



Week 3: Complex Data Types

Unit 1: What Are Tuples?

What are tuples?

Another complex data type is the tuple

- Similar to the list but different intention
 - Lists are used for **many similar** items
 - Tuples are used for items with **many attributes**
 - Example:
Address consists of zip code, city, street, house number
- Syntax:
 - Similar to lists but with parentheses ()
- Semantics:
 - Tuples are immutable, i.e. you cannot change a value of the tuple

```
address = (52066, "Aachen", "Eupener Str. 70", "0241-6009-12345")

student = (
    "Peter",
    "Parker",
    123456,
    "Python for Beginners",
    "pp12345s@university.edu",
    address,
)
```

What are tuples?

Basic operations

- Tuples have an index like lists
 - Actually, square brackets [] are used for indexing tuples!
- Sub-tuples can be defined with slicing
- Combination of list of tuples leads to a powerful structure

```
address = (52066, "Aachen", "Eupener Str. 70", "0241-6009-12345")

student = (
    "Peter",
    "Parker",
    123456,
    "Python for Beginners",
    "pp12345s@university.edu",
    address,
)

print(address[0])
print(student[1])
print(student[5])
print(student[5][2])
print(student[-1])
print(address[-3])
print(address[2])
```

```
numbers = (1, 2, "trois", "four", "V", 6)
print(numbers[2:4])
print(len(numbers))
print(numbers.count(1))
print(numbers.index("V"))
```

What are tuples?

Showtime

Now it's time to get hands on and start programming!

If you like, you can open the [Jupyter Notebook](#) instructions in parallel to the demo.

If you haven't done so yet:

- [Download the Notebook](#)
- [Start the Jupyter Server](#)
- [Open the Notebook](#)



Jupyter Showtime_2

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

Run

Showtime 🎉

It's time to get hands on and start programming!

```
In [1]: 1 for i in range(3, 0, -1):
        2     print("...", i)
        3     print("🎉 Showtime 🎉")

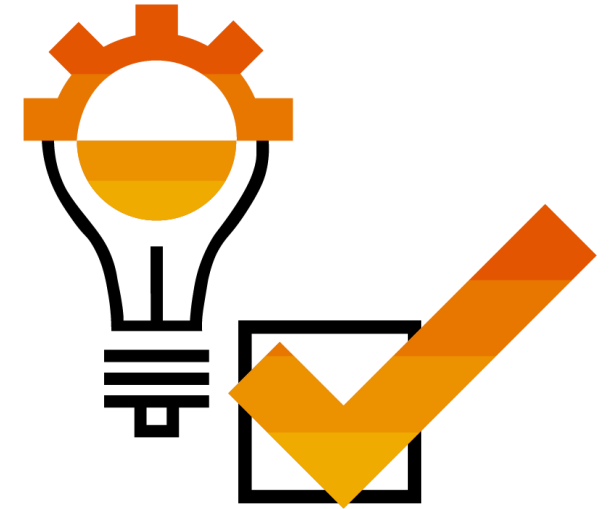
... 3
... 2
... 1
🎉 Showtime 🎉
```

What are tuples?

Summary / key takeaways

In this unit you learned ...

- ... what tuples are
- ... why they are used
- ... the differences between lists and tuples
- ... basic operations



Thank You!

Contact Information:
open@sap.com



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

hosted by
openSAP
open.sap.com



Week 3: Complex Data Types

Unit 2: What Are Dictionaries?

What are dictionaries?

Drawback of lists and tuples: access only by index

- In lists and tuples, individual elements can only be accessed by index.
 - This is sometimes cumbersome.
- Take a telephone book as an example:
 - You do not want to search for the third number.
 - You want to look up the number by a name.
- This is what dictionaries are made for: The elements are accessed not by index but by a key.
 - In the example with the telephone, the name is the key.

Name	Telephone Number
P. McCartney	123 456
J. Lennon	987 654 321
G. Harrison	11 342 555
R. Starr	777 888 32

What are dictionaries?

Definition and handling of dictionaries

- Dictionaries consists of *key-value pairs*. That means, there is always a key, which can be used to access the value.
 - In the telephone book, the key is the name of a person and the value is the telephone number.
- Syntax:
 - Dictionaries are represented by curly braces { }
 - The key-value pairs are separated by commas ,
 - Each key-value pair is represented as follows:
key : value
- An individual value is accessed by putting the key in square brackets.
- It is possible to add, change, and delete entries in the dictionary.
- Trying to access a nonexistent key leads to an error.

```
tel = {"P.McCartney" : 123456,  
      "J.Lennon" : 987654321,  
      "G.Harrison" : 11342555,  
      "R.Starr" : 77788832}  
print(tel)
```

```
print(tel["G.Harrison"])
```

What are dictionaries?


Showtime

Now it's time to get hands on and start programming!

If you like, you can open the [Jupyter Notebook](#) instructions in parallel to the demo.

If you haven't done so yet:

- [Download the Notebook](#)
- [Start the Jupyter Server](#)
- [Open the Notebook](#)



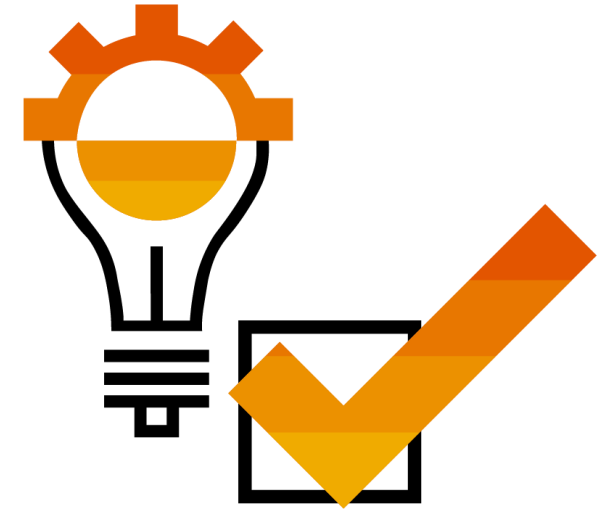
```
Jupyter Showtime_2
File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)
In [1]: 1 for i in range(3, 0, -1):
        2     print("...", i)
        3     print("🎉 Showtime 🎉")
... 3
... 2
... 1
🎉 Showtime 🎉
```

What are dictionaries?

Summary / key takeaways

In this unit you learned ...

- ... that elements in dictionaries can be accessed by a key, not by an index
- ... the elements in dictionaries are key-value pairs
- ... dictionaries are mutable, i.e. it is possible to add, delete, and change elements



Thank You!

Contact Information:
open@sap.com



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

hosted by
openSAP
open.sap.com



Week 3: Complex Data Types

Unit 3: When to Use Lists, Dictionaries, and Tuples

When to use lists, dictionaries, and tuples

Differences

Theoretically, lists, dictionaries, and tuples are almost interchangeable.

Each data structure has its advantages and disadvantages in certain use cases

- Lists
 - Useful when lots of objects of the same type (e.g. lots of cars, lots of students) have to be handled
 - Access only by index
- Tuples
 - Immutable, not changeable
 - This can be an advantage when the structure of objects is not to be changed
- Dictionaries
 - Access via key; a key is required
 - In contrast to lists: random access of elements possible via key; not necessary to search through a list
 - In contrast to tuples: structure of objects can be changed, the elements of the directory can have a name (the key) which has advantages in certain situations

Combination of different types useful and often necessary

When to use lists, dictionaries, and tuples

Showtime

Now it's time to get hands on and start programming!

If you like, you can open the [Jupyter Notebook](#) instructions in parallel to the demo.

If you haven't done so yet:

- [Download the Notebook](#)
- [Start the Jupyter Server](#)
- [Open the Notebook](#)



```

In [1]: 1 for i in range(3, 0, -1):
        2     print("...", i)
        3     print("🎉 Showtime 🎉")

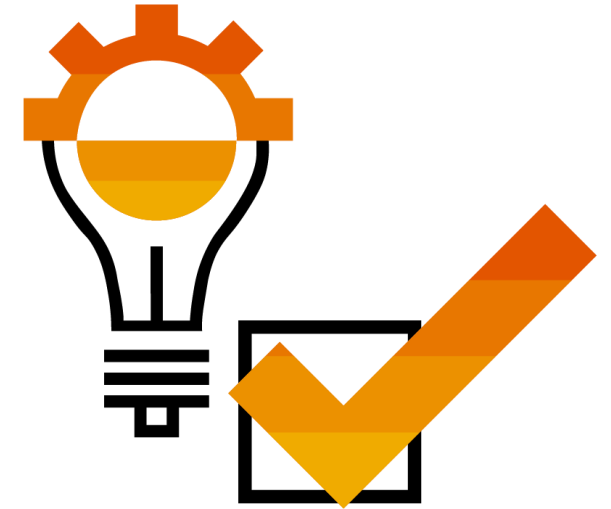
... 3
... 2
... 1
🎉 Showtime 🎉
```

When to use lists, dictionaries, and tuples

Summary / key takeaways

In this unit you learned ...

- ... the different use cases for lists, tuples, and dictionaries



Thank You!

Contact Information:
open@sap.com



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

hosted by
openSAP
open.sap.com



Week 3: Complex Data Types

Unit 4: Safe Dictionary Access

Ensuring safe access to a dictionary

- Accessing a dictionary with a nonexistent key results in a *KeyError*
 - Program is interrupted/aborted
 - Similar to accessing a list with a nonexistent index
- Using the keyword `in` it is possible to prevent this error.
 - `x in dict` returns `True` if the key `x` exists in the dictionary `dict`, `False` if it does not exist.
- Alternatively: Use the dictionary-method `dict.get()`
 - It returns either the assigned value or an error message, which can be defined by the programmer.

```
dict = {1234: "Peter", 2345: "Jane"}  
print(dict[1234])  
print(dict[3456])
```

Peter

```
-----  
KeyError                                Traceback (most recent call last)  
~\AppData\Local\Temp\ipykernel_5544\557728999.py in <module>  
      1 dict = {1234: "Peter", 2345: "Jane"}  
      2 print(dict[1234])  
----> 3 print(dict[3456])  
  
KeyError: 3456
```

Safe dictionary access

Showtime

Now it's time to get hands on and start programming!

If you like, you can open the [Jupyter Notebook](#) instructions in parallel to the demo.

If you haven't done so yet:

- [Download the Notebook](#)
- [Start the Jupyter Server](#)
- [Open the Notebook](#)



```

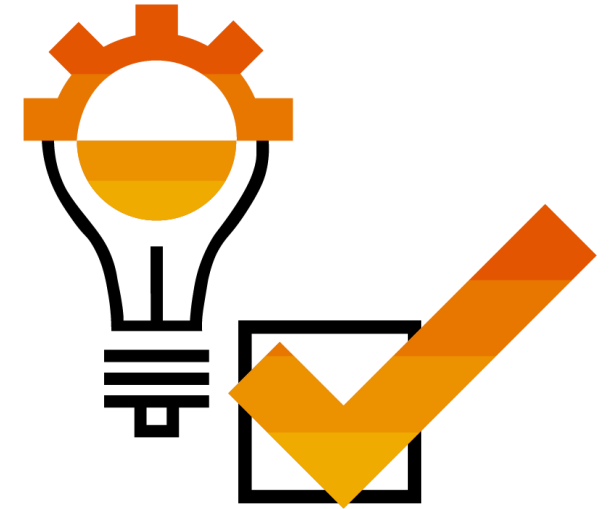
In [1]: 1 for i in range(3, 0, -1):
        2     print("...", i)
        3     print("🎉 Showtime 🎉")

... 3
... 2
... 1
🎉 Showtime 🎉
```

Summary / key takeaways

In this unit you learned ...

- ... how to prevent access errors in dictionaries
- ... how to get the value for specific keys securely without interrupting the main program



Thank You!

Contact Information:
open@sap.com



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

hosted by
openSAP
open.sap.com



Week 3: Complex Data Types

Unit 5: Important Functions and Methods for Complex Data Types

Important functions and methods for complex data types

There are a variety of methods and functions for complex data types

- Structure of lists, tuples, and dictionaries differs
- Different structures → partly different methods/functions applicable

Method	List	Tuple	Dictionary
.sort()	X		
.pop()	X		X
.count()	X	X	
.index()	X	X	

Function	List	Tuple	Dictionary
del()	X		X
len()	X	X	X
min/max()	X	X	
sorted()	X	X	

Important functions and methods for complex data types

Showtime

Now it's time to get hands on and start programming!

If you like, you can open the [Jupyter Notebook](#) instructions in parallel to the demo.

If you haven't done so yet:

- [Download the Notebook](#)
- [Start the Jupyter Server](#)
- [Open the Notebook](#)



Jupyter Showtime_2

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel) Logout

Run Code

Showtime 🎉

It's time to get hands on and start programming!

```
In [1]: 1 for i in range(3, 0, -1):
        2     print("...", i)
        3     print("🎉 Showtime 🎉")
```

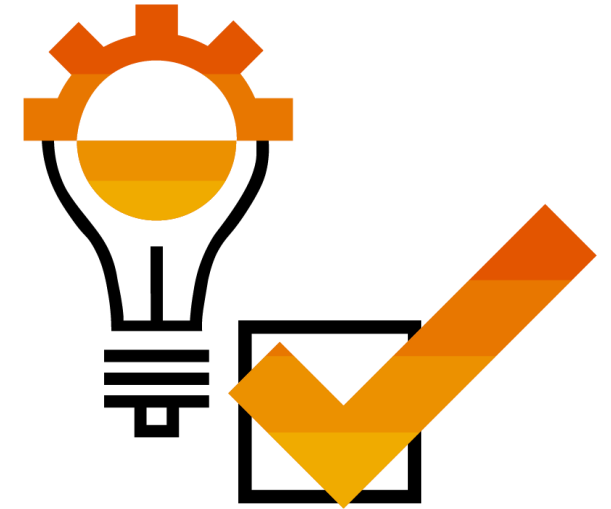
... 3
... 2
... 1
🎉 Showtime 🎉

Important functions and methods for complex data types

Summary / key takeaways

In this unit you learned ...

- ... the common methods and functions for lists, tuples, and dictionaries
- ... why not all methods are available for all types



Thank You!

Contact Information:
open@sap.com



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

hosted by
openSAP
open.sap.com



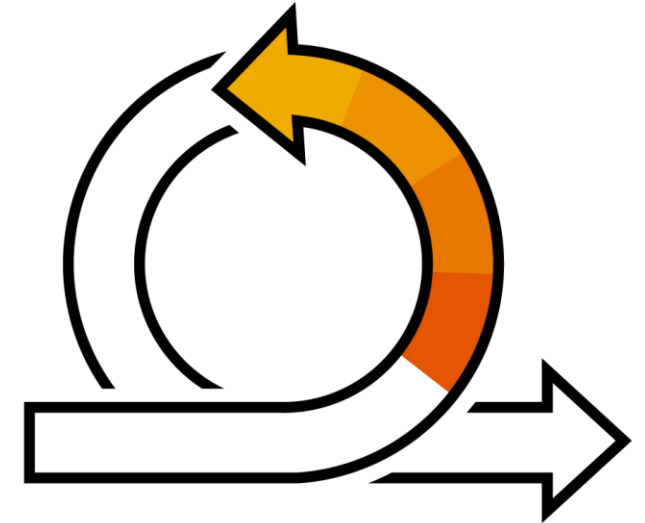
Week 3: Complex Data Types

Unit 6: While Loop

While loop

Sometimes, the 'for' loop does not fit ...

- The for loop is well suited to iterate over a sequence.
- Sometimes, you do not have a sequence. You do not even know how often you have to iterate the loop.
- Examples:
 - A PIN is entered until it is finally correct
 - An input is checked until it finally has the correct data type
 - Input data is taken until the user finishes the input with an empty return
- In these situations, a 'while' loop fits better
 - A while loop does not iterate over a sequence but iterates until a given condition is False



While loop

Syntax of the while loop

- The while loop has the following syntax:
- The behavior of the while loop is as follows:
 - If the condition is `True` all statements are executed.
 - After that, the condition is checked again. If it is `True` the statements are executed again.
 - This continues until the condition is `False`
 - If the condition is `False` right from the beginning, the statements within the loop are not executed at all.
 - If the condition will never get `False`, you are in an endless loop.

```
while condition:  
    statement1  
    statement2  
    ...  
    statementN
```

```
# The loop runs until a suitable input is available  
number = input("Please enter number: ")  
while not (number.isdecimal()):  
    number = input("Please enter number: ")  
  
number = int(number)  
print(number)
```

While loop

Showtime

Now it's time to get hands on and start programming!

If you like, you can open the [Jupyter Notebook](#) instructions in parallel to the demo.

If you haven't done so yet:

- [Download the Notebook](#)
- [Start the Jupyter Server](#)
- [Open the Notebook](#)



```

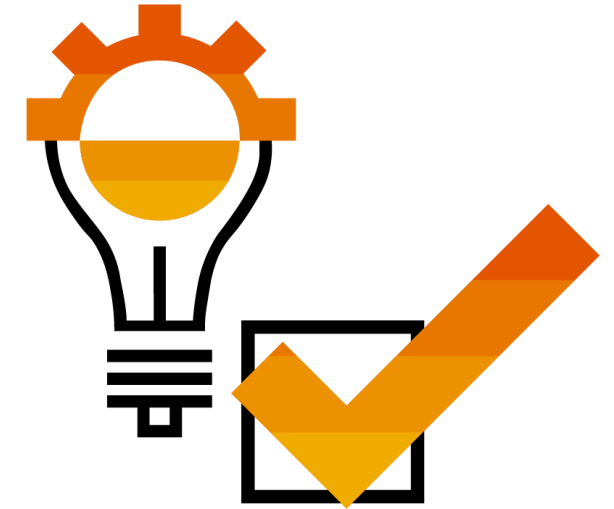
jupyter Showtime_2
File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)
[Save] [Add] [Delete] [Copy] [Paste] [Undo] [Redo] [Run] [Stop] [Refresh] [Code]
Showtime 🎉
It's time to get hands on and start programming!
In [1]: 1 for i in range(3, 0, -1):
        2     print("...", i)
        3     print("🎉 Showtime 🎉")
... 3
... 2
... 1
🎉 Showtime 🎉
```

While loop

Summary / key takeaways

In this unit you learned ...

- ... how to handle the while loop
- ... its syntax and semantics
- ... typical errors
- ... how to make your while loop more powerful using the keyword `break`



Thank You!

Contact Information:
open@sap.com



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

hosted by
openSAP
open.sap.com