Lesson 1 - Intro to Python

Intro to Data Science for Social Scientists

Table of contents

Vincent Van Gogh

Today, we'll cover the following material:

- 1. Python's history, relevance and usefulness
- 2. Arithmetic Operators + Comments + Basic Functions
- 3. Variables
- 4. Elementary Data Types (`int`, `float`, `str`)
- 5. String indexing & slicing + Basic string operations
- 6. Comparison operators + Boolean datatype
- 7. Logical Operators
- 8. Lists
- 9. Homework
- 10. Additional Readings

Figure 1: van_gogh_4_banner

1. Python’s history, relevance and usefulness

Python is a widely-used general-purpose, high-level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

What started as a hobby project, soon became a general purpose programming language: nowadays, you can use Python to build practically any piece of software.

But how did this happen?

Well, first of all, Python is open source. It’s free to use.

Second, it’s very easy to build packages in Python, which is code that you can share with other people to solve specific problems. Throughout time, more and more of these packages specifically built for data science have been developed. Suppose you want to make some fancy visualizations of your company’s sales. There’s a package for that. Or what about connecting to a database to analyze sensor measurements? There’s also a package for that. People often refer to Python as the swiss army knife of programming languages as you can do almost anything with it.

2. Arithmetic Operators + Comments + Basic Functions

Figure 2: van_gogh_2_banner

```
## Multiplication
1 + 2
```

```
3
```

In Python, we use the hash sign (#) to write comments.

Everything written after a hash sign isn't evaluated - it's ignored.

I tend to use two hash signs for my comments, and one for ignoring problematic bits of code before I delete them.

This method only works for one-line comments, though.

We can also write multi-line comments, but we'll talk about it more in the later sessions of the course.

> Can you find out how to write multi-line comments in Python?Write a reproducible example in the forum.

3. Variables

Figure 3: van_gogh_3_banner

```
distance = 180 ## Our first variable. Hooray!

time = 1.5 ## Our second variable. Yuppie!
```

```
distance ## Python keeps our variables in memory! We're safe :)
```

```
180
```

```
180 / 1.5
```

```
120.0
```

```
distance / time
```

```
120.0
```

```
#distance = 180
#time = 1.5
speed = distance / time ## saving the result in a new variable
speed
```

```
120.0
```

```
#distance = 180 ## this variable stays unchanged
time = 2.5 ## this one we do change
speed = distance / time
speed ## this one changes accordingly!
```

```
72.0
```

Return to the top of the page

4. Elementary Data Types (`int`, `float`, `str`)

Figure 4: van_gogh_5.1_banner

```
## Integers
type(distance) ## Introducing the "type()" function
```

```
int
```

```
distance
```

```
180
```

```
## Floats
type(speed)
```

```
float
```

```
speed
```

```
72.0
```

So far, we’ve only worked with numeric data types (integers and floats).

There are, however, many other types of data we can (and will often) use.

> Can you think of other ways how information can be stored?

```
"Hello World" ## Introducing strings!
```

```
'Hello World'
```

```
print("Hello World")
print("How's everyone doing?") ## Using "" + '
print('It is okay, thanks for checking') ## Using '' instead of ""
```

```
Hello World
How's everyone doing?
It is okay, thanks for checking
```

```
print('it's about to get strange') ## Yes it is -> this gives us an error. Why?
```

```
SyntaxError: invalid syntax (<ipython-input-25-9d7f84d1f160>, line 1)
```

```
## Now, let's define a variable that contains string data
my_string = "Hello! How's it going?"
print(my_string)
print(type(my_string)) ## Nesting functions. Be sure the brackets are all paired up
```

```
Hello! How's it going?
<class 'str'>
```