| **Data Type & Description** | **Example Functions** | **Explanations** |
| --- | --- | --- |
| **int: Integer numbers** | **int(x)**: Converts **x** to an integer.  **abs(x)**: Returns the absolute value of **x**.  **divmod(x, y)**: Returns the quotient and remainder of **x** divided by **y**.  **x.bit\_length()**: Returns the number of bits needed to represent **x**. | Integers in Python can be converted, absolute values can be found, and other operations are available. |
| **float: Floating-point numbers** | **float(x)**: Converts **x** to a floating-point number.  **round(x, n)**: Rounds **x** to **n** decimal places.  **x.is\_integer()**: Checks if **x** is an integer. | Floating-point numbers can be converted, rounded, and checked for being integers. |
| **str: Strings (sequences of characters)** | **len(s)**: Returns the length (number of characters) of string **s**. **s.upper()**: Converts string **s** to uppercase.  **s.lower()**: Converts string **s** to lowercase.  **s.split(separator)**: Splits string **s** into a list of substrings using **separator**.  **separator.join(iterable)**: Joins elements of **iterable** into a single string using **separator**. | String length, case conversion, splitting, and joining operations are common string manipulations. |
| **bool: Boolean values (True or False)** | **bool(x)**: Converts **x** to a Boolean.  **not x**: Negates Boolean **x**.  **x and y**: Logical AND between **x** and **y**.  **x or y**: Logical OR between **x** and **y**.  **x == y**: Checks if **x** is equal to **y**. **x != y**: Checks if **x** is not equal to **y**. | Booleans can be created, negated, and combined with logical operations for decision-making. |
| **list: Ordered, mutable sequences** | **list(iterable)**: Creates a new list from **iterable**.  **lst.append(x)**: Appends element **x** to list **lst**. **lst.extend(iterable)**: Appends elements from **iterable** to list **lst**.  **lst.pop()**: Removes and returns the last element of list **lst**. **lst.count(x)**: Counts occurrences of element **x** in list **lst**. | Lists can be created, modified, and searched. |
| **tuple: Ordered, immutable sequences** | **tuple(iterable)**: Creates a new tuple from **iterable**. **t.count(x)**: Counts occurrences of element **x** in tuple **t**. **t.index(x)**: Returns the index of the first occurrence of element **x** in tuple **t**. | Tuples are similar to lists but are immutable, so they are useful for data that shouldn't change. |
| **set: Unordered, mutable collections** | **set(iterable)**: Creates a new set from **iterable**.  **s.add(x)**: Adds element **x** to set **s**.  **s.remove(x)**: Removes element **x** from set **s**. **s.union(other\_set)**: Returns the union of set **s** and **other\_set**. **s.intersection(other\_set)**: Returns the intersection of set **s** and **other\_set**. | Sets are used to store unique elements and perform set operations like union and intersection. |
| **dict: Unordered, mutable key-value mappings** | **dict(key1=value1, key2=value2, ...)**: Creates a new dictionary.  **d.keys()**: Returns keys of dictionary **d**.  **d.values()**: Returns values of dictionary **d**.  **d.items()**: Returns key-value pairs of dictionary **d** as tuples.  **d.get(key)**: Returns the value associated with **key** in dictionary **d**. | Dictionaries are used for key-value storage, and you can access keys, values, or key-value pairs as needed. |
| **NoneType: Represents the absence of a value** | N/A | **None** is a special value representing the absence of a value or a null value. |