

# PYTHON 3.13 DATATYPES GUIDE

by Atomic Badger

2/1/2025

On 2/1/2025, I asked ChatGPT to explain the following:

*Please list all datatypes used in Python 3.13. Give a description of each datatype and provide example code showing each one's use. Present each datatype, it's descriptions and examples in numbered order.*

**ChatGPT:**

“Python 3.13 offers a variety of built-in data types to handle different kinds of data. Here's a comprehensive list of these data types, along with descriptions and example code for each:

## 1. Integer (int)

- **Description:** Represents whole numbers, positive or negative, without decimals. In Python, integers have unlimited precision.

- **Example:**

```
python
CopyEdit
age = 25
temperature = -10
large_number = 12345678901234567890
```

## 2. Floating-Point Number (float)

- **Description:** Represents real numbers with a fractional part, expressed in decimal or scientific notation.

- **Example:**

```
python
CopyEdit
price = 19.99
gravity = 9.8
scientific_number = 1.23e4 # Equivalent to 12300.0
```

### 3. Complex Number (complex)

- **Description:** Represents complex numbers with real and imaginary parts. The imaginary part is denoted with a 'j'.
- **Example:**

```
python
CopyEdit
complex_num = 3 + 4j
print(complex_num.real)      # Output: 3.0
print(complex_num.imag)     # Output: 4.0
```

### 4. String (str)

- **Description:** An immutable sequence of Unicode characters used to represent text data.
- **Example:**

```
python
CopyEdit
greeting = "Hello, World!"
multiline = """This is a
multiline string."""
```

### 5. Boolean (bool)

- **Description:** Represents one of two values: True or False. Often used in conditional statements.
- **Example:**

```
python
CopyEdit
is_valid = True
has_errors = False
```

### 6. List (list)

- **Description:** An ordered, mutable collection of items, which can be of mixed data types.

- **Example:**

```
python
CopyEdit
fruits = ['apple', 'banana', 'cherry']
numbers = [1, 2, 3, 4, 5]
mixed = [1, 'apple', 3.14, True]
```

## 7. Tuple (tuple)

- **Description:** An ordered, immutable collection of items, which can be of mixed data types.

- **Example:**

```
python
CopyEdit
coordinates = (10.0, 20.0)
person = ('Alice', 30, 'Engineer')
single_element = ('single',) # Note the comma
```

## 8. Set (set)

- **Description:** An unordered collection of unique, immutable items. Useful for membership tests and eliminating duplicates.

- **Example:**

```
python
CopyEdit
unique_numbers = {1, 2, 3, 4, 5}
unique_numbers.add(6)
unique_numbers.add(3) # No effect, 3 is already in the set
```

## 9. Frozen Set (frozenset)

- **Description:** An immutable version of a set. Once created, its elements cannot be changed.

- **Example:**

```
python
CopyEdit
frozen_numbers = frozenset([1, 2, 3, 4, 5])
# frozen_numbers.add(6) # This will raise an AttributeError
```

## 10. Dictionary (dict)

- **Description:** An unordered, mutable collection of key-value pairs. Keys must be immutable and unique.

- **Example:**

```
python
CopyEdit
student = {'name': 'John', 'age': 21, 'major': 'Computer Science'}
print(student['name']) # Output: John
```

## 11. Bytes (bytes)

- **Description:** An immutable sequence of bytes, typically used for binary data.

- **Example:**

```
python
CopyEdit
byte_data = b'Hello, World!'
print(byte_data[0]) # Output: 72 (ASCII value of 'H')
```

## 12. Byte Array (bytearray)

- **Description:** A mutable sequence of bytes. Similar to bytes, but can be modified in place.

- **Example:**

```
python
CopyEdit
mutable_bytes = bytearray(b'Hello')
mutable_bytes[0] = 74 # ASCII value of 'J'
print(mutable_bytes) # Output: bytearray(b'Jello')
```

## 13. Range (range)

- **Description:** Represents an immutable sequence of numbers, commonly used for looping a specific number of times.

- **Example:**

```
python
CopyEdit
for i in range(5):
    print(i)
# Output: 0 1 2 3 4
```

## 14. NoneType (None)

- **Description:** Represents the absence of a value or a null value. There's only one instance of NoneType, which is None.
- **Example:**

```
python
CopyEdit
result = None
if result is None:
    print("No result available")
```

These built-in data types form the foundation of Python's type system, allowing for flexible and efficient data manipulation.”