

Python

2023-12-26

Contents

| | | |
|----------|--|----------|
| 1 | VARIABLES AND SIMPLE DATA TYPES | 2 |
| 2 | INTRODUCING LISTS | 3 |
| 3 | WORKING WITH LISTS | 5 |
| 4 | IF STATEMENTS | 7 |
| 5 | DICTIONARIES | 9 |

1 VARIABLES AND SIMPLE DATA TYPES

- Changing Case in a String with Methods

```
# Escribe la primera letra de cada palabra en mayúscula
name = "ada lovelace"
print(name.title())
```

```
## Ada Lovelace
```

```
# Escribe toda la palabra en mayúsculas
print(name.upper())
```

```
## ADA LOVELACE
```

```
# Escribe toda la palabra en minúsculas
print(name.lower())
```

```
## ada lovelace
```

- Using Variables in Strings

```
first_name = "ada"
last_name = "lovelace"
full_name = f"{first_name} {last_name}"
print(full_name)
```

```
## ada lovelace
```

- Adding Whitespace to Strings with Tabs or Newlines

```
print("Languages:\n\tPython\n\tC\n\tJavaScript")
```

```
## Languages:
```

```
## Python
```

```
## C
```

```
## JavaScript
```

- Stripping Whitespace

```
favorite_language = ' python '
```

```
favorite_language.rstrip()
```

```
## ' python'
```

```
favorite_language.lstrip()
```

```
## 'python '
```

```
favorite_language.strip()
```

```
## 'python'
```

- Removing Prefixes

```
nostarch_url = 'https://nostarch.com'
```

```
nostarch_url.removeprefix('https://')
```

```
## 'nostarch.com'
```

- Underscores in Numbers

```
universe_age = 14_000_000_000
```

```
print(universe_age)
```

```
## 14000000000
```

- Multiple Assignment

```
x, y, z = 0, 0, 0
```

2 INTRODUCING LISTS

```
bicycles = ['trek', 'cannondale', 'redline', 'specialized']  
print(bicycles)
```

```
## ['trek', 'cannondale', 'redline', 'specialized']
```

- Accessing Elements in a List

```
print(bicycles[0].title())
```

```
## Trek
```

Python has a special syntax for accessing the last element in a list. If you ask for the item at index -1, Python always returns the last item in the list:

```
print(bicycles[-1])
```

```
## specialized
```

- Using Individual Values from a List

```
message = f"My first bicycle was a {bicycles[0].title()}."  
print(message)
```

```
## My first bicycle was a Trek.
```

- Modifying Elements in a List

```
motorcycles = ['honda', 'yamaha', 'suzuki']  
print(motorcycles)
```

```
## ['honda', 'yamaha', 'suzuki']
```

```
motorcycles[0] = 'ducati'  
print(motorcycles)
```

```
## ['ducati', 'yamaha', 'suzuki']
```

- Adding Elements to a List

```
motorcycles = ['honda', 'yamaha', 'suzuki']  
print(motorcycles)
```

```
## ['honda', 'yamaha', 'suzuki']
```

```
motorcycles.append('ducati')  
print(motorcycles)
```

```
## ['honda', 'yamaha', 'suzuki', 'ducati']
```

- Inserting Elements into a List

```
motorcycles = ['honda', 'yamaha', 'suzuki']  
motorcycles.insert(0, 'ducati')  
print(motorcycles)
```

```
## ['ducati', 'honda', 'yamaha', 'suzuki']
```

- Removing an Item Using the del Statement

```
motorcycles = ['honda', 'yamaha', 'suzuki']  
print(motorcycles)
```

```
## ['honda', 'yamaha', 'suzuki']
```

```
del motorcycles[0]  
print(motorcycles)
```

```
## ['yamaha', 'suzuki']
```

- Removing an Item Using the pop() Method

```
motorcycles = ['honda', 'yamaha', 'suzuki']  
print(motorcycles)
```

```
## ['honda', 'yamaha', 'suzuki']  
popped_motorcycle = motorcycles.pop()  
print(motorcycles)
```

```
## ['honda', 'yamaha']  
print(popped_motorcycle)
```

```
## suzuki
```

- Popping Items from Any Position in a List

```
motorcycles = ['honda', 'yamaha', 'suzuki']  
first_owned = motorcycles.pop(0)  
print(f"The first motorcycle I owned was a {first_owned.title()}.")
```

```
## The first motorcycle I owned was a Honda.
```

- Removing an Item by Value

```
motorcycles = ['honda', 'yamaha', 'suzuki', 'ducati']  
print(motorcycles)
```

```
## ['honda', 'yamaha', 'suzuki', 'ducati']  
motorcycles.remove('ducati')  
print(motorcycles)
```

```
## ['honda', 'yamaha', 'suzuki']
```

- Sorting a List Permanently with the sort() Method

```
cars = ['bmw', 'audi', 'toyota', 'subaru']  
cars.sort()  
print(cars)
```

```
## ['audi', 'bmw', 'subaru', 'toyota']
```

- Sorting a List Temporarily with the sorted() Function

```
cars = ['bmw', 'audi', 'toyota', 'subaru']  
print("Here is the original list:")
```

```
## Here is the original list:
```

```

print(cars)

## ['bmw', 'audi', 'toyota', 'subaru']
print("\nHere is the sorted list:")

##
## Here is the sorted list:
print(sorted(cars))

## ['audi', 'bmw', 'subaru', 'toyota']
print("\nHere is the original list again:")

##
## Here is the original list again:
print(cars)

## ['bmw', 'audi', 'toyota', 'subaru']

```

- Printing a List in Reverse Order

```

cars = ['bmw', 'audi', 'toyota', 'subaru']
print(cars)

## ['bmw', 'audi', 'toyota', 'subaru']
cars.reverse()
print(cars)

## ['subaru', 'toyota', 'audi', 'bmw']

```
- Finding the Length of a List

```

cars = ['bmw', 'audi', 'toyota', 'subaru']
len(cars)

## 4

```

3 WORKING WITH LISTS

- Looping Through an Entire List

```

magicians = ['alice', 'david', 'carolina']
for magician in magicians:
    print(magician)

## alice
## david
## carolina

```
- Using the range() Function

```

for value in range(1, 5):
    print(value)

## 1
## 2
## 3

```

```
## 4
```

- Using range() to Make a List of Numbers

```
numbers = list(range(1, 6))  
print(numbers)
```

```
## [1, 2, 3, 4, 5]
```

```
even_numbers = list(range(2, 11, 2))  
print(even_numbers)
```

```
## [2, 4, 6, 8, 10]
```

```
squares = []  
for value in range(1,11):  
    squares.append(value**2)  
print(squares)
```

```
## [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

- Simple Statistics with a List of Numbers

```
digits = [1, 2, 3, 4, 5, 6, 7, 8, 9, 0]  
min(digits)
```

```
## 0
```

```
max(digits)
```

```
## 9
```

```
sum(digits)
```

```
## 45
```

- List Comprehensions

```
squares = [value**2 for value in range(1, 11)]  
print(squares)
```

```
## [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

- Slicing a List

```
players = ['charles', 'martina', 'michael', 'florence', 'eli']  
print(players[0:3])
```

```
## ['charles', 'martina', 'michael']
```

```
print(players[:4])
```

```
## ['charles', 'martina', 'michael', 'florence']
```

```
print(players[2:])
```

```
## ['michael', 'florence', 'eli']
```

```
# con el signo negativo cuenta desde el final  
print(players[-3:])
```

```
## ['michael', 'florence', 'eli']
```

- Looping Through a Slice

```
print("Here are the first three players on my team:")
```

```
## Here are the first three players on my team:
```

```
for player in players[:3]:  
    print(player.title())
```

```
## Charles
```

```
## Martina
```

```
## Michael
```

- Copying a List

```
my_foods = ['pizza', 'falafel', 'carrot cake']  
friend_foods = my_foods[:]  
print("My favorite foods are:")
```

```
## My favorite foods are:
```

```
print(my_foods)
```

```
## ['pizza', 'falafel', 'carrot cake']
```

```
print("\nMy friend's favorite foods are:")
```

```
##
```

```
## My friend's favorite foods are:
```

```
print(friend_foods)
```

```
## ['pizza', 'falafel', 'carrot cake']
```

- Defining a Tuple Las tuplas son como las listas, pero no se pueden modificar. Si queremos cambiar una tupla tenemos que redefinirla.

```
dimensions = (200, 50)  
print(dimensions[0])
```

```
## 200
```

```
print(dimensions[1])
```

```
## 50
```

4 IF STATEMENTS

- A Simple Example

```
cars = ['audi', 'bmw', 'subaru', 'toyota']  
for car in cars:  
    if car == 'bmw':  
        print(car.upper())  
    else:  
        print(car.title())
```

```
## Audi
```

```
## BMW
```

```
## Subaru
```

```
## Toyota
```

- Checking for Inequality

```
requested_topping = 'mushrooms'
if requested_topping != 'anchovies':
    print("Hold the anchovies!")
```

```
## Hold the anchovies!
```

- Numerical Comparisons

```
answer = 17
if answer != 42:
    print("That is not the correct answer. Please try again!")
```

```
## That is not the correct answer. Please try again!
```

- Checking Whether a Value Is in a List

```
requested_toppings = ['mushrooms', 'onions', 'pineapple']
'mushrooms' in requested_toppings
```

```
## True
```

```
'pepperoni' in requested_toppings
```

```
## False
```

- Checking Whether a Value Is Not in a List

```
banned_users = ['andrew', 'carolina', 'david']
user = 'marie'
if user not in banned_users:
    print(f"{user.title()}, you can post a response if you wish.")
```

```
## Marie, you can post a response if you wish.
```

- Testing Multiple Conditions The if- elif- else block would stop running after only one test passes.
- Using if Statements with Lists

```
requested_toppings = ['mushrooms', 'green peppers', 'extra cheese']
for requested_topping in requested_toppings:
    print(f"Adding {requested_topping}.")
```

```
## Adding mushrooms.
```

```
## Adding green peppers.
```

```
## Adding extra cheese.
```

```
print("\nFinished making your pizza!")
```

```
##
```

```
## Finished making your pizza!
```

```
for requested_topping in requested_toppings:
    if requested_topping == 'green peppers':
        print("Sorry, we are out of green peppers right now.")
    else:
        print(f"Adding {requested_topping}.")
```

```
## Adding mushrooms.
```

```
## Sorry, we are out of green peppers right now.
```

```
## Adding extra cheese.
```



```
print("\nFinished making your pizza!")
```

```
##
```

```
## Finished making your pizza!
```

- Checking That a List Is Not Empty

```
requested_toppings = []  
if requested_toppings:  
    for requested_topping in requested_toppings:  
        print(f"Adding {requested_topping}.")  
        print("\nFinished making your pizza!")  
else:  
    print("Are you sure you want a plain pizza?")
```

```
## Are you sure you want a plain pizza?
```

- Using Multiple Lists

```
available_toppings = ['mushrooms', 'olives', 'green peppers',  
                      'pepperoni', 'pineapple', 'extra cheese']  
requested_toppings = ['mushrooms', 'french fries', 'extra cheese']  
for requested_topping in requested_toppings:  
    if requested_topping in available_toppings:  
        print(f"Adding {requested_topping}.")  
    else:  
        print(f"Sorry, we don't have {requested_topping}.")
```

```
## Adding mushrooms.
```

```
## Sorry, we don't have french fries.
```

```
## Adding extra cheese.
```

```
print("\nFinished making your pizza!")
```

```
##
```

```
## Finished making your pizza!
```

5 DICTIONARIES