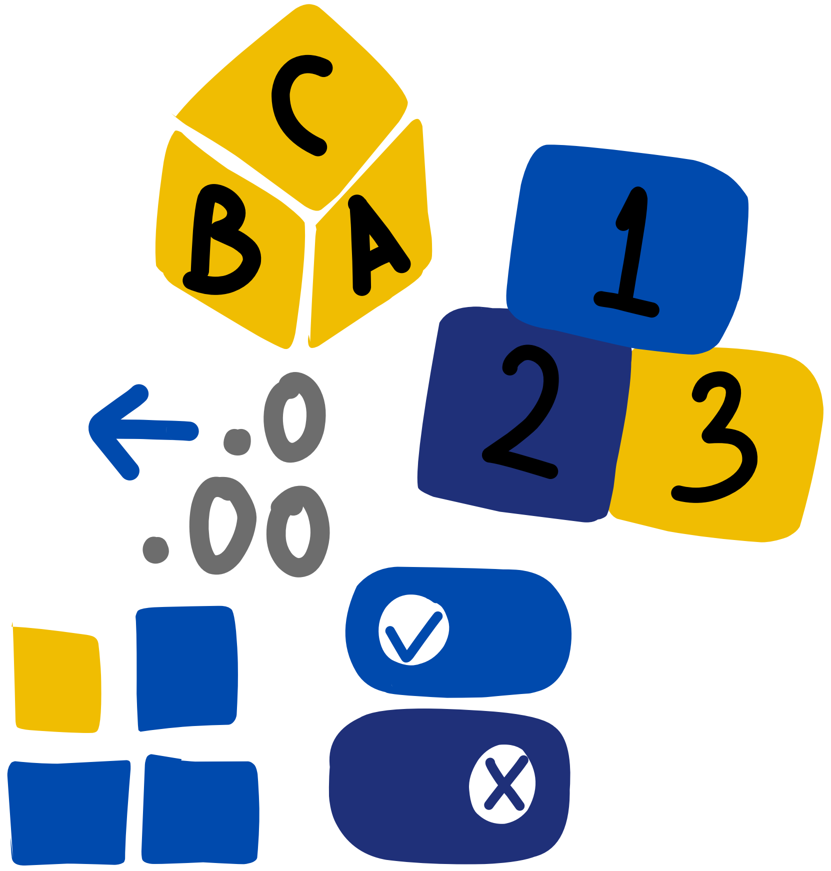
SUNRISE INSTITUTE

WELLCOME

INTRO. TO DATA SCIENCE

EMBARKING ON A JOURNEY INTO DATA SCIENCE

YA MANON



DATA TYPES

DATA TYPES

In this section we'll introduce Python **data types**, review their properties, and cover how to identify and convert between them

TOPICS WE'LL COVER:

- Data Types
- The Types Function
- Type Conversion
- Iterables
- Mutability

GOALS FOR THIS SECTION:

- Review the basics of Python data types
- Learn how to determine an object's data type
- Learn to convert between compatible data types
- Understand the concepts of iterability & mutability

PYTHON DATA TYPES

Data Types

The Types Function

Type Conversion

Iterables

Mutability

Python has a series of built-in **data types** with different properties and use cases.

These can be grouped into the following categories:

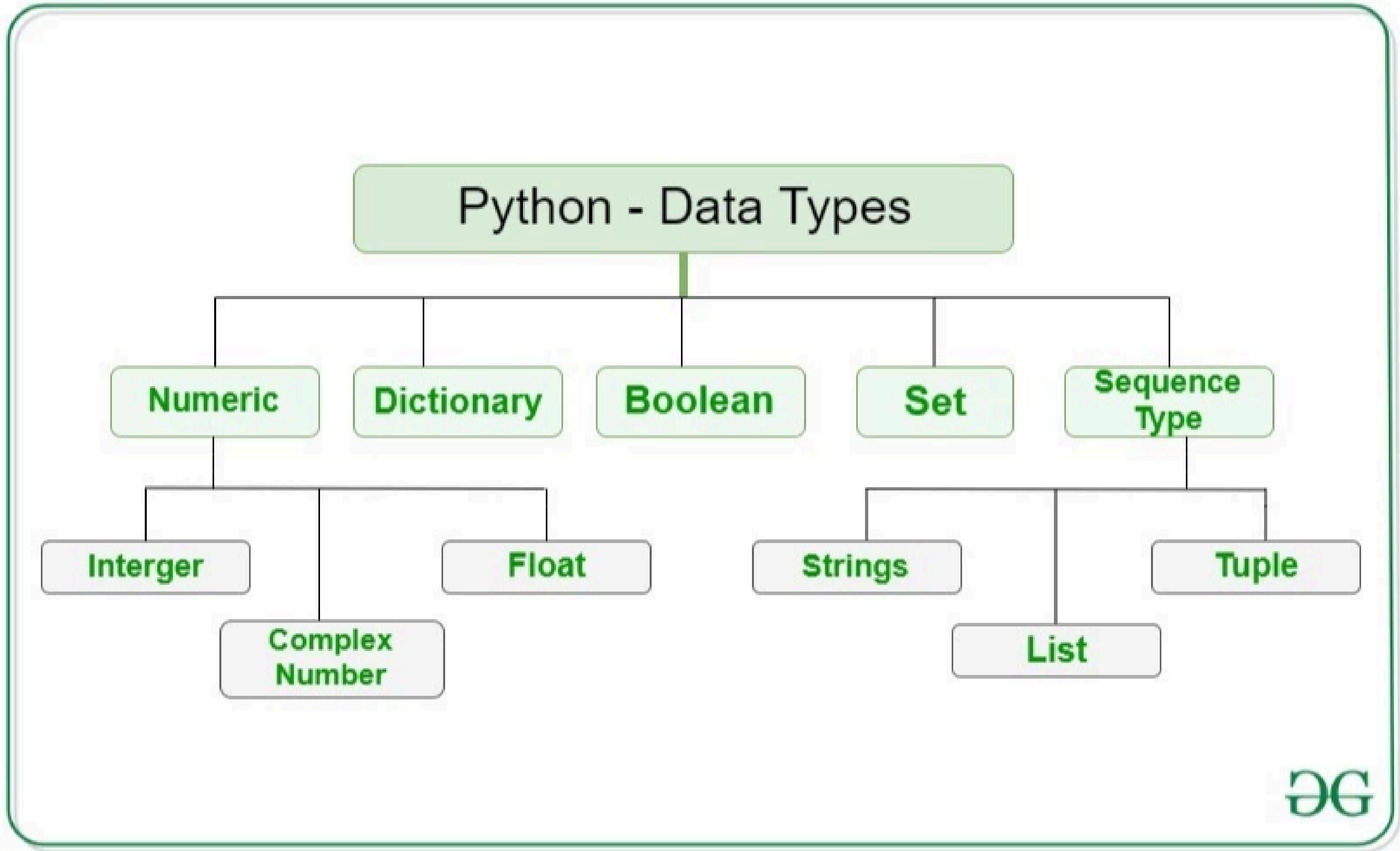


Photo by geeksforgeeks

PYTHON DATA TYPES

- **Data Types**
- The Types Function
- Type Conversion
- Iterables
- Mutability

Single value *(simple)*:

Numeric

Represents numeric values

- Integer(int)
- Float(float)
- Complex(complex)

Text

Represents sequences of characters, usually text

- String(`str`) `'snowboard'`

Boolean

Represents True and False values

, • Boolean(`bool`) – `True False`

None

Represents the absence of a value

– • `NoneType` `None`

Multiple values:

Sequence

Represents sequences of values, usually text or numeric

- List(`list`) – `[1, 3, 5, 7, 9]`
- Tuple(`tuple`) – `('snowboard', 'skis')`
- Range(`range`) – `range(1, 10, 2)`

Mapping

Maps keys to values for efficient information retrieval

- Dictionary(`dict`) – `{'snowboard': 24, 'skis': 17}`

Set

Represents a collection of unique, non-duplicate values

- Set(`set`) – `{'snowboard', 'skis'}`
- Frozen Set(`frozenset`) –`{'snowboard','skis'}`

THE TYPE FUNCTION

Data Types

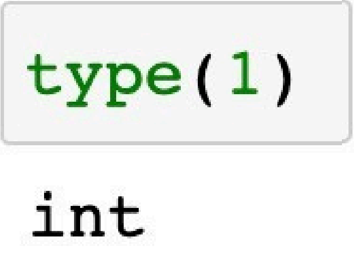
The Types Function

Type Conversion

Iterables

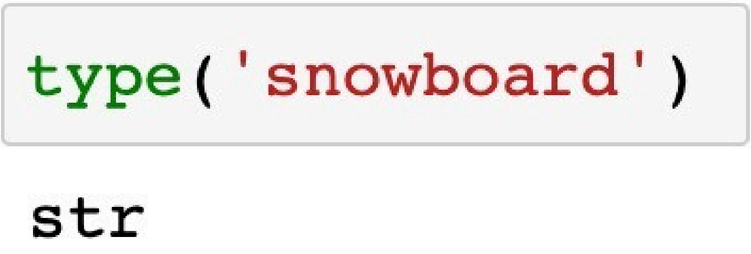
Mutability

The **type()** function will return the data type of the object passed to it



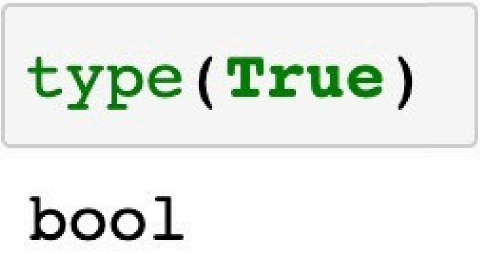
```
type(1)
```

```
int
```



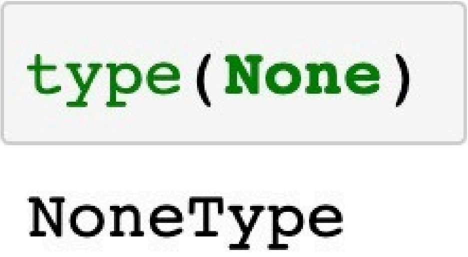
```
type('snowboard')
```

```
str
```



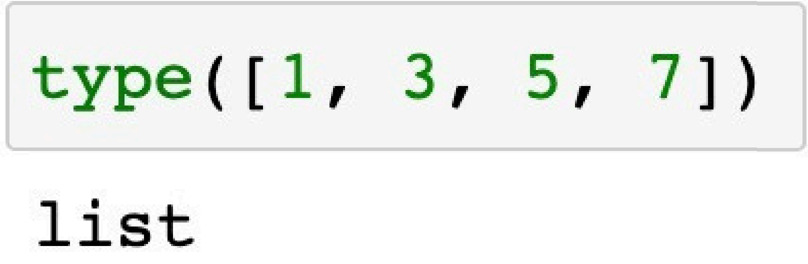
```
type(True)
```

```
bool
```



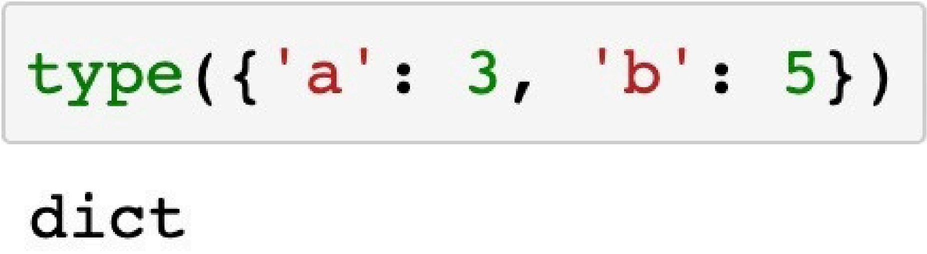
```
type(None)
```

```
NoneType
```



```
type([1, 3, 5, 7])
```

```
list
```



```
type({'a': 3, 'b': 5})
```

```
dict
```

TIP: Use **type**() if you are getting aTypeError or unexpected behavior when passing data through a function to make sure the data type is correct; it's not uncommon for values to be stored incorrectly

TYPE CONVERSION

Data Types

The Types Function

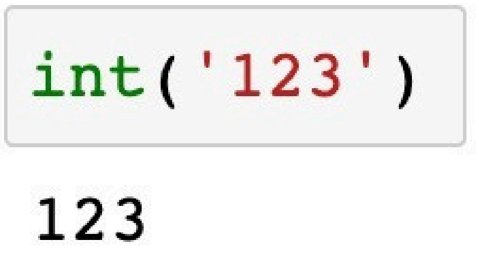
Type Conversion

Iterables

Mutability

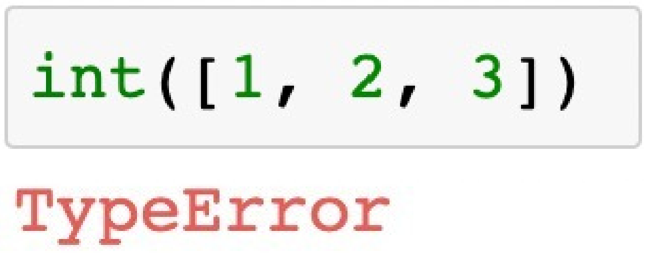
Convertdata types by using "<data type>(object)"

EXAMPLE *Converting data into an integer data type*



```
int('123')
123
```

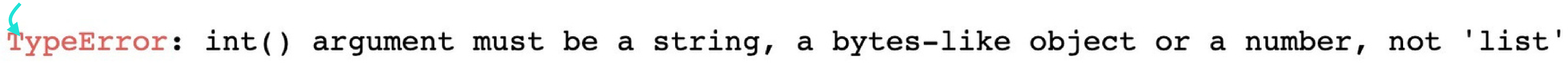
Using ***int*** *converts the text string of '123' into an integer data type*



```
int([1, 2, 3])
TypeError
```

Using ***int*** *attempts to convert the list into an integer data type, but returns a* ***TypeError*** *instead*

This operation isn't valid because the list has multiple values, whil e an integer can only be one value



```
TypeError: int() argument must be a string, a bytes-like object or a number, not 'list'
```

Errors in Python will **cause the code to stop running**, so it's important to learn to diagnose and fix them

ITERABLES

Data Types

The Types Function

Type Conversion

Iterables

Mutability

Iterables are data types that can be iterated, or looped through, allowing you to move from one value to the next

These data types are considered iterable:

Sequence

Mapping

Set

Text

While text strings are treated as a single value, individual characters in a text string can be iterated through

MUTABILITY

Data Types

The Types Function

Type Conversion

Iterables

Mutability

A data type is **mutable** if it can be modified after its creation

Mutable Data Types

Flexible –can add, remove, change values in the object

- Lists
- Dictionaries
- Sets

Immutable Data Types

To modify a value must delete and recreate the entire object

- Integers
- Floats
- Strings
- Booleans
- Tuples
- Frozenset

Note that all simple data types are immutable

KEY TAKEAWAYS

✓ Python has a wide variety of **data types**with different properties

- *It’s important to understand them at a high level for now, as we’ll dive deeper into each later in the course*

✓ The **type() function** allows you to determine an object’s data type

- *You can convert between data types if the underlying values are compatible*

✓ **Iterables** are data types with values that you can loop through

- *Text strings are iterable despite containing a single value, as you can iterate through its characters*

✓ Data types can be **mutable** or **immutable**

- *Mutable data types can be modified after their creation, while immutable data types cannot be changed without being overwritten*