

TITLE: Data strings - lists

LEARNING SCENARIO	
School:	Duration (minutes): 45
Teacher:	Students age: 13

Essential Idea:	Let's meet data strings - lists
------------------------	--

Topics:

- Pupils deepen their understanding of the use of various software and policies.

Aims:

- Pupils will be able to design and create programs that utilize subroutines, appropriate structures and data types, expressions, variables and iterative and conditional commands.
- General programming languages are used to create programs.
- Pupils understands the different ways to use simulations and step-by-step organization algorithms to solve problems.

Outcomes:

- Pupils create a more complex game, application, or mobile application that solves a particular problem from specific subject or topic.
- Pupils learn how to outline the operation of a more complex program into various patterns and generalizations.

Work forms:

- individual work
- work in pairs
- group work

Methods:

- presentation
- discussion
- interactive exercise

ARTICULATION

Course of action (duration, minutes)

INTRODUCTION

Teacher explains and starts discussion with pupils:

In the examples so far we have seen different types of input data which we have further worked on. If we have been working with multiple input data, we have stored them in different variables. Even when we have used them to perform the same operation, we have still had to write them in a special command, which has made our solutions more complex. Can we make simpler solutions?

MAIN PART

If we need to write a larger amount of data which all has a common characteristic, it is advisable to store them in a single data structure. Then we can use the input data through their common name and indexes that tell us the position of the single part of data inside the larger data structure. A common example is for example names of students, temperatures of air every day and so on. A list is a structure of data in Python that can be used to store such data. We can say that a list is made out of data that we call list elements.

Examples of lists	
[8,5,3,7,6,2]	A list of numbers with 6 elements
['Ana','Ivo','Marko','Lucija','Nikola']	A list of character strings with 5 elements
[7,8,'Leona','Darko',3.14,1.4]	A list of 6 elements – whole and decimal numbers, and character strings
[12,[2,5],[],['a','b'],'c']	A list with a whole number, a list with numbers, a list with characters, and an empty list

Elements of a list can be different types of data. A single list can contain whole and decimal numbers, characters, and even other lists.

Let's use some examples in our interface to show different ways of defining lists. Lists can be added to a variable, and we can also make a new list by applying addition (+) and multiplication (*).

```

>>> lista1=[3,7,8,2,34,5]
>>> lista1
[3, 7, 8, 2, 34, 5]
>>> lista2=[12,15,24]
>>> lista2
[12, 15, 24]

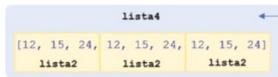
>>> lista3=lista1+lista2
>>> lista3
[3, 7, 8, 2, 34, 5, 12, 15, 24]
>>>
    
```



lista3 is the result of adding two lists (lista1 and lista2)

```
>>> lista4=lista2*3
>>> lista4
[12, 15, 24, 12, 15, 24, 12, 15, 24]
>>>
```

lista4 is the result of multiplying lista2 with the whole number 3



Grabbing a part of a list with indexing

```
>>> lista=[5, 12, 4, 26, 'a', 'b', 'c', 33]
>>> print(lista)
[5, 12, 4, 26, 'a', 'b', 'c', 33]
>>> lista[3]
26
>>> lista[5]
'b'
>>>
```



How can we grab different elements of a list? For that we can use indexes, in the same way in which we have grabbed elements of strings.

A part of a list

Just like with strings, a part of a list is defined by typing the first and the last element we want, separated with two dots (:).

```
>>> lista[2:5]
[4, 26, 'a']
>>> lista[:3]
[5, 12, 4]
>>> lista[4:]
['a', 'b', 'c', 33]
>>>
```



EXERCISE 1

Print all elements of a single data list, one below the other.

How can we know the number of elements in a list? Just like with strings, we can use the function len() which returns as a result the number of elements in a list.

```
>>> imena=['Ivan', 'Marko', 'Ana', 'Leona']
>>> for i in range (len(imena)):
    print (imena[i])
Ivan
Marko
Ana
Leona
>>>
```

1. We have defined a list of names
2. We've used len(imena) to determine the length of the list
3. We've used a loop to print out all of the elements

Number of the repetition	Value of the indeks in the loop	Executed command	Printed value
1. repetition	i=0	Print (imena[0])	Ivan
2. repetition	i=1	Print (imena[1])	Marko
3. repetition	i=2	Print (imena[2])	Ana
4. repetition	i=3	Print (imena[3])	Leona

EXERCISE 2

According to the previous example, pupils can design, create and test their own examples.

CONCLUSION

Pupils and teacher discuss and evaluate the presented solutions.

Methods

presentation
discussion
work on the text
graphic work
interactive exercise /simulation on the computer

Work forms

interview
demonstration
role playing

individual work
work in pairs
group work
frontal work

Material:

-

Literature

PERSONAL OBSERVATIONS, COMMENTS AND NOTES