Introduction to Programming

For Archaeologists

Part 2: Lists & Loops & Functions

2021-2022

Universiteit Leiden
The Netherlands

Discover the world at Leiden University

Topics of this lecture series

1. Introduction: Python, variables, comments
2. **Lists & Loops**
3. Loading and manipulating data
4. Graphs & Plots
5. SQL & Databases
6. Advanced methods: Machine Learning, QGIS integration

- Short lecture + exercises every week
- Assignment every 2 weeks
- Exam at the end

Assignment

Assignment deadlines

- Assignment 1: **22 April**
- Assignment 2: 6 May
- Assignment 3: 20 May

Any Questions?

Topics of this lecture

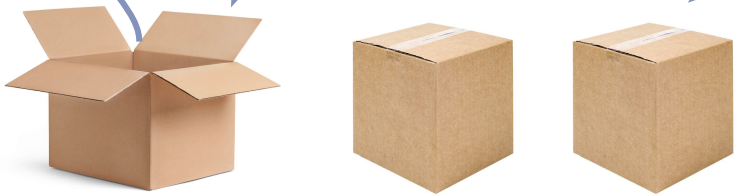
- What is a list?
- Selecting elements and slicing
- Iteration over elements using loops
- What is a dictionary?
- Functions

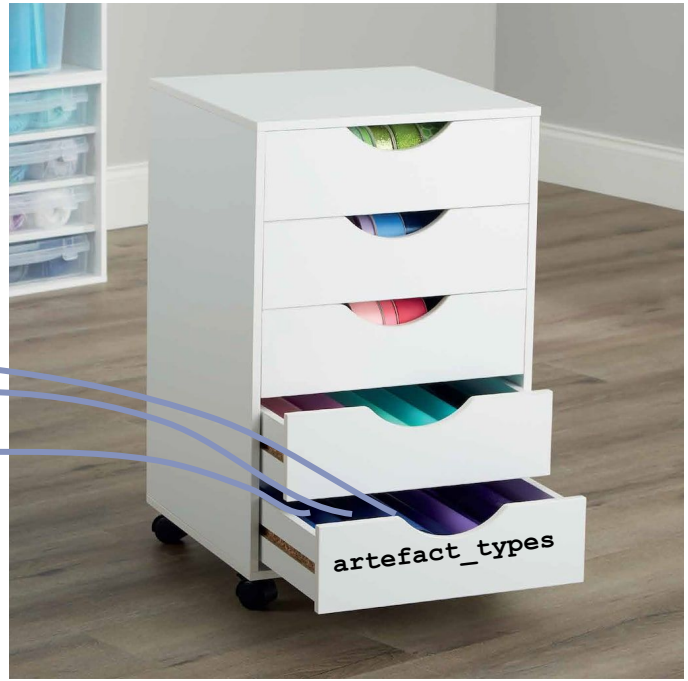
After this lecture:

- You know the difference between a list and a dictionary
- You know what a key and a value are
- You can select elements in a list, and make slices
- You can conceptually describe a loop
- You can write a basic function

Lists

A drawer containing multiple boxes, each box contains a value





'pottery'

1 value

Values in list

List name

Lists

- A variable that holds multiple values
- Each value is called an ‘element’ of the list
- Can contain a mix of different variable types (but preferably not!)
- Defined by using square brackets: [ ]

Looks like:

```
artefact_types = [‘pottery’, ‘flint’, ‘bone’]
find_numbers = [12, 13, 14, 15]
bad_list = [‘Indiana’, 123, True]
```

Remember `len()` ?

- `len()` function returns number of characters in string, but also number of elements in list!

Looks like:

```
artefact_types = [‘pottery’, ‘flint’, ‘bone’]
number_of_types = len(artefact_types)
print(number_of_types)
```

Output: `3`

Selecting 1 element

- We might want specific information from a list
- We can select 1 element from a list, using `variable[index]`

Looks like:

```
                                0           1           2
artefact_types = [‘pottery’, ‘flint’, ‘bone’]
print(artefact_types[0])
```

Output: `‘pottery’`

Selecting multiple elements (slicing)

- We can select multiple element from a list, using `variable[index:index]`

Looks like:

```
                        0                 1                 2
artefact_types = [‘pottery’, ‘flint’, ‘bone’]
print(artefact_types[0:2])
```

Output: `[‘pottery’, ‘flint’]`

Combining lists

- You can join lists together

```
pottery_weights = [12, 45]
flint_weights = [42, 10]

all_weights = pottery_weights + flint_weights
print(all_weights)
```

Output: `[12, 45, 42, 10]`

Adding to lists

- You can add elements to lists using the `append()` function

```
pottery_weights = [12, 45]
pottery_weights.append(42)

print(pottery_weights)
```

Output: `[12, 45, 42]`

Removing from lists

- You can remove elements by using the `remove()` function

```
artefact_types = [‘pottery’, ‘flint’, ‘bone’]
artefact_types.remove(‘bone’)

print(artefact_types)
```

Output: `[‘pottery’, ‘flint’]`

Loops

- Iterating over a list (go over each element in a list)
- Do something with each element in the list

Looks like:

```
artefact_types = [‘pottery’, ‘flint’, ‘bone’]
for type in artefact_types:
    print(type)
```

Always end ‘for’ with colon

indent!

Loops + if statement

- Combine `for` and `if` to check something for each value

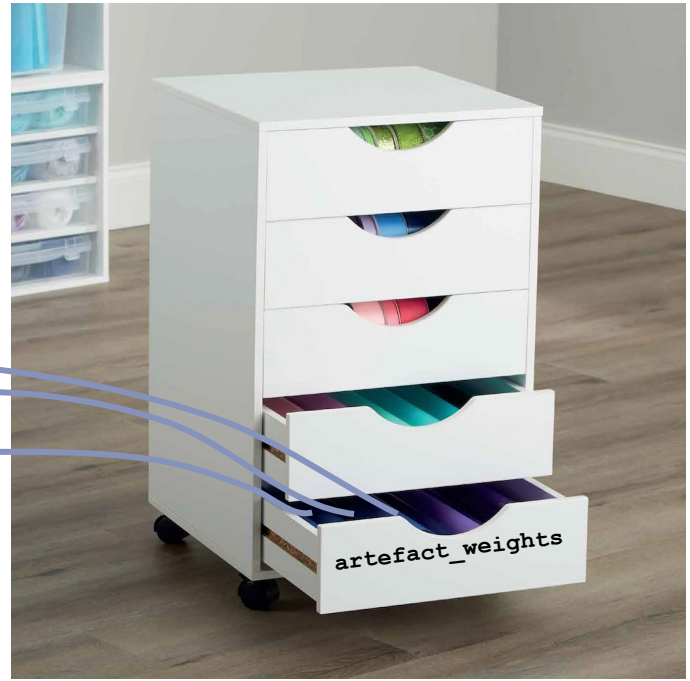
Looks like:

```
pot_weights = [12, 3, 56, 18, 20]

for weight in pot_weights:
    if weight > 15:
        print(weight)
```

Dictionaries

A drawer containing multiple **labelled** boxes, each box contains a value



12.65

bone

pottery

flint

1 value

Key/Values in dictionary

Dict name

Dictionaries

- A variable that holds multiple keys, each key corresponds to a value
- Often contains a mix of different variable typesDefined by using curly brackets: { }

Looks like:

```
artefact_weights = {'pottery': 15.5, 'flint': 56}
```

Dictionary name — Key — Value

Selecting 1 element

- We might want specific information from a dictionary
- We can select 1 element from a list, using `variable[index]`
- But here *index* is a name, not a number!

Looks like:

```
artefact_weights = {'pottery': 15.5, 'flint': 56}
print(artefact_weights['pottery'])
```

Output: `15.5`

Adding / removing elements in dictionaries

- Bit different from lists:
  - Add elements using `dict['new_key'] = new_value`
  - remove elements from dicts using the `pop()` function

```
artefact_weights = {'pottery': 15.5, 'flint': 56}
artefact_weights.pop('flint') #removes key and val
artefact_weights['bone'] = 13.6 #adds bone to dict

print(artefact_weights)
```

Output: `{'pottery': 15.5, 'bone': 13.6}`

Functions

- Already seen some: `len()`, `print()`, `remove()`, `pop()`
- Recognisable by normal brackets: `()`
- Takes one or more variables as *arguments:* `print(`*`argument`*`)`
- Performs some sort of computation, *returns* or *prints* something
- You can make your own!
- Useful when you want to do something multiple times: you re-use the code and only have to write it once

Functions

- Define a function (define = python speak for ‘create’)

```
def say_hello(name):
   print(‘Hello ’ + name)

say_hello(‘Alex’)
```

Output: ‘Hello Alex’

Functions can return stuff

- Use `return` to give a variable back:

```
def add_one(number):
    return number + 1

final_number = add_one(5)
print(final_number)
```

Output: `6`

After this lecture:

- You know the difference between a list and a dictionary
- You know what a key and a value are
- You can select elements in a list, and make slices
- You can conceptually describe a loop
- You can write a basic function

Questions?

- **Any questions about any of the subjects?**

- Contact me at
  - a.brandsen@arch.leidenuniv.nl

Slides are available on Brightspace

Exercises

github.com/alexbrandsen/Introduction-to-Programming-for-Archaeologists

- Go to github
- Click on ‘modules’
- Right click on the second module (lists & loops)
- Select ‘save link as’ or ‘download as’
- Save the file in the ‘modules’ folder within your own Scripts folder
- Start Anaconda
- Start Jupyter Notebook
- Navigate to the notebook file and run it