0. Built-in Data Types (RECOMENDED)

This is a personal lesson from me, because IFB104 doesn't teach this first in-depth.

Numbers

- `int` (Integer; e.g `1`,`2`,`3`,`4`,`5`)
- `float` (Floating point; e.g `5.5`, `1.2`, `6.9`)
- `complex` (Complex number; e.g `2+3j`)

Booleans

- `bool` (Boolean; True or False)
  - `True` and `False` MUST have a capital first letter

Strings

- `str` (String; e.g `"Hello, World!"`)

Sequence Types

- `list` (List; e.g `[1, 2, 3, 4, 5]`)
- `tuple` (Tuple; e.g `(1,2,3,4,5)`)
- `range` (Range; Represents immutable sequence of numbers)

Mapping Type

- `dict` (Dictionary; e.g `{"name": "Alice", "age": 25}`)
  - Used to store an unordered collection of key-value pairs
  - `name` in the example is called a key
  - `Alice` in the example is called a value of `name` key

Set Types

- `set` (Set; e.g `{1,2,3,4,5}`)
  - Used to store unordered collection of unique elements
- `frozenset` (Frozen Set; Immutable version of a set)

Binary Types

- `bytes` (Bytes; e.g `b'Hello'`)
  - Used to represent a sequence of bytes

- `bytearray` (Byte Array; A mutable version of bytes)

None

- `NoneType` (None; Used to represent a null value)

Defining data types

Unlike in other languages, you do not specifically have to define the data type of a variable or value.

```
x = 5 # Auto assigned as an integer
x = "5" # Auto assigned as a string
x = 5.0 # Auto assigned as a floating point
x = [5,6,7] # Auto assigned as a list
x = True # Auto assigned as a bool
x = (5) # Auto assigned as a tuple
... # And so on
```

Converting different data types

To convert a different data type into another, you just state the data type as function and put the value/variable as an argument.

For example

```
x = 10 # Int

# Convert x into a string
x = str(x) # x now becomes "10" instead of 10

# Convert x into a float
x = float(x) # x becomes 10.0 instead of "10"
```

See Also:
1. Built-in Operators