

# Tuples

## Textbook

# Tuples



Remember that Python groups information in 4 ways: [Lists](#), [Dictionaries](#), [Tuples](#), and [Sets](#).

[Lists](#): We have learned about lists in previous lessons.

```
holidays = ["Christmas", "Hannukah", "Thanksgiving", "Halloween"]
```

[Dictionaries](#): We have learned about dictionaries in previous lessons.

```
limbs = { toes : 10, feet : 2, hairs : 1000 }
```

[Tuples](#): We will be learning more about tuples in this lesson.

```
instruments = ("clarinet", "piano", "drum", "violin")
```

[Sets](#): We will be learning more about sets in the next lesson.

```
colors = {"orange", "green", "purple", "yellow"}
```

You can learn about the differences between the data sets below.

Group Type	Symbol	Are the items ordered?	Can they be changed?	Can it have duplicate items?
List	[ ]	Yes	Yes	Yes
Dictionary	{ : }	Yes	Yes	No
Tuple	( )	No	No ( <i>immutable</i> )	Yes
Set	{ }	No	No ( <i>immutable</i> )	No

Each of these group types are useful in different situations.

## Tuples

The main difference about tuples is that they are **unchangeable**.

Below is an example of a tuple:

```
1 instruments = ("clarinet", "piano", "drum", "violin")
2
3 print(instruments)
```

Try it!

This will print out `('clarinet', 'piano', 'drum', 'violin')`. It may look a lot like a list, but note the use of parenthesis ( ) instead of square brackets.

## Accessing Tuple Items

```
1 instruments = ("clarinet", "piano", "drum", "violin")
2
3 print(instruments[1])
```

Try it!

This will print out `piano`. Remember with indexing we start counting at 0.

## Negative Indexing

You can also start counting indexing from the end of your tuple. This starts counting at -1.

```
1 instruments = ("clarinet", "piano", "drum", "violin")
2
3 print(instruments[-1])
```

Try it!

This will print out `violin`.

Note: For indexing, at the beginning we start counting at 0, but from the end we start counting at -1.



## Range of Indexes

If you want to access multiple values that appear right in a row, you can print out a [range](#) of indexes with one simple command.

```
1 instruments = ("clarinet", "piano", "drum", "violin", "guitar")
2
3 print(instruments[1:3])
```

Try it!

This will print out `('piano', 'drum')`. Notice that the first number is where you start your range and the second number is where you end your range.

Play around with the index numbers and see if you can print a range of indexes using negative numbers.

## Slicing the Tuple

Although the content of the tuple cannot be changed, it can be sliced. Here is an example.

```
1 instruments = ("clarinet", "piano", "drum", "violin")
2
3 print(instruments[0:1])
```

Try it!

This will print out

```
('clarinet',)
```

## Length of the Tuple

You can find out how many items a tuple has by using the `len()` method.

```
1 instruments = ("clarinet", "piano", "drum", "violin")
2
3
4 print(len(instruments))
```

Try it!

This will print out `4`.

## Concatenating Tuples

Although the items inside a tuple cannot change, they can be concatenated together.

```
1 instruments = ("clarinet", "piano", "drum", "violin")
2
3 hobbies = ("swimming", "drawing")
4
5
6 print(hobbies + instruments)
```

Try it!

This will print out

```
('swimming', 'drawing', 'clarinet', 'piano', 'drum', 'violin')
```

## Change Tuple Items

A characteristic of tuples is that they are unchangeable. You cannot alter the tuple items.

## Remove from the Tuple

A characteristic of tuples is that they are unchangeable. You cannot alter the tuple items.

## Check if an Item exists in the tuple.

Sometimes if you have a very large tuple with hundreds of items inside, it can be useful to check if a certain item can be found inside the tuple.

```
1 instruments = ("clarinet", "piano", "drum", "violin", "guitar")
2
3 if "piano" in instruments:
4     print("The tuple contains the value of piano.")
5 else:
6     print("The tuple does not contain the value of piano.")
```

Try it!

Notice that checking for an item is done by using a simple if statement.

## Lists inside Tuples and Tuples inside Lists

Tuples can hold strings, integers, and even lists! Let's see what this looks like

```
1 information = ([3, 14, 5], "swimming", "walking", "flying")
2
3 print(information)
```

Try it!

This will print out the following

```
([3, 14, 5], 'swimming', 'walking', 'flying')
```

Now let's see a tuple inside of a list.

```
1 plants = ["tree", "bush", ("seed", "stem", "leaf"), "moss"]
2
3 print(plants)
```

Try it!

This will print out the following

```
['tree', 'bush', ('seed', 'stem', 'leaf'), 'moss']
```

### Not a Tuple

The following is not considered a tuple.

```
(32)
```

In order to be considered a tuple it must have a comma

```
(32,)
```

## Checkpoint

---

### Tuples

1. Create a **tuple** named `grades` .
2. Inside the tuple, include these integer values of different grades: `95, 70, 85, 92, 100.`
3. Print out the tuple.
4. Print out the second to last value using negative indexing.
5. Using a **range**, print out the first three values in the tuple.

### Requirements:

- Create a tuple named `grades` and fill it with the designated values.
- Print out the tuple named `grades`.
- Using negative indexing, print out the second to last value.
- Using a `range`, print out the first three values in the tuple.

## Questions (10)

### 1. What can be done with a tuple? Select all that apply:

SELECT MULTIPLE

Select all that apply:

- A. Tuple items can be changed.
- B. Items can be rearranged in a tuple
- C. Items can be removed from a tuple.
- D. Items can be accessed from inside a tuple.

### 2. What are ways that Python groups information together? Select all that apply:

SELECT MULTIPLE

Select all that apply:

- A. Tuples
- B. Dictionaries
- C. Genres
- D. Lists
- E. Categories

### 3. What will print from this code?

MULTIPLE CHOICE

```
desserts = ("candy", "popcorn", "soda", "cake") print(desserts[-1])
```

Choose the correct answer:

- A. It will throw an error.
- B. "cake"
- C. "candy"
- D. "popcorn"
- E. "soda"



#### 4. Debug the following code: Assume you are trying to create a tuple

DEBUG CODE

Code to Debug:

```
1 scores = [12, 39, 4, 70, 10]
```

#### 5. What is the main difference between tuples and lists?

MULTIPLE CHOICE

Choose the correct answer:

- A. Lists are changeable, tuples are not
- B. Tuples are used for integers, and lists for strings
- C. Tuples are changeable, lists are not
- D. Tuples are longer

#### 6. True or False: Tuples can be changed

MULTIPLE CHOICE

Choose the correct answer:

- A. True
- B. False

#### 7. How can you access the second item in a tuple named 'instruments'?

MULTIPLE CHOICE

Choose the correct answer:

- A. instruments(1)
- B. instruments[2]
- C. instruments(2)
- D. instruments[1]

#### 8. What is the result of the following code:

MULTIPLE CHOICE

```
instruments = ("clarinet", "piano", "drum", "violin", "guitar") print(instruments[1:3])
```

Choose the correct answer:

- A. ('piano', 'drum')
- B. ("clarinet", "piano")
- C. ("piano", "drum", "violin")
- D. ("piano", "violin")

**9. How do you check if the value "piano" exists in the tuple 'instruments'?**

Choose the correct answer:

- A. instruments.exists("piano")
- B. instruments.contains("piano")
- C. "piano" in instruments
- D. instruments.has("piano")

**10. What is the result of the following code:**

```
instruments = ("clarinet", "piano", "drum", "violin") print(instruments[-2])
```

Choose the correct answer:

- A. drum
- B. clarinet
- C. piano
- D. violin

## Challenges (3)

**1. Check for Number**

Consider the following tuple:

```
friends = (3, 5, 7, 8, 10, 2, 12, 4)
```

Create a program that will input one number.

If that number is in your tuple, print **Yes**, otherwise print **No**.

For example:

Input: **1**

Output: **No**

Another example:

Input: **7**

Output: **Yes**



## 2. Third from last

Write a program that creates a tuple with **8 strings**.

Then print the third to last value using positive indexing and negative indexing.

Using range, print the first half of the tuple.

### Requirements:

- Create a tuple with 8 items.
- Using negative indexing, print out the third to last item in the tuple.
- Using positive indexing, print out the third to last item in the tuple.
- Using range, print out the first half of the tuple.

## 3. Negative Range

Consider the following tuple:

```
candies = (3, 5, 7, 8, 10, 2, 50, 4, 29, 14)
```

Using a range that has negative numbers for both values in the range, print out `(50, 4, 29)`

### Requirements:

- Use negative numbers to print out the range.

## Answer Keys & Solutions

### Checkpoint Solutions

#### Tuples

```
1 grades = (95, 70, 85, 92, 100)
2
3 print(grades)
4
5 print(grades[-2])
6
7 print(grades[0:3])
```

### Questions

1. What can be done with a tuple? Select all that apply:

SELECT MULTIPLE

Correct Answers:

- A. Tuple items can be changed. ✗ Incorrect
- B. Items can be rearranged in a tuple ✗ Incorrect
- C. Items can be removed from a tuple. ✗ Incorrect
- D. Items can be accessed from inside a tuple. ✓ Correct

#### Explanation:

Tuples cannot be changed

2. What are ways that Python groups information together? Select all that apply:

SELECT MULTIPLE

Correct Answers:

- A. Tuples ✓ Correct
- B. Dictionaries ✓ Correct
- C. Genres ✗ Incorrect

D. Lists

✓ Correct

E. Categories

✗ Incorrect

#### Explanation:

Examples of each fruit = ["apples", "bananas"], veggies = ("carrots") desserts = { favorite: "cake", least: "pie"}

### 3. What will print from this code?

MULTIPLE CHOICE

#### Correct Answer:

A. It will throw an error.

✗ Incorrect

B. "cake"

✓ Correct

C. "candy"

✗ Incorrect

D. "popcorn"

✗ Incorrect

E. "soda"

✗ Incorrect

#### Explanation:

Indexing counting from the end starts at 1

### 4. Debug the following code: Assume you are trying to create a tuple

DEBUG CODE

#### Incorrect Code:

```
1 scores = [12, 39, 4, 70, 10]
```

#### Correct Solution:

```
1 scores = (12, 39, 4, 70, 10)
```

#### Explanation:

Tuples have parentheses

### 5. What is the main difference between tuples and lists?

MULTIPLE CHOICE

#### Correct Answer:

A. Lists are changeable, tuples are not

✓ Correct

B. Tuples are used for integers, and lists for strings

✗ Incorrect

C. Tuples are changeable, lists are not

✗ Incorrect

D. Tuples are longer

✗ Incorrect

#### Explanation:

Tuples cannot be changed

## 6. True or False: Tuples can be changed

MULTIPLE CHOICE

#### Correct Answer:

A. True

✗ Incorrect

B. False

✓ Correct

#### Explanation:

Tuples cannot be changed

## 7. How can you access the second item in a tuple named 'instruments'?

MULTIPLE CHOICE

#### Correct Answer:

A. instruments(1)

✗ Incorrect

B. instruments[2]

✗ Incorrect

C. instruments(2)

✗ Incorrect

D. instruments[1]

✓ Correct

#### Explanation:

Square brackets are used to retrieve items from a tuple.

## 8. What is the result of the following code:

MULTIPLE CHOICE

### Correct Answer:

- A. ('piano', 'drum') ✓ Correct
- B. ("clarinet", "piano") ✗ Incorrect
- C. ("piano", "drum", "violin") ✗ Incorrect
- D. ("piano", "violin") ✗ Incorrect

### Explanation:

Remember that indexing starts counting at 0. A range does not include the second number

## 9. How do you check if the value "piano" exists in the tuple 'instruments'?

MULTIPLE CHOICE

### Correct Answer:

- A. instruments.exists("piano") ✗ Incorrect
- B. instruments.contains("piano") ✗ Incorrect
- C. "piano" in instruments ✓ Correct
- D. instruments.has("piano") ✗ Incorrect

### Explanation:

You are checking to see if the string is IN the tuple

## 10. What is the result of the following code:

MULTIPLE CHOICE

### Correct Answer:

- A. drum ✓ Correct
- B. clarinet ✗ Incorrect
- C. piano ✗ Incorrect
- D. violin ✗ Incorrect

### Explanation:

Negative indexing starts counting at 1

## Challenges

### 1. Check for Number

Solution:

```
1 friends = (3, 5, 7, 8, 10, 2, 50, 4)
2
3 guess = int(input("Pick a number"))
4
5 if guess in friends:
6     print("Yes")
7 else:
8     print("No")
```

### 2. Third from last

Solution:

```
1 friends = ("Bob", "Zoe", "Tim", "Kat", "Sam", "Jim", "Jon", "Kim")
2
3 print(friends[5])
4
5 print(friends[-3])
6
7 print(friends[0:4])
```

### 3. Negative Range

Solution:

```
1 candies = (3, 5, 7, 8, 10, 2, 50, 4, 29, 14)
2
3 print(candies[-4:-1])
```