

---

# NSEnumerator

<b>Inherits From:</b>	NSObject
<b>Conforms To:</b>	NSObject (NSObject)
<b>Declared In:</b>	Foundation/NSUtilities.h

---

## Class at a Glance

### Purpose

An abstract class whose instances enumerate collections of other objects, such as arrays and dictionaries.

### Creation

All creation methods are defined in the collection classes such as NSArray and NSDictionary. These methods contain the word “Enumerator,” as in NSArray’s **objectEnumerator** method or NSDictionary’s **keyEnumerator** method.

### Commonly Used Methods

– nextObject Returns the next object in the collection being enumerated.

### Primitive Methods

– nextObject

---

## Class Description

NSEnumerator is a simple abstract class whose subclasses enumerate collections of other objects. Collection objects—such as arrays, sets, and dictionaries—provide special NSEnumerator objects with which to enumerate their contents. You send **nextObject** repeatedly to a newly-created NSEnumerator

object to have it return the next object in the original collection. When the collection is exhausted, **nil** is returned.

Use **allObjects** if you need an array containing the objects being enumerated.

Collection classes such as `NSArray`, `NSSet`, and `NSDictionary` include methods that return an enumerator appropriate to the type of collection. For instance, `NSArray` has two methods that return an `NSEnumerator` object: **objectEnumerator** and **reverseObjectEnumerator**. `NSDictionary` also has two methods that return an `NSEnumerator` object: **keyEnumerator** and **objectEnumerator**. These methods let you enumerate the contents of an `NSDictionary` by key or by value, respectively.

**Note:** It isn't safe to modify a mutable collection while enumerating through it.

### Method Types

Getting the objects

- `allObjects`
- `nextObject`

### Instance Methods

#### **allObjects**

– (NSArray \*)**allObjects**

Returns the collection being enumerated as an `NSArray`.

#### **nextObject**

– (id)**nextObject**

Returns the next object from the collection being enumerated. When **nextObject** returns **nil**, all objects have been enumerated. The following code illustrates how this works using `NSArray`:

```
NSEnumerator *enumerator = [anArray objectEnumerator];
id object;

while ((object = [enumerator nextObject])) {
    // do something with object...
}
```