

Python programming and data visualization for beginners

Dr Joel Martin



Week 1

Syntax, data types, and built-in functions

- What is the *syntax* of a programming language?
- Python's syntax
- Comments, variables, keywords, and whitespace
- Built-in data types
 - numbers, booleans, strings, lists, tuples, dictionaries
- Built-in functions
 - e.g., `print()`, `input()`, `len()`, `range()`, `round()`, `del()`

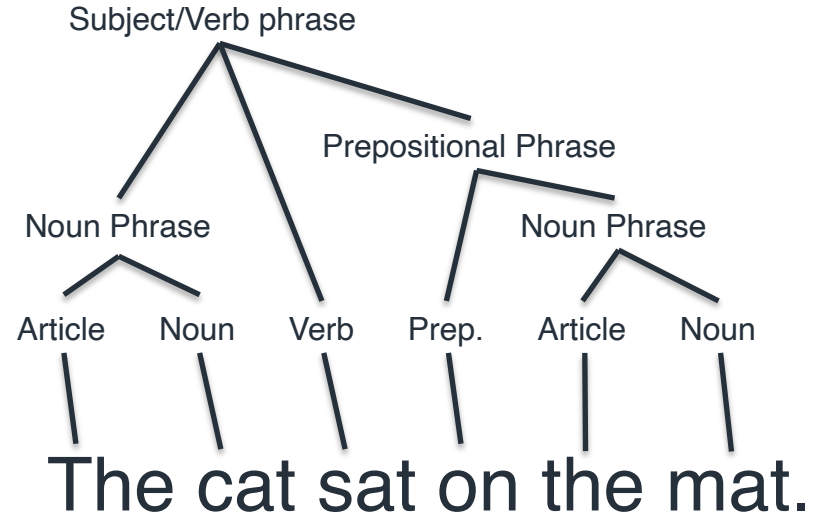
Syntax

What is it?

Syntax

noun

1. The arrangement of words and phrases to create well-formed sentences in a language.



Syntax

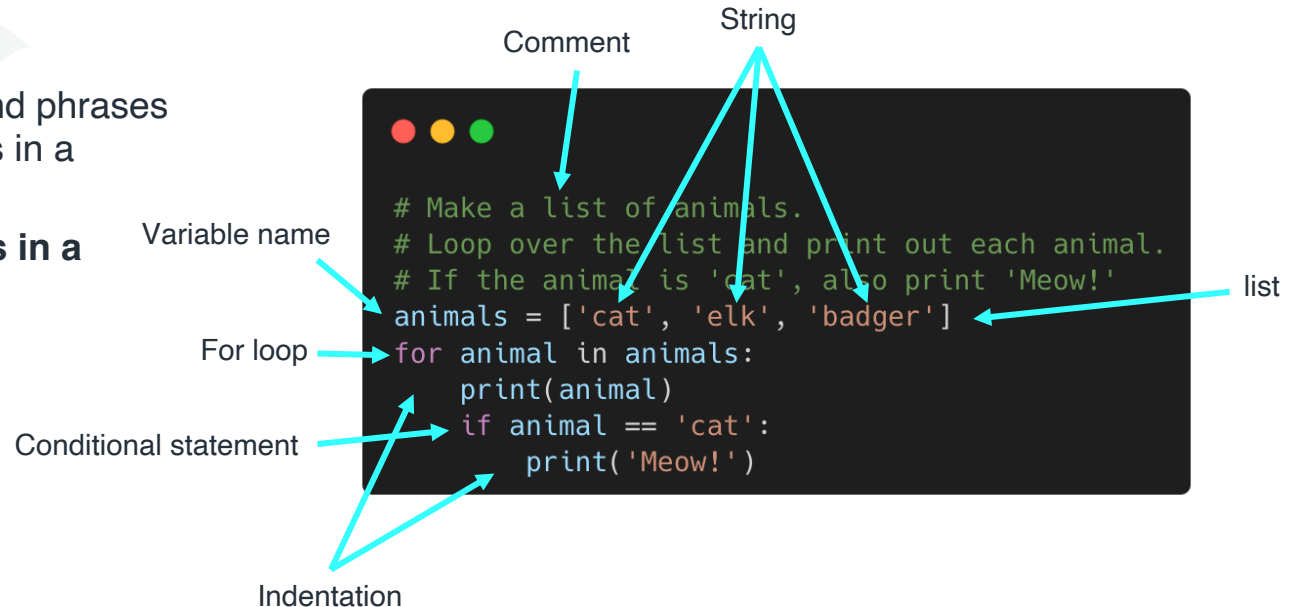
What is it?

Syntax

noun

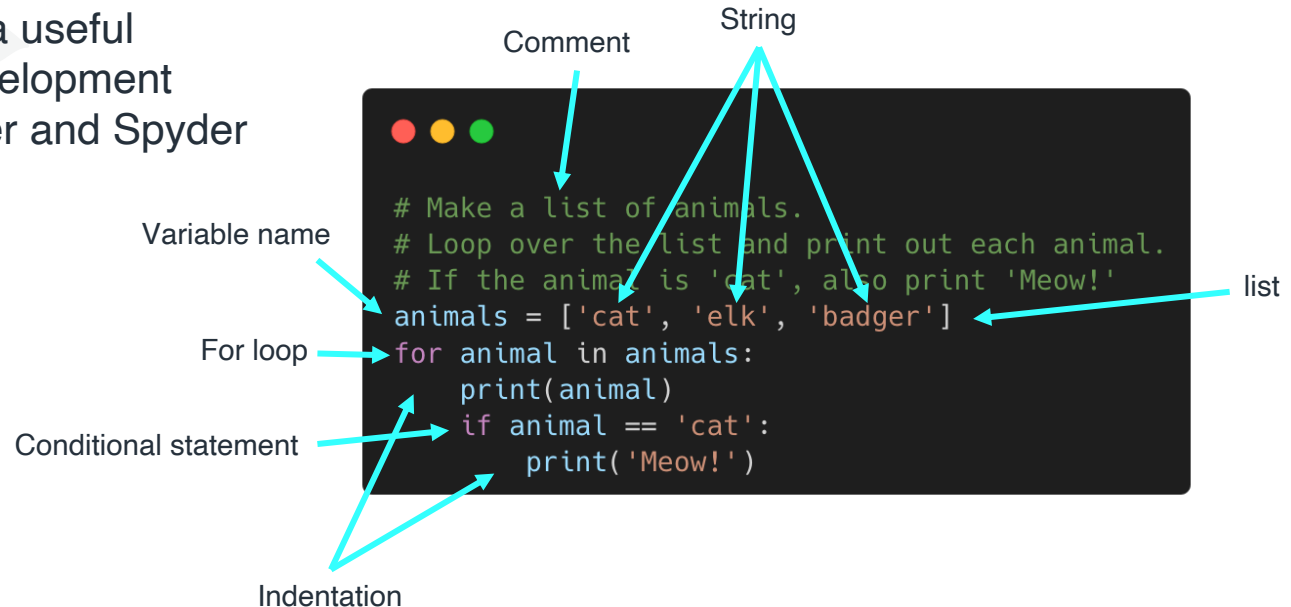
1. The arrangement of words and phrases to create well-formed sentences in a language.

2. The structure of statements in a computer language.



Python's syntax

- Python has a clear, concise and readable syntax
- Syntax highlighting** is a useful feature of integrated development environments like Jupyter and Spyder



Python's syntax

Comments

- Bits of text that don't get executed when the code is run
- May explain what the code does or why it was written that way
- Purely for the benefit of the human reading the code (i.e., future you, or a colleague)
- Anything following a # (hash) is a comment
- Can be single line, inline, multiline



```
# This is a single line comment
```

```
name = "Fred" # This is an inline comment
```

```
# This is a multiline comment  
# that requires more than one line  
# for it to be complete
```

Python's syntax

Variables

- Values are assigned to variables with the assignment operator (=)
- `<variable_name> = <value>`
- Variable names can use lower [a-z] and uppercase [A-Z] letters, the underscore character [_] and digits [0-9]
- But **they can not start with digits!**



```
number_42 = 42 # OK  
fourty_two = "fourty two" # OK  
FourtyTwo = "Fourty Two" # OK  
42_number = 42 # Not OK!
```

Python's syntax

Keywords

- Python has a set of keywords
- In the latest version of Python (3.10) there are 35 keywords (shown right)
- These are **reserved words**, which means they cannot be used as names for variables, classes or functions
- Each has its own special meaning and together they help define the Python syntax
- Don't worry about learning all of them! We will focus mainly on the ones in bold

False **break** **for** **not**
None **class** **from** **or** **True**
continue **global** **pass**
def **if** **raise** **and** **del**
import **return** **as** **elif**
in **try** **assert** **else** **is**
while **async** **except**
lambda **with** **await**
finally **nonlocal** **yield**

Python's syntax

Whitespace

- An important feature of the Python syntax!
- **Whitespace indentation is used to denote blocks of code following control statements** (e.g., if, for, while)
- Use 4 spaces (preferred) or a Tab for indentation

```
world_is_round = True

# Indentation OK (4 spaces or Tab)
if world_is_round:
    print("The world is round")

# Indentation not OK
if world_is_round:
print("The world is round")
```

Python's data types

Numbers

- Integers are whole numbers
- Floating point numbers have a fractional component
- Complex numbers have a real and imaginary part
- `int()`, `float()`, `complex()`



```
# Integer (whole number)
num_int = 42

# Floating point (fractional component)
num_float = 365.25

# Complex number (real and imaginary part)
# Included for completeness...
# Don't worry about these!
num_complex = 4+2j
```



Python's data types

Booleans

- Two possible values
 - True
 - False
- In Python, bool is a subclass of int
 - True is equal to 1
 - False is equal to 0
- Pictured top-right: George Boole (1815 - 1864), who first defined an algebraic system of logic



```
• • •

# True
world_is_round = True
if world_is_round:
    print("The world is round")

# False
world_is_flat = False
if not world_is_flat:
    print("The world is not flat")
```

Python's data types

Strings

- A sequence of values represented by Unicode points
- Can be expressed using **double or single quotes**
- Strings are **immutable**, which means they can not be modified once they have been created
- Various string methods available (e.g., `.format()`, `.upper()`, `.replace()`)



```
# Double quotes
print("Hello")

# Single quotes
print('Hello')

# Mixing quotes
print("Fred's favourite word is 'hello'")

# Mixing quotes
print('"Hello", said Fred')
```

Python's data types

Lists

- A compound data type for grouping other values together
- May contain items of different types, including other lists!
- Defined with **square brackets** and commas separating each item
- `my_list = [item, item, ..., item]`
- Support indexing, assignment, concatenation, iteration
- Built-in list methods (e.g., `.pop()`, `.append()`, `.sort()`)



```
letters = ["a", "b", "c"]
```

```
numbrs = [1, 2, 3]
```

```
mixed = [42, 365.25, 'hello', ['nested', 'list']]
```

Python's data types

Tuples

- Like lists, but they are **immutable**!
- i.e., a tuple can not be changed after it has been created
- Smaller memory footprint than lists
- Defined with **curly brackets** and commas between each item
- `my_tuple = (item, item, ..., item)`

```
# Location of York Minster
location = (53.96228434988206, -1.081881847036603)

# Display size in pixels
display = (1024, 768)
```

Python's data types

Dictionaries

- Collections of objects stored in key-value pairs
- Defined with **curly braces**, a colon between keys and values, and commas to separate key-value pairs
- `my_dict = {key: value, ..., key: value}`
- Access values by key
- e.g., `car['brand']`
- Built-in dictionary methods (e.g., `.keys()`, `.values()`, `.items()`, `.get()`)



```
# Dictionary
car = {
    'brand': 'Tesla',
    'model': 'X',
    'year': 2015
}
```

Python's built-ins

- Python has a set of built in functions that are always available
- Some are more obscure than others!
- We will mainly encounter the ones in bold

`abs()`
`aiter()`
`all()`
`any()`
`anext()`
`ascii()`
`bin()`
`bool()`
`breakpoint()`
`bytearray()`
`bytes()`
`callable()`
`chr()`
`classmethod()`
`compile()`
`complex()`
`delattr()`
`dict()`

`dir()`
`divmod()`
`enumerate()`
`eval()`
`exec()`
`filter()`
`float()`
`format()`
`frozenset()`
`getattr()`
`globals()`
`hasattr()`
`hash()`
`help()`
`hex()`
`id()`
`input()`
`int()`

`isinstance()`
`issubclass()`
`iter()`
`len()`
`list()`
`locals()`
`map()`
`max()`
`memoryview()`
`min()`
`next()`
`object()`
`oct()`
`open()`
`ord()`
`pow()`
`print()`
`property()`
`range()`
`repr()`
`reversed()`
`round()`
`set()`
`setattr()`
`slice()`
`sorted()`
`staticmethod()`
`str()`
`sum()`
`super()`
`tuple()`
`type()`
`vars()`
`zip()`