# Python Dunder Methods

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Specific To | Description | Example Snippet |
| \_\_new\_\_ | Object Creation | Called to create a new instance of a class. | obj = ClassName() |
| \_\_getitem\_\_ | Indexing | Defines behavior for getting an item using indexing. | value = obj[index] |
| \_\_repr\_\_ | Representation | Defines the “official” string representation of an object. | repr(obj) |
| \_\_str\_\_ | String Conversion | Defines the string representation of an object. | str(obj) |
| \_\_setitem\_\_ | Indexing | Defines behavior for setting an item using indexing. | obj[index] = value |
| \_\_le\_\_ | Comparison | Defines behavior for the <= operator. | result = obj1 <= obj2 |
| \_\_eq\_\_ | Comparison | Defines behavior for the == operator. | result = obj1 == obj2 |
| \_\_init\_\_ | Object Initialization | Called when an instance is created. | obj = ClassName() |
| \_\_del\_\_ | Destructor | Called when an object is about to be destroyed. | del obj |
| \_\_get\_\_ | Descriptor | Called to get an attribute from an instance. | value = descriptor.\_\_get\_\_(instance) |
| \_\_set\_\_ | Descriptor | Called to set an attribute on an instance. | descriptor.\_\_set\_\_(instance, value) |
| \_\_subclasshook\_\_ | Subclassing | Used to customize behavior for subclass checks. | issubclass(subclass, ClassName) |
| \_\_dir\_\_ | Attributes | Returns a list of valid attributes for the object. | attributes = dir(obj) |
| \_\_class\_\_ | Class | Returns the class type of the instance. | cls = obj.\_\_class\_\_ |
| \_\_ge\_\_ | Comparison | Defines behavior for the >= operator. | result = obj1 >= obj2 |
| \_\_gt\_\_ | Comparison | Defines behavior for the > operator. | result = obj1 > obj2 |
| \_\_lt\_\_ | Comparison | Defines behavior for the < operator. | result = obj1 < obj2 |
| \_\_hash\_\_ | Hashing | Defines behavior for hashing the object. | hash\_value = hash(obj) |
| \_\_ne\_\_ | Comparison | Defines behavior for the != operator. | result = obj1 != obj2 |
| \_\_add\_\_ | Arithmetic | Defines behavior for the + operator. | result = obj1 + obj2 |
| \_\_sub\_\_ | Arithmetic | Defines behavior for the - operator. | result = obj1 - obj2 |
| \_\_mul\_\_ | Arithmetic | Defines behavior for the \* operator. | result = obj1 \* obj2 |
| \_\_truediv\_\_ | Arithmetic | Defines behavior for the / operator. | result = obj1 / obj2 |
| \_\_floordiv\_\_ | Arithmetic | Defines behavior for the // operator. | result = obj1 // obj2 |
| \_\_mod\_\_ | Arithmetic | Defines behavior for the % operator. | result = obj1 % obj2 |
| \_\_pow\_\_ | Arithmetic | Defines behavior for the \*\* operator. | result = obj1 \*\* obj2 |
| \_\_lshift\_\_ | Bitwise | Defines behavior for the << operator. | result = obj1 << obj2 |
| \_\_rshift\_\_ | Bitwise | Defines behavior for the >> operator. | result = obj1 >> obj2 |
| \_\_and\_\_ | Bitwise | Defines behavior for the & operator. | result = obj1 & obj2 |
| \_\_xor\_\_ | Bitwise | Defines behavior for the ^ operator. | result = obj1 ^ obj2 |
| \_\_or\_\_ | Bitwise | Defines behavior for the | operator. | result = obj1 | obj2 |
| \_\_neg\_\_ | Unary | Defines behavior for the unary - operator. | result = -obj |
| \_\_pos\_\_ | Unary | Defines behavior for the unary + operator. | result = +obj |
| \_\_invert\_\_ | Unary | Defines behavior for the ~ operator. | result = ~obj |
| \_\_abs\_\_ | Built-in | Defines behavior for the abs() function. | result = abs(obj) |
| \_\_round\_\_ | Built-in | Defines behavior for the round() function. | result = round(obj, ndigits) |
| \_\_trunc\_\_ | Built-in | Defines behavior for truncating the object. | result = math.trunc(obj) |
| \_\_floor\_\_ | Built-in | Defines behavior for floor function. | result = math.floor(obj) |
| \_\_ceil\_\_ | Built-in | Defines behavior for ceiling function. | result = math.ceil(obj) |
| \_\_cmp\_\_ | Comparison | Defines behavior for the comparison operators. | result = cmp(obj1, obj2) |
| \_\_ior\_\_ | Bitwise | Defines behavior for the |= operator. | obj |= other |
| \_\_ror\_\_ | Bitwise | Defines behavior for the reverse | operator. | result = other | obj |
| \_\_iadd\_\_ | Arithmetic | Defines behavior for the += operator. | obj += other |
| \_\_iand\_\_ | Bitwise | Defines behavior for the &= operator. | obj &= other |
| \_\_idiv\_\_ | Arithmetic | Defines behavior for the /= operator. | obj /= other |
| \_\_imod\_\_ | Arithmetic | Defines behavior for the %= operator. | obj %= other |
| \_\_imul\_\_ | Arithmetic | Defines behavior for the \*= operator. | obj \*= other |
| \_\_ipow\_\_ | Arithmetic | Defines behavior for the \*\*= operator. | obj \*\*= other |
| \_\_isub\_\_ | Arithmetic | Defines behavior for the -= operator. | obj -= other |
| \_\_ixor\_\_ | Bitwise | Defines behavior for the ^= operator. | obj ^= other |
| \_\_radd\_\_ | Arithmetic | Defines behavior for the reverse + operator. | result = other + obj |
| \_\_rand\_\_ | Bitwise | Defines behavior for the reverse & operator. | result = other & obj |
| \_\_rdiv\_\_ | Arithmetic | Defines behavior for the reverse / operator. | result = other / obj |
| \_\_rmul\_\_ | Arithmetic | Defines behavior for the reverse \* operator. | result = other \* obj |
| \_\_rmod\_\_ | Arithmetic | Defines behavior for the reverse % operator. | result = other % obj |
| \_\_rpow\_\_ | Arithmetic | Defines behavior for the reverse \*\* operator. | result = other \*\* obj |
| \_\_rrshift\_\_ | Bitwise | Defines behavior for the reverse >> operator. | result = other >> obj |
| \_\_rlshift\_\_ | Bitwise | Defines behavior for the reverse << operator. | result = other << obj |
| \_\_rtruediv\_\_ | Arithmetic | Defines behavior for the reverse / operator. | result = other / obj |
| \_\_rtrunc\_\_ | Built-in | Defines behavior for the reverse truncating the object. | result = math.trunc(other) |
| \_\_complex\_\_ | Type Conversion | Defines behavior for converting to complex. | result = complex(obj) |
| \_\_int\_\_ | Type Conversion | Defines behavior for converting to int. | result = int(obj) |
| \_\_long\_\_ | Type Conversion | Defines behavior for converting to long. | result = long(obj) |
| \_\_float\_\_ | Type Conversion | Defines behavior for converting to float. | result = float(obj) |
| \_\_oct\_\_ | Type Conversion | Defines behavior for converting to octal. | result = oct(obj) |
| \_\_hex\_\_ | Type Conversion | Defines behavior for converting to hexadecimal. | result = hex(obj) |
| \_\_index\_\_ | Indexing | Defines behavior for indexing. | result = obj.\_\_index\_\_() |
| \_\_rsub\_\_ | Arithmetic | Defines behavior for the reverse - operator. | result = other - obj |
| \_\_rxor\_\_ | Bitwise | Defines behavior for the reverse ^ operator. | result = other ^ obj |
| \_\_delitem\_\_ | Indexing | Defines behavior for deleting an item using indexing. | del obj[key] |
| \_\_missing\_\_ | Dictionary | Called when a key is not found in a dictionary. | value = obj[key] if key in obj else obj.\_\_missing\_\_(key) |
| \_\_getattribute\_\_ | Attribute Access | Called when an attribute is accessed. | value = obj.attr |
| \_\_contains\_\_ | Membership Test | Defines behavior for membership tests using `in`. | if key in obj: |
| \_\_ilshift\_\_ | Bitwise Left Shift Assignment | Performs a left shift operation and assigns the result. | obj <<= value |
| \_\_imatmul\_\_ | Matrix Multiplication Assignment | Performs a matrix multiplication operation and assigns the result. | obj @= value |
| \_\_irshift\_\_ | Bitwise Right Shift Assignment | Performs a right shift operation and assigns the result. | obj >>= value |
| \_\_itruediv\_\_ | True Division Assignment | Performs true division and assigns the result. | obj /= value |
| \_\_rdivmod\_\_ | Reverse Division and Modulus | Called for the built-in function `divmod` when the object is on the right. | divmod(value, obj) |
| \_\_rmatmul\_\_ | Reverse Matrix Multiplication | Called for matrix multiplication when the object is on the right. | value @ obj |
| \_\_setstate\_\_ | State Restoration | Used to restore state during unpickling. | obj.\_\_setstate\_\_(state) |
| \_\_deepcopy\_\_ | Deep Copy | Called for creating a deep copy of the object. | new\_obj = copy.deepcopy(obj) |
| \_\_instancecheck\_\_ | Instance Check | Called to check if an object is an instance of a class. | isinstance(instance, ClassName) |
| \_\_subclasscheck\_\_ | Subclass Check | Called to check if a class is a subclass of another class. | issubclass(SubClass, SuperClass) |
| \_\_coerce\_\_ | Coercion | Called to convert one object to another type for binary operations. | result = obj + other |
| \_\_delattr\_\_ | Attribute Deletion | Called to delete an attribute from an object. | del obj.attr |
| \_\_divmod\_\_ | Division and Modulus | Called for the built-in function `divmod`. | divmod(obj, other) |
| \_\_getattr\_\_ | Attribute Access | Called when an attribute is not found through the usual lookup. | value = obj.non\_existing\_attr |
| \_\_prepare\_\_ | Class Preparation | Called before class creation to prepare a namespace. | namespace = Class.\_\_prepare\_\_(name, bases) |
| \_\_matmul\_\_ | Matrix Multiplication | Defines the behavior for the matrix multiplication operator `@`. | result = obj @ other |
| \_\_delete\_\_ | Delete Descriptor | Called to delete a descriptor. | del obj.attr |
| \_\_ifloordiv\_\_ | Floor Division Assignment | Performs floor division and assigns the result. | obj //= value |
| \_\_rfloordiv\_\_ | Reverse Floor Division | Called for floor division when the object is on the right. | other // obj |
| \_\_bool\_\_ | Boolean Conversion | Called to determine the truth value of an object. | if obj: |
| \_\_copy\_\_ | Shallow Copy | Called for creating a shallow copy of the object. | new\_obj = copy.copy(obj) |
| \_\_iter\_\_ | Iterator | Called to return an iterator object. | for item in obj: |
| \_\_next\_\_ | Next Item | Called to return the next item from an iterator. | item = next(iterator) |
| \_\_len\_\_ | Length | Called to return the length of the object. | length = len(obj) |
| \_\_aiter\_\_ | Async Iterator | Called to return an asynchronous iterator object. | async for item in obj: |
| \_\_anext\_\_ | Async Next Item | Called to return the next item from an async iterator. | item = await async\_iter.\_\_anext\_\_() |
| \_\_await\_\_ | Await | Called to make an object awaitable. | await obj |
| \_\_bytes\_\_ | Bytes Conversion | Called to convert an object to bytes. | byte\_representation = bytes(obj) |
| \_\_enter\_\_ | Context Manager Enter | Called when entering a context manager. | with obj as value: |
| \_\_delslice\_\_ | Slice Deletion | Called to delete a slice from a sequence. | del obj[start:end] |
| \_\_call\_\_ | Callable | Allows an object to be called as a function. | result = obj(arg1, arg2) |
| \_\_setattr\_\_ | Attribute Setting | Called to set an attribute on an object. | obj.attr = value |
| \_\_aenter\_\_ | Async Context Manager Enter | Called when entering an async context manager. | async with obj as value: |
| \_\_fspath\_\_ | Filesystem Path | Called to convert an object to a filesystem path. | path = os.fspath(obj) |
| \_\_reduce\_\_ | Serialization | Called to prepare the object for serialization. | data = pickle.dumps(obj) |
| \_\_sizeof\_\_ | Size of Object | Called to return the size of the object in bytes. | size = sys.getsizeof(obj) |
| \_\_class\_getitem\_\_ | Class Item Access | Called for indexing class objects. | item = MyClass[index] |
| \_\_mro\_entries\_\_ | Method Resolution Order Entries | Called for building the method resolution order. | mro = MyClass.\_\_mro\_\_ |
| \_\_format\_\_ | Formatting | Called to format an object using `format()`. | formatted = format(obj, "format\_spec") |
| \_\_reduce\_ex\_\_ | Serialization with Protocol | Called for serialization with a specific protocol. | data = pickle.dumps(obj, protocol) |
| \_\_init\_subclass\_\_ | Subclass Initialization | Called when a class is subclassed. | class SubClass(ClassName): pass |
| \_\_getstate\_\_ | State Retrieval | Called to retrieve the state for serialization. | state = obj.\_\_getstate\_\_() |
| \_\_reversed\_\_ | Reversed | Called for the built-in `reversed()` function. | for item in reversed(obj): |
| \_\_unicode\_\_ | Unicode Conversion | Called to convert an object to a Unicode string. | unicode\_str = unicode(obj) |
| \_\_getnewargs\_\_ | New Arguments | Called to get arguments for a new instance. | args = obj.\_\_getnewargs\_\_() |
| \_\_getinitargs\_\_ | Init Arguments | Called to get arguments for initialization. | args = obj.\_\_getinitargs\_\_() |
| \_\_setslice\_\_ | Slice Setting | Called to set a slice in a sequence. | obj[start:end] = sequence |
| \_\_set\_name\_\_ | Name Setting | Called to set the name of a descriptor. | self.\_\_set\_name\_\_(owner, name) |
| \_\_aexit\_\_ | Async Context Manager Exit | Called when exiting an async context manager. | async with obj as value: |
| \_\_exit\_\_ | Context Manager Exit | Called when exiting a context manager. | with obj as value: |
| \_\_text\_signature\_\_ | Text Signature | Provides a signature for function documentation. | signature = func.\_\_text\_signature\_\_() |
| \_\_mro\_\_ | Method Resolution Order | Returns the method resolution order for a class. | mro = ClassName.\_\_mro\_\_ |
| \_\_base\_\_ | Base Class | Returns the base class of a type. | base = TypeName.\_\_base\_\_ |
| \_\_dict\_\_ | Attributes Dictionary | Returns the dictionary of an object's attributes. | attributes = obj.\_\_dict\_\_ |
| \_\_flags\_\_ | Flags | Returns the flags of a type. | flags = TypeName.\_\_flags\_\_ |
| \_\_basicsize\_\_ | Basic Size | Returns the size of the basic structure. | size = TypeName.\_\_basicsize\_\_ |
| \_\_weakrefoffset\_\_ | Weak Reference Offset | Returns the offset for weak references. | offset = TypeName.\_\_weakrefoffset\_\_ |
| \_\_dictoffset\_\_ | Dictionary Offset | Returns the offset for the dictionary. | offset = TypeName.\_\_dictoffset\_\_ |
| \_\_subclasses\_\_ | Subclasses | Returns a list of subclasses for a class. | subclasses = TypeName.\_\_subclasses\_\_() |
| \_\_itemsize\_\_ | Item Size | Returns the size of items in a type. | size = TypeName.\_\_itemsize\_\_ |
| \_\_buffer\_\_ | Buffer | Called to expose the object as a buffer. | buffer = obj.\_\_buffer\_\_(flags) |
| \_\_release\_buffer\_\_ | Release Buffer | Called to release the buffer. | obj.\_\_release\_buffer\_\_(buffer) |
| \_\_alloc\_\_ | Allocation | Called to allocate space for the object. | obj.\_\_alloc\_\_() |