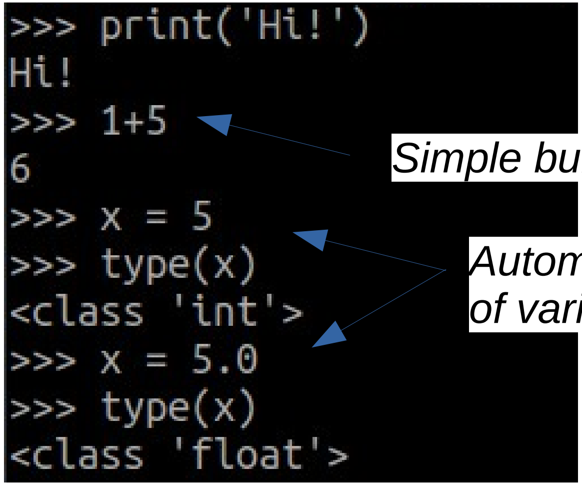
python™

A very short introduction.

print-function



```
>>> print('Hi!')
Hi!
>>> 1+5
6
>>> x = 5
>>> type(x)
<class 'int'>
>>> x = 5.0
>>> type(x)
<class 'float'>
```

Simple built-in maths

Automatic assignment
of variable types...

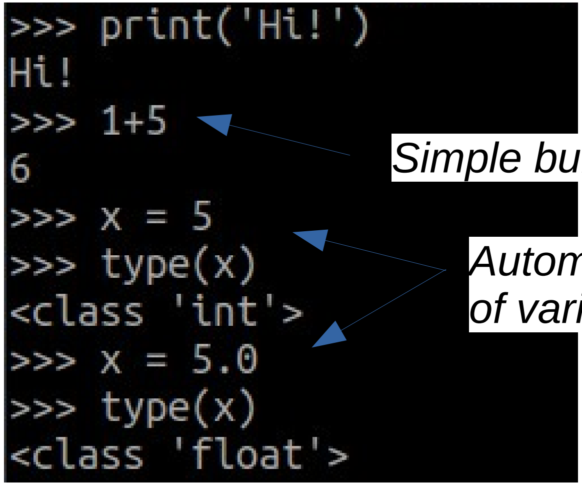
- Since 1991
  (Guido van Rossum)

- Interpreted language
  → The source code is not directly translated by the machine.
  A different program, the interpreter, reads and executed the code.

- Can also be executed as a compiled program.

- “object-oriented”

print-function

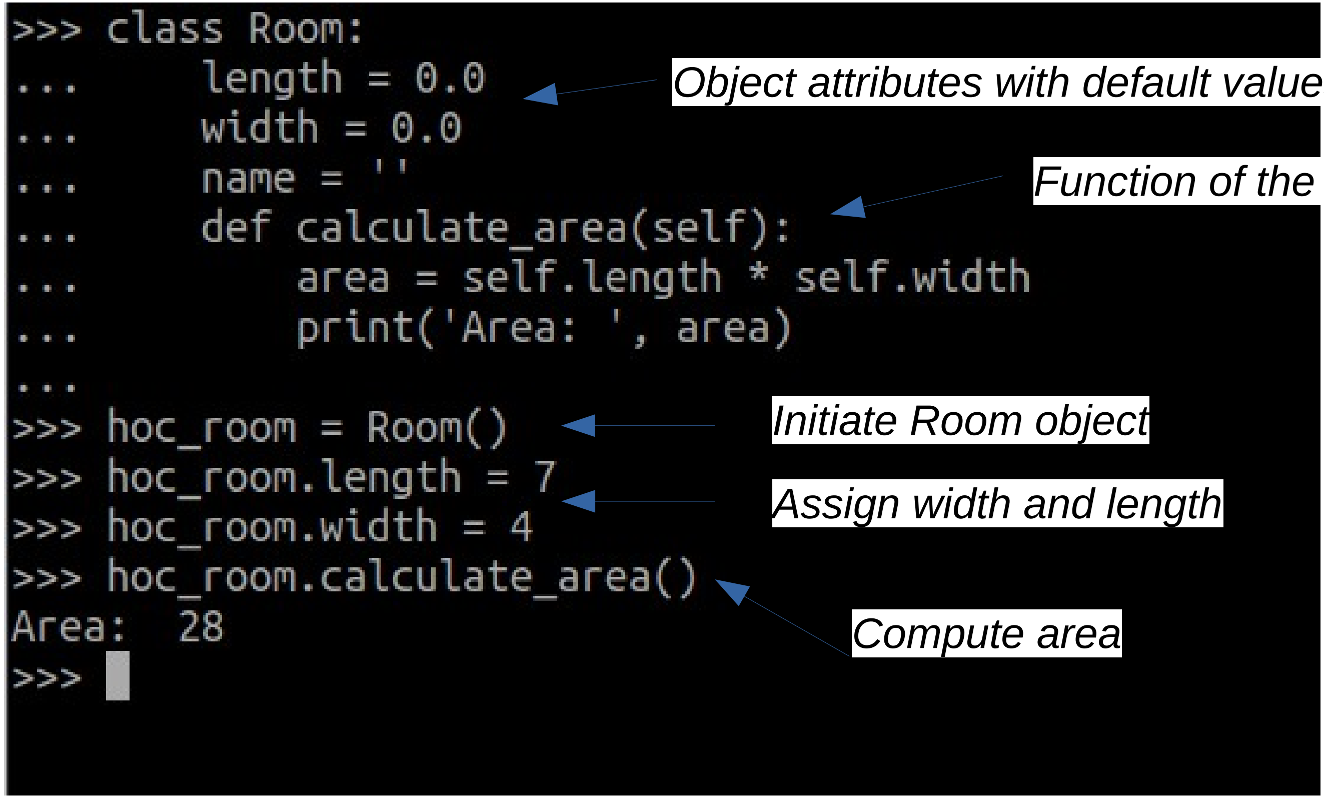


```
>>> print('Hi!')
Hi!
>>> 1+5
6
>>> x = 5
>>> type(x)
<class 'int'>
>>> x = 5.0
>>> type(x)
<class 'float'>
```

Simple built-in maths

Automatic assignment of variable types...

Object



```
>>> class Room:
...     length = 0.0
...     width = 0.0
...     name = ''
...     def calculate_area(self):
...         area = self.length * self.width
...         print('Area: ', area)
...
>>> hoc_room = Room()
>>> hoc_room.length = 7
>>> hoc_room.width = 4
>>> hoc_room.calculate_area()
Area:  28
>>> 
```

Object attributes with default values

Function of the object

Initiate Room object

Assign width and length

Compute area

Quick example
→ *Object-oriented...*

Why Python?

- Designed to be accessible.
- Easy to learn & intuitive to read.
- Very large user community, easy to find support at every level.
- Tons of useful frameworks and libraries.

Why Python?

- Designed to be accessible.
- Easy to learn & intuitive to read.
- Very large user community, easy to find support at every level.
- Tons of useful frameworks and libraries.

But...

- Python is slow.

  ...but easy to integrate other high-performance languages like C.

Applications & Libraries

(absolutely incomplete brief overview...)

- **Data science** & Academia

NumPy:

- Efficient handling of multi-dimensional arrays/matrices.
- Efficient numerical implementations.

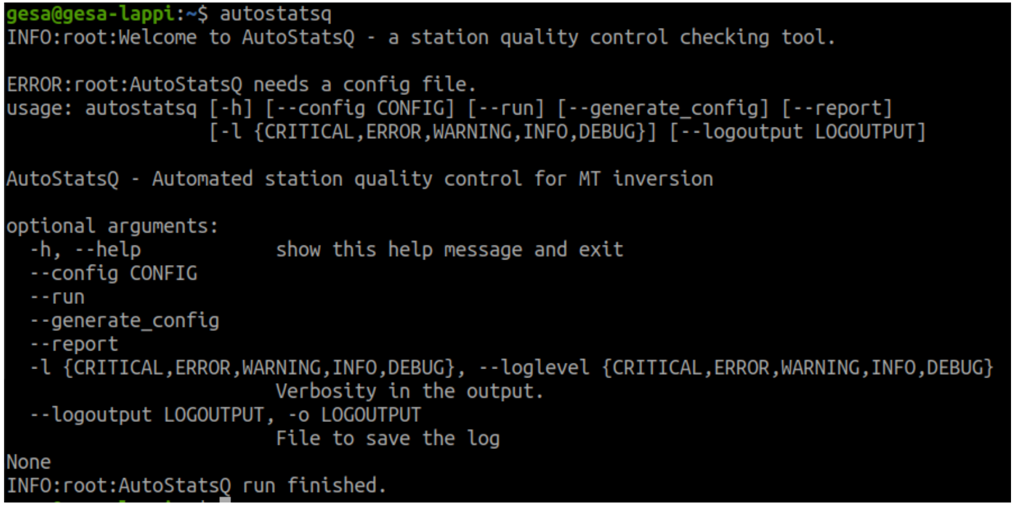
SciPy:

- Algorithms for
  - optimization tasks, Integration, Interpolations, Eigenvalues, differential equations, statistics, …
- optimized with Fortran, C, C++, ...

Applications & Libraries

(absolutely incomplete brief overview...)

- Data science & Academia
- Easy development of **command line tools** with or without graphical user interfaces



```
gesa@gesa-lappi:~$ autostatsq
INFO:root:Welcome to AutoStatsQ - a station quality control checking tool.

ERROR:root:AutoStatsQ needs a config file.
usage: autostatsq [-h] [--config CONFIG] [--run] [--generate_config] [--report]
                  [-l {CRITICAL,ERROR,WARNING,INFO,DEBUG}] [--logoutput LOGOUTPUT]

AutoStatsQ - Automated station quality control for MT inversion

optional arguments:
  -h, --help            show this help message and exit
  --config CONFIG
  --run
  --generate_config
  --report
  -l {CRITICAL,ERROR,WARNING,INFO,DEBUG}, --loglevel {CRITICAL,ERROR,WARNING,INFO,DEBUG}
                        Verbosity in the output.
  --logoutput LOGOUTPUT, -o LOGOUTPUT
                        File to save the log
None
INFO:root:AutoStatsQ run finished.
```

Applications & Libraries

(absolutely incomplete brief overview...)

- Data science & Academia
- Easy development of command line tools with or without graphical user interfaces
- **Machine learning/ AI**
  → Tensor flow, pytorch, scipy

Sorry, no figures because of copy rights ;)

Applications & Libraries

(absolutely incomplete brief overview...)

- Data science & Academia
- Easy development of command line tools with or without graphical user interfaces
- Machine learning/ AI
  → Tensor flow, pytorch, scipy
- **Publication-ready plotting**
  → matplotlib, plotly, cartopy, gmtpy, …

Sorry, no figures because of copy rights ;)

Applications & Libraries

(absolutely incomplete brief overview...)

- Data science & Academia
- Easy development of command line tools with or without graphical user interfaces
- Machine learning/ AI
  → Tensor flow, pytorch, scipy
- Publication-ready plotting
  → matplotlib, plotly, cartopy, gmtpy, …
- Many **other libraries**
  → e.g., Pyrocko for Seismology ;-)
- **Web development**: backend, server-side scripting

Sorry, no figures because of copy rights ;)

Links & Tutorials

(absolutely incomplete brief overview...)

- python.org

- Installation:
  - Linux + UNIX distributions usually include a recent Python
  - **https://www.python.org/downloads/**
  - Unix + Linux: Any editor + terminal will do.
  - Windows: Anaconda? (Simple to install, includes a programming interface)

- Getting started without installation:
  - Online python editor: https://python-editor.adamemery.dev/
  - Tutorial with python editor:
    - https://learn-python.adamemery.dev/

Find me later to talk about...

- getting started with python (Newbies absolutely welcome! <3)
- development of command line tools, plotting 2D, 3D or maps, ...