

Python

A quickstart into the key concepts of programming
Basic syntax



Thank You

- <https://github.com/gjbex/training-material/tree/master/Python>
- *Whirlwind Tour of Python* by Jake VanderPlas (O'Reilly).
Copyright 2016 O'Reilly Media, Inc., 978-1-491-96465-1.
<https://www.oreilly.com/programming/free/files/a-whirlwind-tour-of-python.pdf>
- University of Virginia, Advanced Research Computing Services, Python Quickstart
<https://arcs.virginia.edu/python-quickstart>
- <http://www.cs.cornell.edu/courses/cs1110/2018sp/>
- https://fabienmaussion.info/scientific_programming/html/00-Introduction.html



See also

- <https://www.math.ubc.ca/~pwalls/math-python/>
- <http://troll.cs.ua.edu/ACP-PY/index.html>
- <https://data-flair.training/blogs/python-lambda-expressions/>
- <http://pages.physics.cornell.edu/~myers/teaching/ComputationalMethods/GettingStarted.html>
- <https://anh.cs.luc.edu/python/hands-on/3.1/handsonHtml/index.html>
- <https://www2.cs.duke.edu/courses/spring18/compsci101/index.php>
- <https://github.com/parrt/msan501>
- <https://docs.python-guide.org/intro/learning/>



Key concepts in programming

- Variables (*integers, strings, dates, etc.*)
- Flow control (*if then, loop, etc.*)
- Functions (*named list of steps the code will follow*)



Python: basic syntax

basic_syntax.ipynb



Python files

- Python scripts are saved with `.py`
- Jupyter notebook files: `.ipynb`
- Naming convention
 - lowercase, with words separated by underscores as necessary to improve readability
 - https://visualgit.readthedocs.io/en/latest/pages/naming_convention.html



Python syntax

- *Syntax:*
 - refers to the structure of the language
 - i.e., what constitutes a correctly formed program.
- *Semantics:*
 - involve the meaning of the statements



Comment

- Comments are marked by #
- Anything on the line following the hash sign is ignored by the interpreter
- `x += 2 # shorthand for x = x + 2`
- multiline comments:
 - no syntax out of the box
 - use consecutive # single-line comments
 - Python oriented editor can help



Statement

- A statement is one complete *sentence* in the language. It contains one complete instruction. End-of-Line terminates a statement

```
B = A
surfC = math.pi*r**2
```

- Continuation: continue to the next line, use the \ marker

```
In [2]: x = 1 + 2 + 3 + 4 + \
5 + 6 + 7 + 8
```

- Semicolon can optionally terminate a statement (discouraged)

```
lower = []; upper = []
```



Code blocks

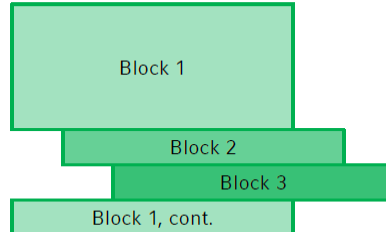
- Statements can be grouped into *blocks*
 - a *block* of code is a set of statements that should be treated as a unit.
 - Structures that introduce a block end with a colon :
- Blocks are indicated by **indentation level**.
- Indent each block by however many spaces you wish, but each block level must be indented by exactly the same number.
 - Do not use tabs, spaces are the preferred indentation method.
 - Python oriented editors (e.g. Spyder) will automatically indent the next statement to the same level as the one before it.



Code blocks

```
from math import sqrt

my_list = [1,2,3,4]
result = 0
for i in my_list:
    if i%2 == 0:
        result += sqrt(i)
print(result)
```



File: block_code.py

<http://www.dbs.ifi.lmu.de/Lehre/MaschLernen/SS2017/Orga/U01-Slides.pdf>



Indentation: Whitespace Matters

- In C, code blocks are denoted by curly braces:

```
// C code
for(int i=0; i<100; i++)
{
    total += i;
}
```

- In Python, code blocks are denoted by *indentation*:

```
for i in range(100):
    # indentation indicates code block
    total += i
```

- In Python, indented code blocks are always preceded by a colon (:) on the previous line.
- The Python standard is to use **four white spaces** to indent code.



Indentation

```
>>> if x < 4:  
...     y = x * 2  
...     print(x)
```

- `print(x)` is in the indented block, and will be executed only if `x` is less than 4.
- *File: `check_whitespace.py`*

```
>>> if x < 4:  
...     y = x * 2  
... print(x)
```

- `print(x)` is outside the block, and will be executed regardless of the value of `x`,



Whitespace

- *amount* of whitespace used for indenting code blocks is up to the user, as long as it is consistent throughout the script.
- Whitespace *within* lines does not matter

```
In [4]: x=1+2  
        x = 1 + 2  
        x      =      1      +      2
```

