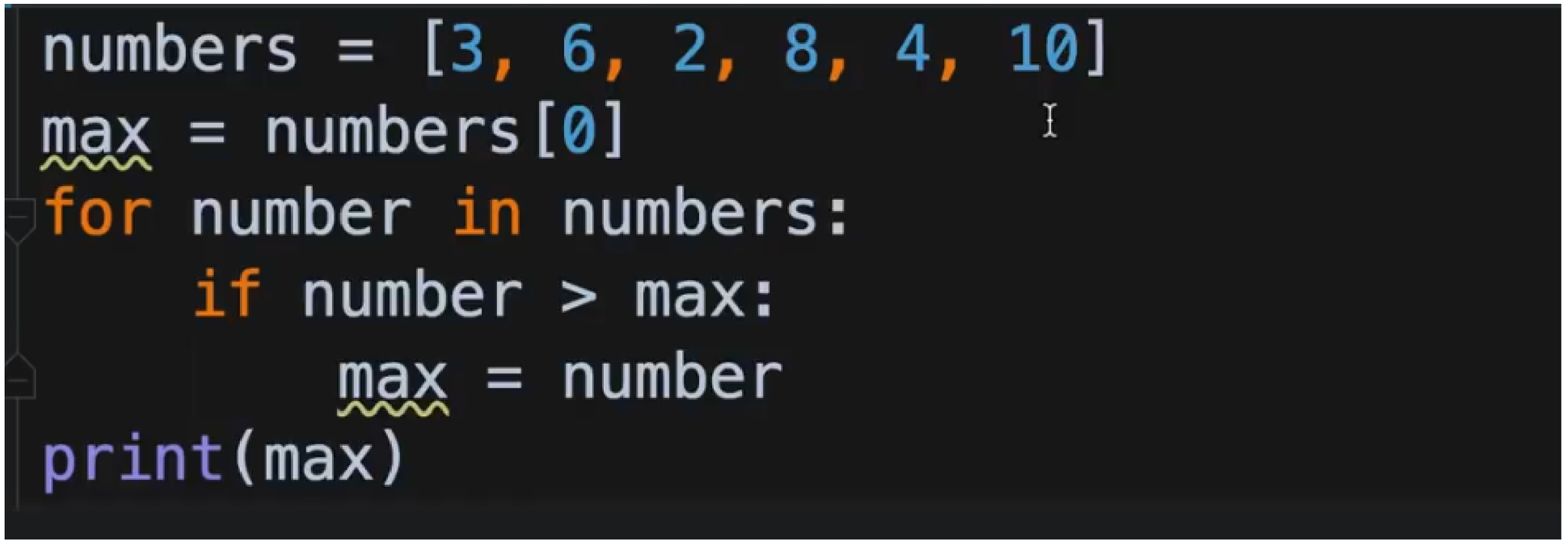
```
numbers = [5, 2, 5, 2, 2]
for x_count in numbers:
    output = ''
    for count in range(x_count):
        output += 'x'
    print(output)
```

LISTS

```
numbers = [1, 2, 3, 4, 5]
numbers[0]              # returns the first item
numbers[1]              # returns the second item
numbers[-1]             # returns the first item from the end
numbers[-2]             # returns the second item from the end
```

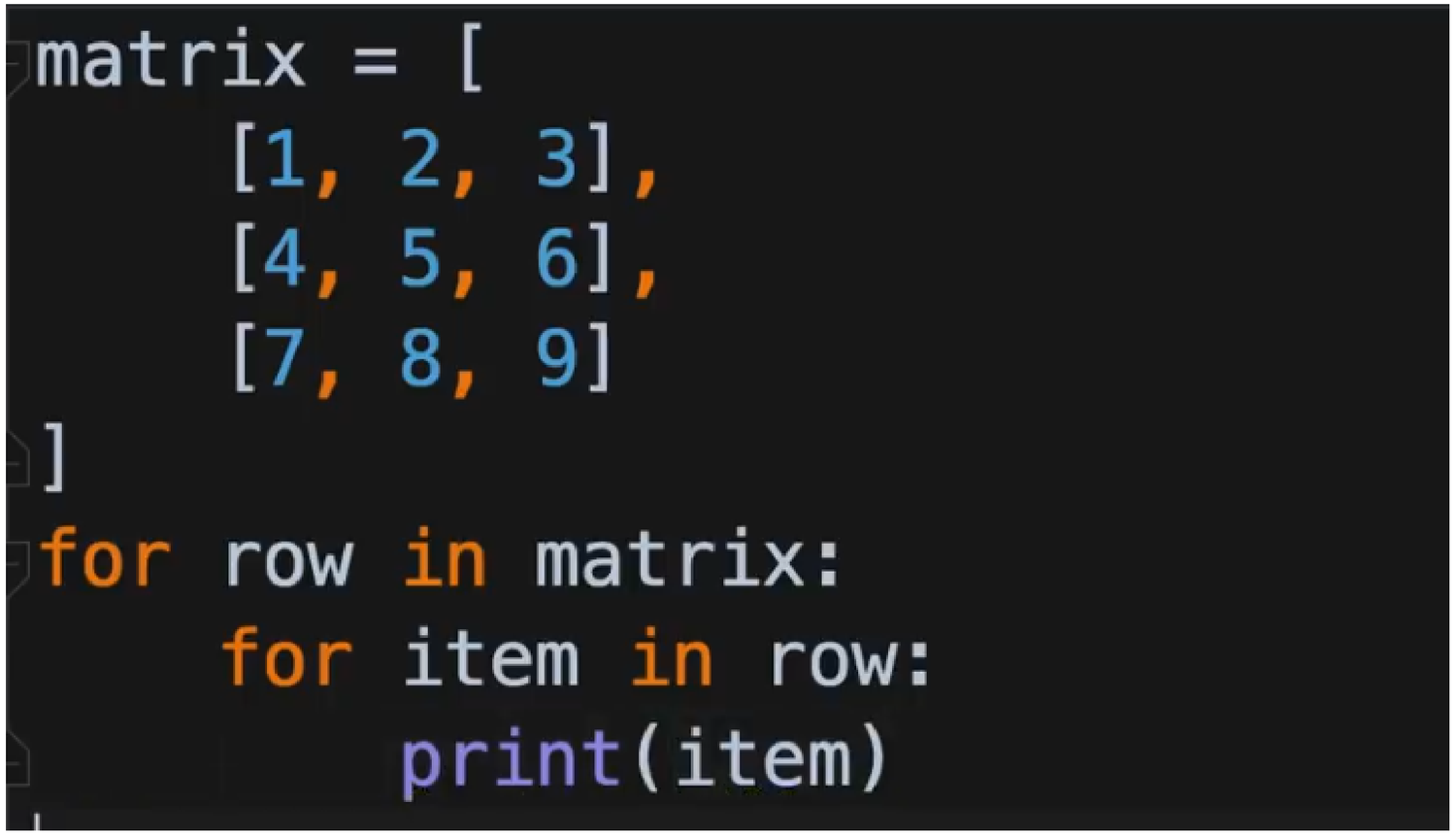
Write a program to find the largest number in a list.

Solution


numbers = [3, 6, 2, 8, 4, 10]
max = numbers[0]
for number in numbers:
if number > max:
max = number
print(max)

2D LISTS


[
1 2 3
4 5 6
7 8 9
]


matrix = [
[1, 2, 3],
[4, 5, 6],
[7, 8, 9]
]
for row in matrix:
for item in row:
print(item)

LIST METHODS

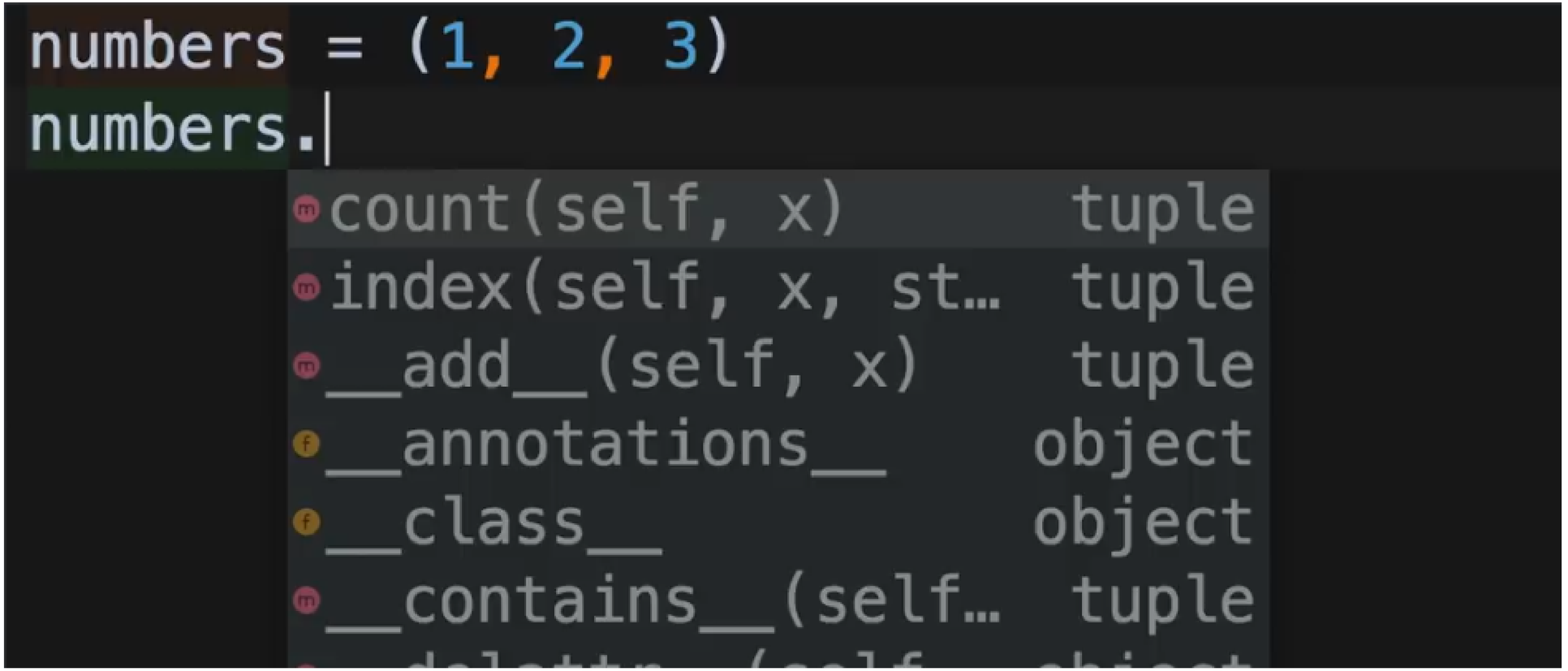
```
numbers.append(6)     # adds 6 to the end
numbers.insert(0, 6) # adds 6 at index position of 0
numbers.remove(6)     # removes 6
numbers.pop()         # removes the last item
numbers.clear()       # removes all the items
numbers.index(8)      # returns the index of first occurrence of 8
numbers.sort()        # sorts the list
numbers.reverse()     # reverses the list
numbers.copy()        # returns a copy of the list
```

Write a program to remove the duplicates in a list.

Solution

```python
numbers = [2, 2, 4, 6, 3, 4, 6, 1]
uniques = []
for number in numbers:
    if number not in uniques:
        uniques.append(number)
print(uniques)
```

TUPLES


numbers = (1, 2, 3)
numbers.
count(self, x) tuple
index(self, x, st… tuple
__add__(self, x) tuple
__annotations__ object
__class__ object
__contains__(self… tuple

```python
numbers = (1, 2, 3)
numbers[0] = 10
print(numbers[0])
```

```
object does not support item assignment
```

UNPACKING

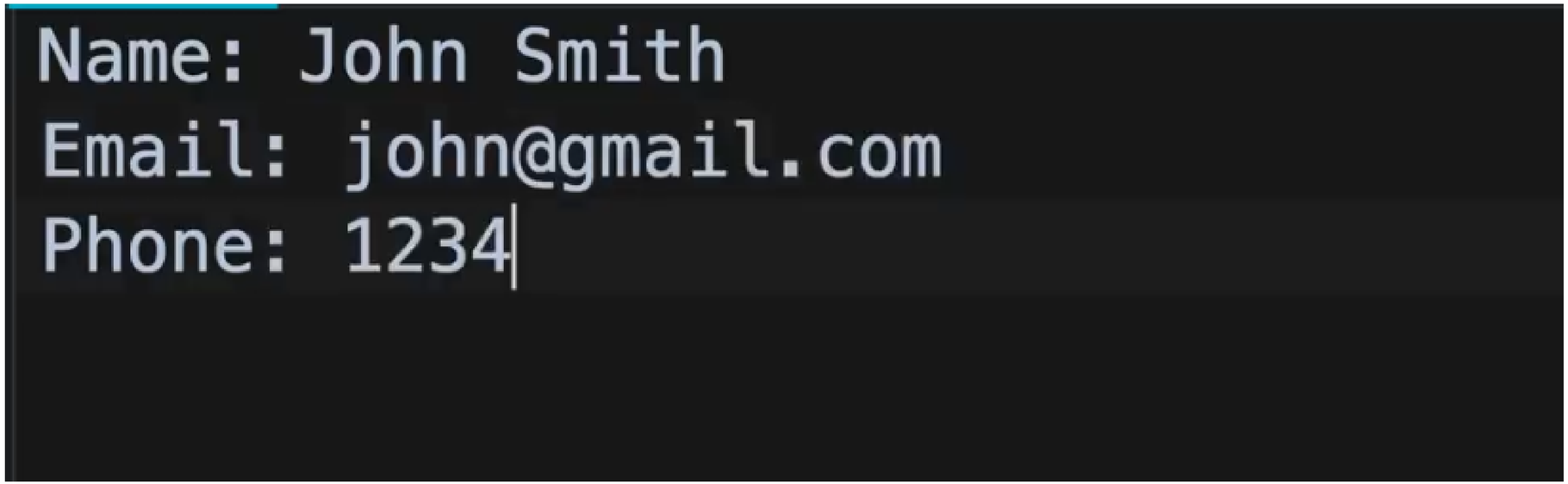
```
coordinates = (1, 2, 3)
coordinates[0] * coordinates[1] * coordinate
```

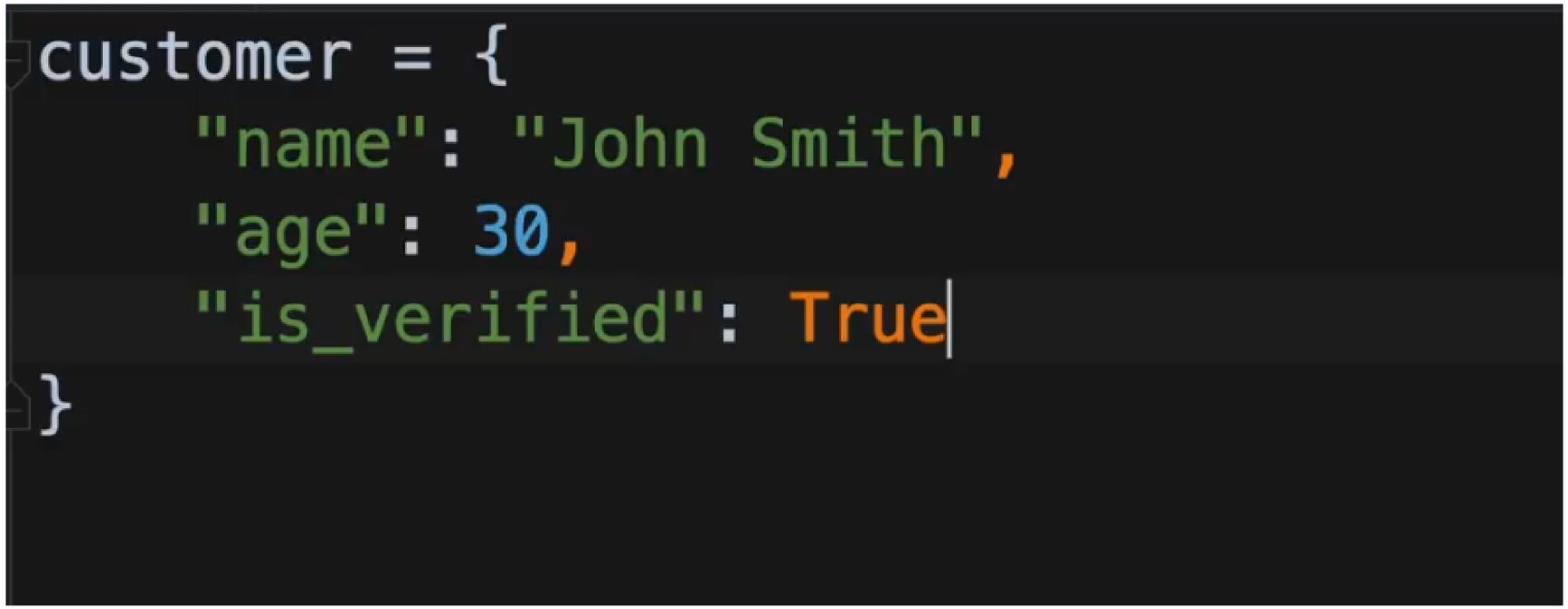
```
coordinates = (1, 2, 3)
x = coordinates[0]
y = coordinates[1]
z = coordinates[2]

x * y * z
```

```
coordinates = (1, 2, 3)
x, y, z = coordinates
```

```
coordinates = [1, 2, 3]
x, y, z = coordinates
```

DICTIONARIES


Name: John Smith
Email: john@gmail.com
Phone: 1234


customer = {
"name": "John Smith",
"age": 30,
"is_verified": True
}

```
customer = {
    "name": "John Smith",
    "age": 30,
    "is_verified": True
}
print(customer["Name"])
```

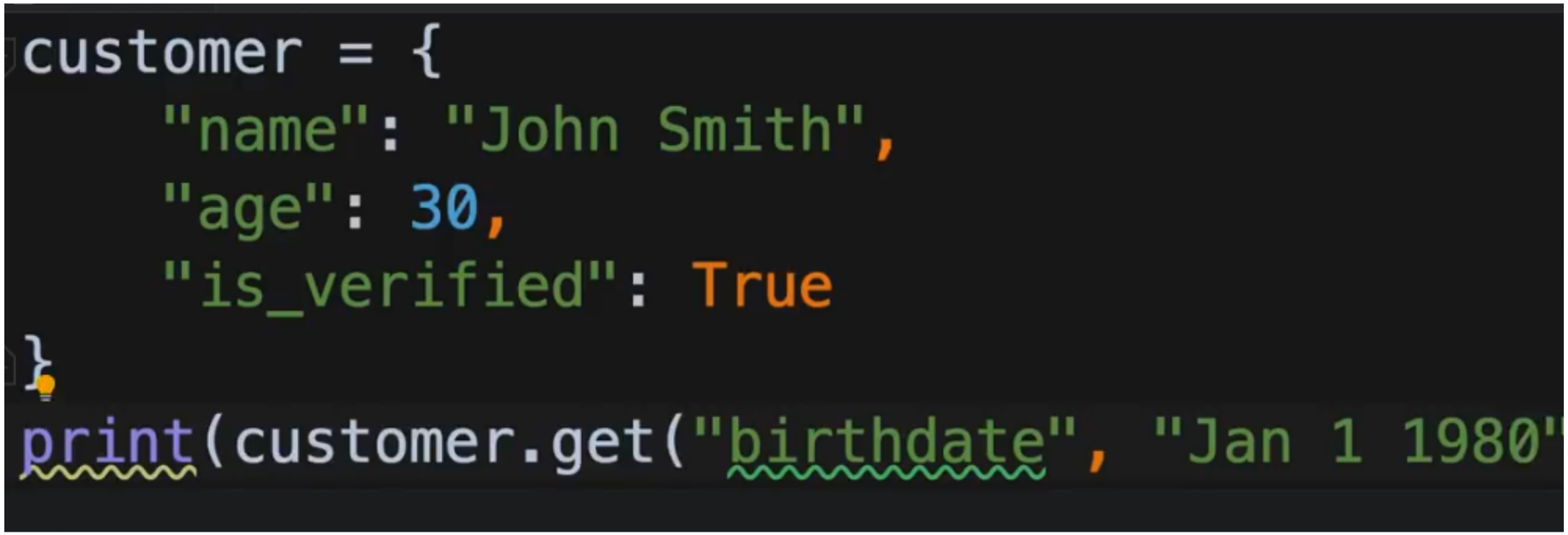
```
KeyError: 'Name'
```

```
print(customer["birthdate"])
```

```
KeyError: 'birthdate'
```

```
customer = {
    "name": "John Smith",
    "age": 30,
    "is_verified": True
}
print(customer.get("birthdate"))
```

```
None
```


customer = {
"name": "John Smith",
"age": 30,
"is_verified": True
}
print(customer.get("birthdate", "Jan 1 1980"

```
Phone:
```

```
Phone: 1234
One Two Three Four
```

Solution

```python
phone = input("Phone: ")
digits_mapping = {
        "1": "One",
        "2": "Two",
        "3": "Three",
        "4": "Four",
        "5": "Five",
        "6": "Six",
        "7": "Seven",
        "8": "Eight",
        "9": "Nine",
        "0": "Zero"
        }

output = ""
for ch in phone:
    output += digits_mapping.get(ch, "!")+ " "
print(output)
```

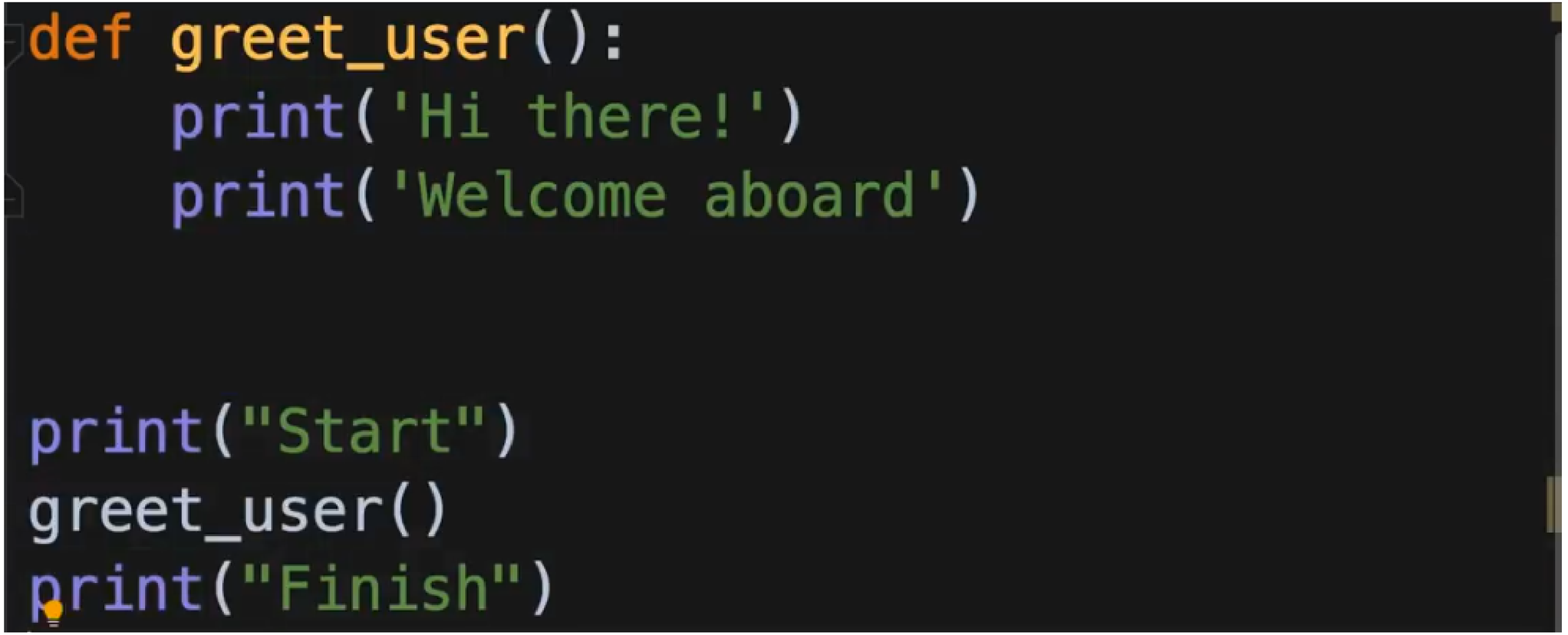
EMOJI CONVERTOR


>Good morning :)
Good morning 😀


>I am sad :(
I am sad 😞

```
message = input(">")
words = message.split(' ')
emojis = {
    ":)": "😀",
    ":(": "😞"
}
output = ""
for word in words:
    output += emojis.get(word, word) + " "
print(output)
```

FUNCTIONS


def greet_user():
print('Hi there!')
print('Welcome aboard')
print("Start")
greet_user()
print("Finish")


greet_user()
Unresolved reference 'greet_user' more... (⌘F1)
def greet_user():
print('Hi there!')
print('Welcome aboard')
print("Start")
greet_user()
print("Finish")

PARAMETERS


def greet_user(name):
print(f'Hi {name}!')
print('Welcome aboard')
print("Start")
greet_user("John")
greet_user("Mary")
print("Finish")

```python
def greet_user(first_name, last_name):
    print(f'Hi {first_name} {last_name}!')
    print('Welcome aboard')


print("Start")
greet_user("John", "Smith")
print("Finish")
```

KEYWORD ARGUMENT

```python
def greet_user(first_name, last_name):
    print(f'Hi {first_name} {last_name}!')
    print('Welcome aboard')


print("Start")
greet_user("Smith", "John")
print("Finish")
```


def greet_user(first_name, last_name):
print(f'Hi {first_name} {last_name}!')
print('Welcome aboard')
print("Start")
greet_user(last_name="Smith", first_name="J
print("Finish")

```
cost(50, 5, 0.1)
```

```
cost(total=50, shipping=5, discount=0.1)
```

RETURN STATEMENT

```
def square(number):
    return number * number
```

```python
def square(number):
    print(number * number)


print(square(3))
```

```
9
None
```

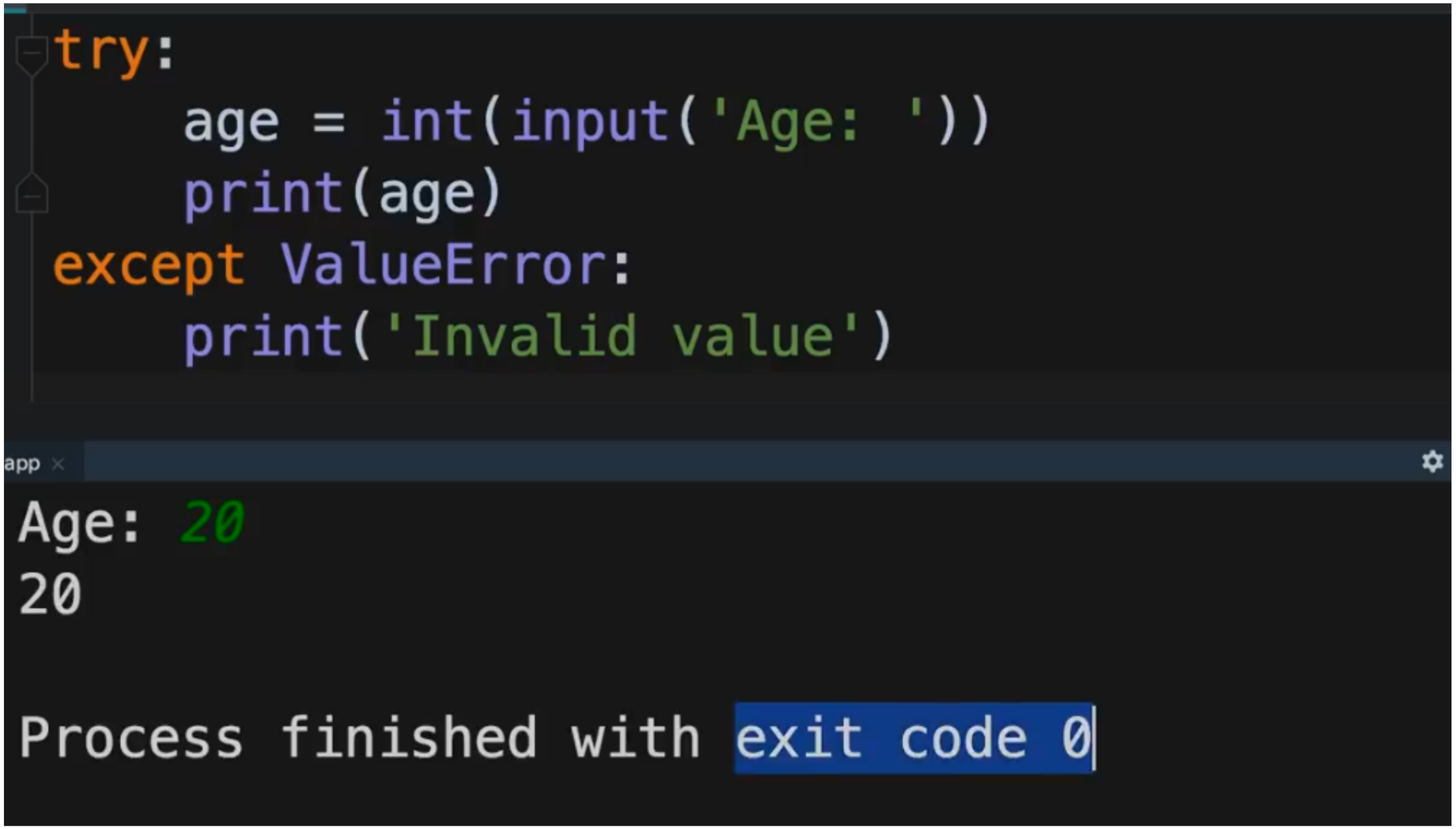
```python
def square(number):
    print(number * number)
    return None
```

```
9
None
```

CREATING A REUSABLE FUNCTION

```python
phone = input("Phone: ")

def convertor(phone_no):
    digits_mapping = {
          "1": "One",
          "2": "Two",
          "3": "Three",
          "4": "Four",
          "5": "Five",
          "6": "Six",
          "7": "Seven",
          "8": "Eight",
          "9": "Nine",
          "0": "Zero"
          }

    output = ""
    for ch in phone:
        output += digits_mapping.get(ch, "!")+ " "
    return output

print(convertor(phone))
```

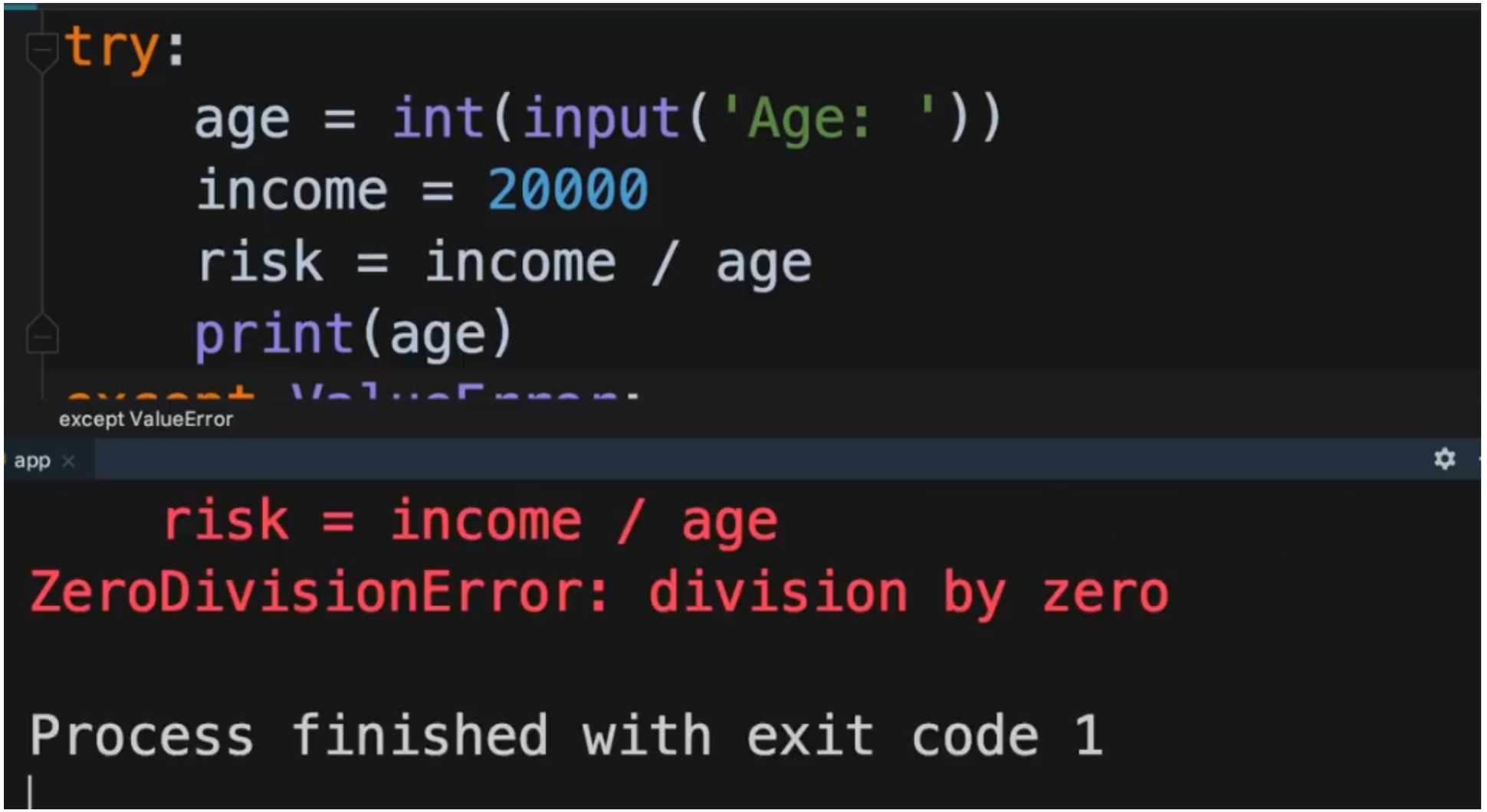
EXCEPTIONS


try:
age = int(input('Age: '))
print(age)
except ValueError:
print('Invalid value')
app
Age: 20
20
Process finished with exit code 0

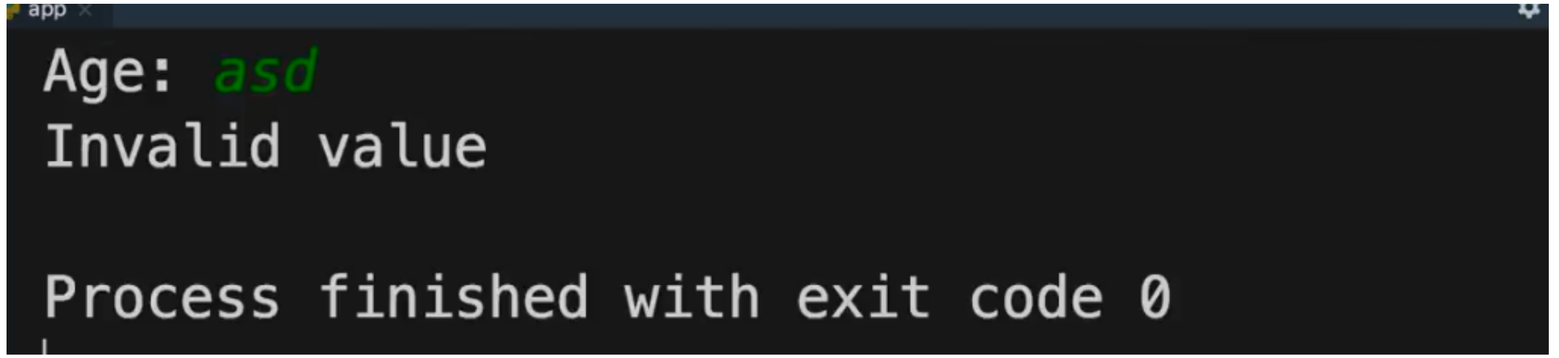
```
Age: asd
```

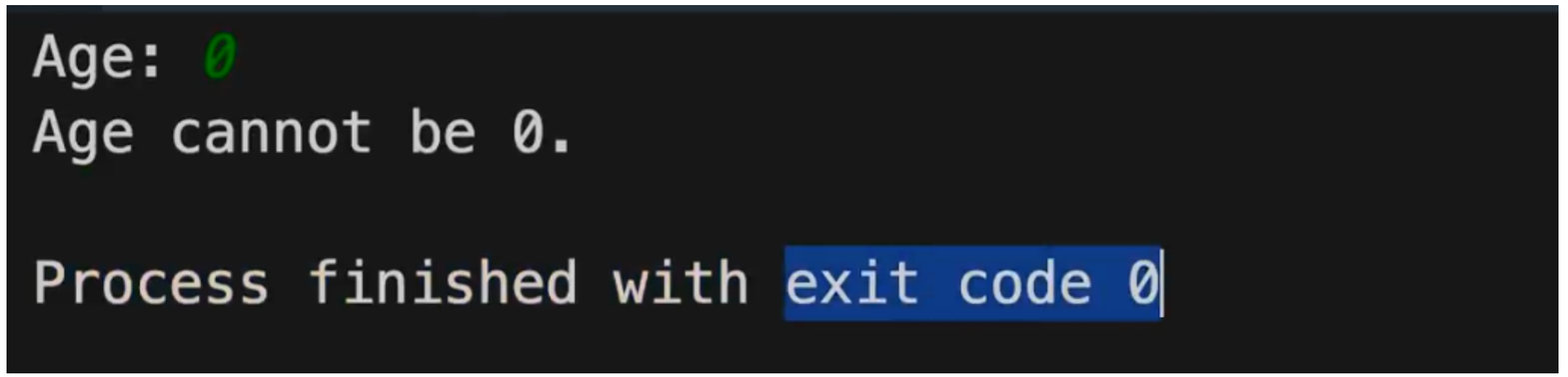
```
    age = int(input('Age: '))
ValueError: invalid literal for int() with ba

Process finished with exit code 1
```

```
try:
    age = int(input('Age: '))
    income = 20000
    risk = income / age
    print(age)
except ValueError:
    print('Invalid value')
```


try:
age = int(input('Age: '))
income = 20000
risk = income / age
print(age)
except ValueError
app
risk = income / age
ZeroDivisionError: division by zero
Process finished with exit code 1

```
try:
    age = int(input('Age: '))
    income = 20000
    risk = income / age
    print(age)
except ZeroDivisionError:
    print('Age cannot be 0.')
except ValueError:
    print('Invalid value')
```


app
Age: asd
Invalid value
Process finished with exit code 0


Age: 0
Age cannot be 0.
Process finished with exit code 0

COMMENTS

```python
# asdlkjasdlkajsld
print("Sky is blue")
```

```python
# print Sky is blue
print("Sky is blue")
```

```python
# print Ocean is blue
print("Ocean is blue")

# Calculates and returns the square
def square(number):
    return number * number
```