

🌟 Data Types in Python

In Python, a data type defines the type of a value that a variable can hold. Python has several built-in data types that allow you to store and manipulate different kinds of data. Understanding these data types is fundamental to writing effective and efficient code.

🚀 Common Data Types in Python

- **int**: Number without decimal point.
- **float**: Number with decimal point.
- **string**: Text
 - [Methods: capitalize, upper, lower,]
- **list[]**: ordered and changeable. Allows duplicate items.
 - [Methods: append, pop, clear, sort]
- **tuple()**: ordered and unchangeable. Allows duplicate items.
- **set{}**: unordered, unchangeable, and unindexed. No duplicate items.
- **dict{}**: Key Value Pairs
- **boolean**: True or False
- **None**: None of the type

🎉 Real Life Example

Imagine you're managing a small library and need to keep track of the following:

Total number of books	int
Average book rating	float
Name of librarian	str
Whether library is currently open	bool
List of book names available	list
Location coordinates of the library	tuple
Contact details of the library staff	dict
A set of unique genres available	set

Example Demo

Here is the Python code demonstrating the above information:

```
# Variables representing library data
total_books = 500 # int
average_rating = 4.5 # float
librarian_name = "John Doe" # str
is_open = True # bool
book_names = ["Python Programming", "Introduction to Data Science",
"History of Art"] # list
location_coordinates = (40.7128, -74.0060) # tuple (latitude,
longitude)
contact_details = {
    "phone": "123-456-7890",
    "email": "library@example.com",
    "address": "123 Library St, City, State"
} # dict
unique_genres = {"Fiction", "Non-fiction", "Science Fiction"} # set

# Printing out the library information
print("Library Information:")
print(f"Total number of books: {total_books}")
print(f"Average book rating: {average_rating}")
print(f"Librarian's name: {librarian_name}")
print(f"Is library open?: {'Yes' if is_open else 'No'}")
print(f"List of book names available: {book_names}")
print(f"Location coordinates: {location_coordinates}")
print(f"Contact details: {contact_details}")
print(f"Unique genres available: {unique_genres}")
```

Using Type int

Here **number_of_books** represents the total count of books available in a library. It is an integer (int) data type.

```
# Total number of books
number_of_books = 350
print(f"Total number of books: {number_of_books}")
print(f"Type of number_of_books: {type(number_of_books)}")
```

Using Type float

Here **temperature** represents the current temperature outside. It is a floating-point number (double) data type, used for values that can have decimal points, such as temperatures, measurements, or calculations involving fractions.

```
#Example: Temperature in Celsius
temperature = 28.5
print(f"Current temperature: {temperature} degrees Celsius")
print(f"Type of temperature: {type(temperature)}")
```

Using Type str

Here **username** stores the username of a user's social media account. It is a string (str) data type, used for storing textual data like names, addresses, or any sequence of characters.

```
# Example: Username of a social media account
username = "johndoe85"
print(f"Username: {username}")
print(f"Type of username: {type(username)}")
```

Using Type bool

Here **is_subscribed** indicates whether a user is subscribed to a service (True) or not (False). It is a boolean (bool) data type, used for storing logical values, typically used for conditions or flags.

```
# Example: Subscription status of a service
is_subscribed = True
print(f"Is the user subscribed?: {'Yes' if is_subscribed else 'No'}")
print(f"Type of is_subscribed: {type(is_subscribed)}")
```

Using Type list

Here **student_grades** contains the grades of students. It is a list (list) data type, used for storing ordered collections of items, allowing easy access and modification.

```
# Example: Grades of students
student_grades = [85, 90, 75, 95, 88]
print(f"Student grades: {student_grades}")
print(f"Type of student_grades: {type(student_grades)}")
```

Using Type tuple

Here **city_coordinates** stores the latitude and longitude coordinates of a specific city. It is a tuple (tuple) data type, used for fixed collections where the order and number of elements should not change.

```
# Example: Geographic coordinates of a city
city_coordinates = (40.7128, -74.0060)
print(f"City coordinates: {city_coordinates}")
print(f"Type of city_coordinates: {type(city_coordinates)}")
```

Using Type dict

Here **product_info** contains various details about a product, such as its name, brand, price, and availability. It is a dictionary (dict) data type, used for storing key-value pairs and enabling efficient retrieval and manipulation of data based on keys.

```
# Example: Details of a product
product_info = {
    "name": "Smartphone",
    "brand": "Apple",
    "price": 999.99,
    "in_stock": True
}
# Print product information
print("Product information:")
print(f"  Name: {product_info['name']}")
print(f"  Brand: {product_info['brand']}")
print(f"  Price: ${product_info['price']}")
print(f"  In Stock: {'Yes' if product_info['in_stock'] else 'No'}")
# Print type of product_info
print(f"Type of product_info: {type(product_info)}")
```

Using Type set

In this example, **blog_post_tags** represents unique tags associated with a blog post, such as topics or keywords. It is a set (set) data type, used for storing unique and unordered collections of items, ensuring each element is distinct.

```
# Example: Tags associated with a blog post
blog_post_tags = {"Python", "Data Science", "Machine Learning"}
print(f"Tags for the blog post: {blog_post_tags}")
print(f"Type of blog_post_tags: {type(blog_post_tags)}")
```



Challenge 3: Your Finance Manager

Write down what you spend each day from Sunday to Saturday using a dictionary, add them up to find the total for the week, and figure out the average amount spent per day.