List, ArrayList and Generics

By the end of this video you will be able to

- Work with Java's ArrayList class and List type
- Create containers with generic types

Abstract data type: List

ignore for now

```
List<Feature> countries = new ArrayList<Feature>();
```

Ordered list of things of type Feature

Abstract data type: List

```
List<Feature> countries = new ArrayList<Feature>();
```

Ordered list of things of type Feature

Abstract data type: List

Types don't match?

```
List<Feature> countries = new ArrayList<Feature>();
```

Ordered list of things of type Feature

Abstract data type: List

Java "interface"
Specifies behaviors, not implementation

```
List<Feature> countries = new ArrayList<Feature>();
```

Abstract data type: List

Actual Java class
Implements List behaviors

```
List<Feature> countries = new ArrayList<Feature>();
```

More to come in future modules and courses in this specialization!

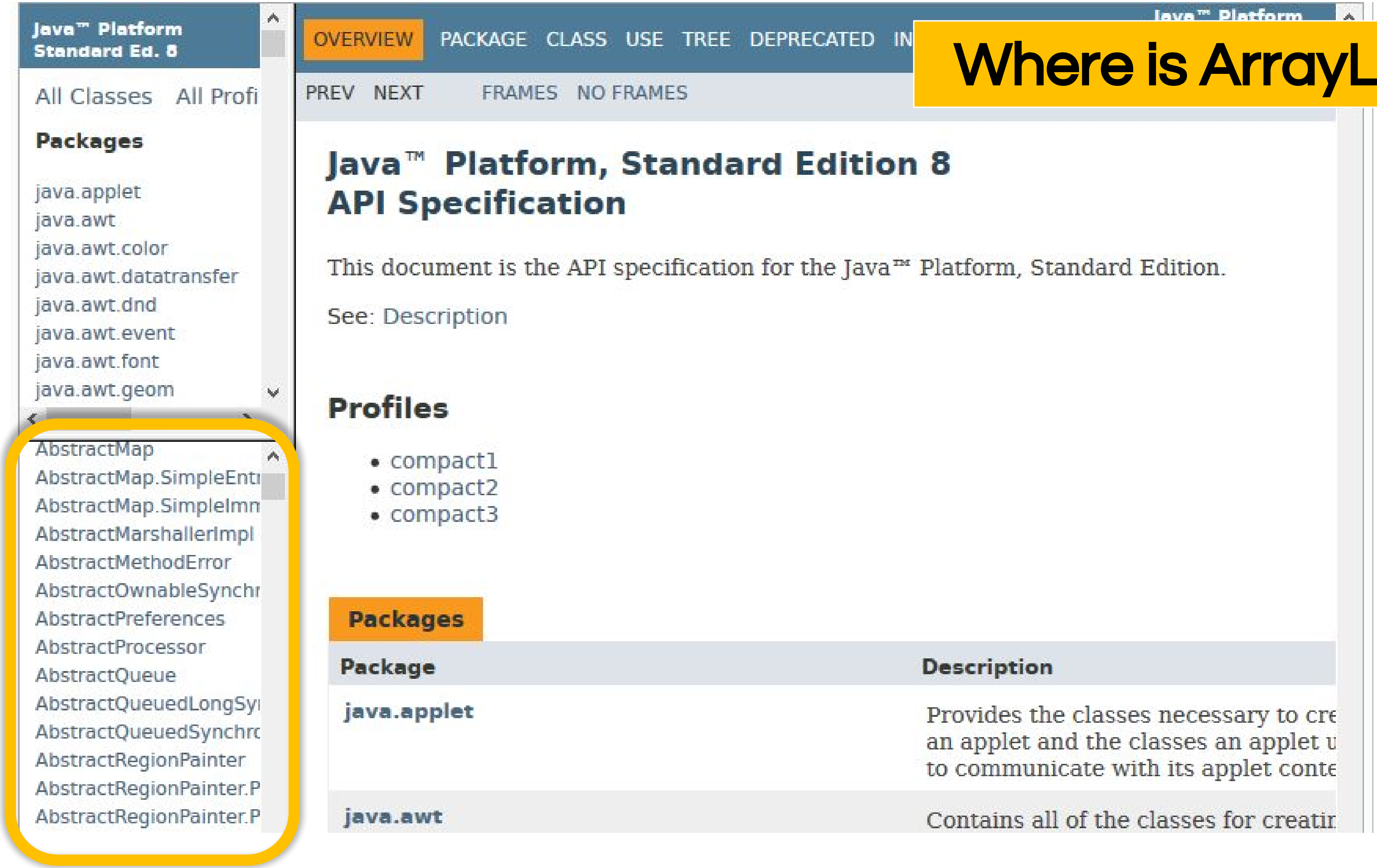
```
List<Feature> countries = new ArrayList<Feature>();
```

What can an ArrayList do?

https://docs.oracle.com/javase/8/docs/api/

Match java version

Where is ArrayList?



Java™ Platform Standard Ed. 8

All Classes All Profi

Packages

java.applet
java.awt
java.awt.color
java.awt.datatransfer
java.awt.dnd
java.awt.event
java.awt.font
java.awt.geom

AbstractMap
AbstractMap.SimpleEntı
AbstractMap.SimpleImn
AbstractMarshallerImpl
AbstractMethodError
AbstractOwnableSynchı
AbstractPreferences
AbstractProcessor
AbstractQueue
AbstractQueuedLongSyı
AbstractQueuedSynchrc
AbstractRegionPainter
AbstractRegionPainter.P
AbstractRegionPainter.P

Java™ Platform, Standard Edition 8 API Specification

This document is the API specification for the Java™ Platform, Standard Edition.

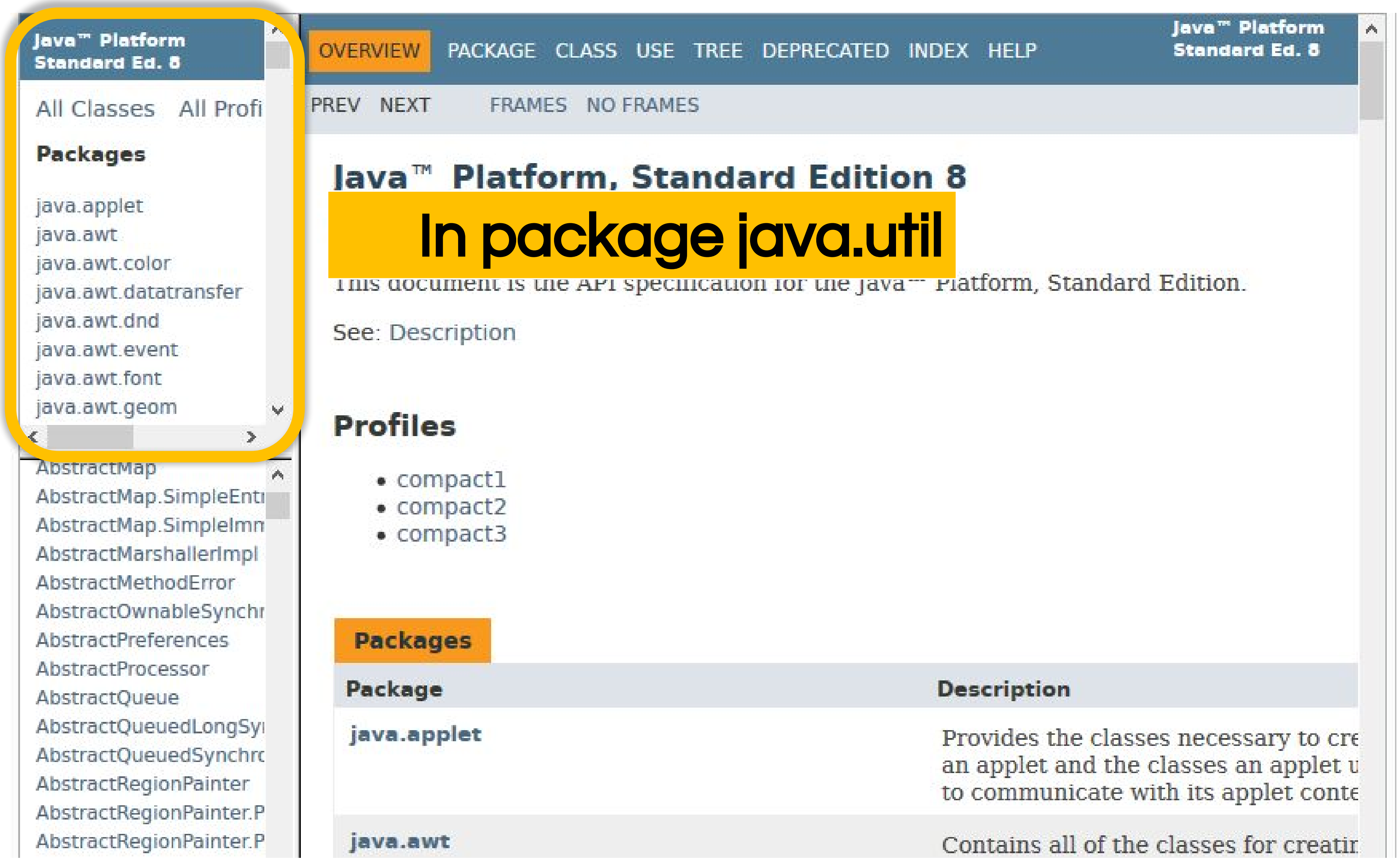
See: Description

Profiles

- compact1
- compact2
- compact3

Packages

Package	Description
java.applet	Provides the classes necessary to cre an applet and the classes an applet u to communicate with its applet conte
java.awt	Contains all of the classes for creatir



Packages

java.applet
java.awt
java.awt.color
java.awt.datatransfer
java.awt.dnd
java.awt.event
java.awt.font
java.awt.geom

AbstractMap
AbstractMap.SimpleEntı
AbstractMap.SimpleImm
AbstractMarshallerImpl
AbstractMethodError
AbstractOwnableSynchr
AbstractPreferences
AbstractProcessor
AbstractQueue
AbstractQueuedLongSy
AbstractQueuedSynchrc
AbstractRegionPainter
AbstractRegionPainter.P
AbstractRegionPainter.P

Java™ Platform, Standard Edition 8

In package java.util

This document is the API specification for the Java™ Platform, Standard Edition.

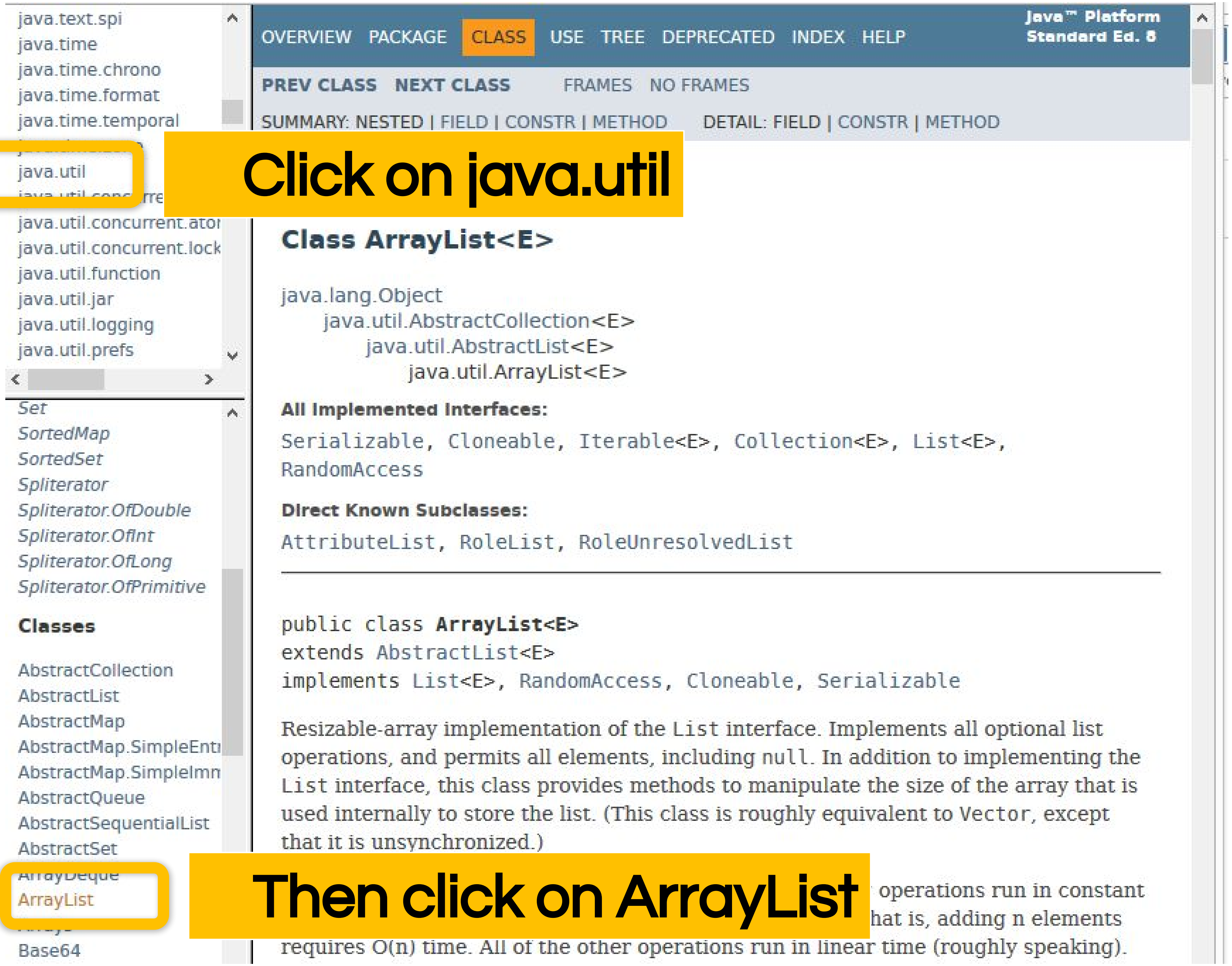
See: Description

Profiles

- compact1
- compact2
- compact3

Packages

Package	Description
java.applet	Provides the classes necessary to cre an applet and the classes an applet u to communicate with its applet conte
java.awt	Contains all of the classes for creatir



Click on java.util

java.text.spi
java.time
java.time.chrono
java.time.format
java.time.temporal
java.util
java.util.concurrent.ator
java.util.concurrent.lock
java.util.function
java.util.jar
java.util.logging
java.util.prefs

Set
SortedMap
SortedSet
Spliterator
Spliterator.OfDouble
Spliterator.OfInt
Spliterator.OfLong
Spliterator.OfPrimitive

Classes

AbstractCollection
AbstractList
AbstractMap
AbstractMap.SimpleEntı
AbstractMap.SimpleImn
AbstractQueue
AbstractSequentialList
AbstractSet
ArrayDeque
ArrayList
Base64

OVERVIEW PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

Java™ Platform Standard Ed. 8

PREV CLASS **NEXT CLASS** FRAMES NO FRAMES

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Class ArrayList<E>

java.lang.Object
    java.util.AbstractCollection<E>
        java.util.AbstractList<E>
            java.util.ArrayList<E>

All Implemented Interfaces:

`Serializable, Cloneable, Iterable<E>, Collection<E>, List<E>, RandomAccess`

Direct Known Subclasses:

`AttributeList, RoleList, RoleUnresolvedList`

```
public class ArrayList<E>
extends AbstractList<E>
implements List<E>, RandomAccess, Cloneable, Serializable
```

Resizable-array implementation of the `List` interface. Implements all optional list operations, and permits all elements, including `null`. In addition to implementing the `List` interface, this class provides methods to manipulate the size of the array that is used internally to store the list. (This class is roughly equivalent to `Vector`, except that it is unsynchronized.)

operations run in constant hat is, adding n elements requires O(n) time. All of the other operations run in linear time (roughly speaking).

Then click on ArrayList

ArrayLists are like Arrays

get an element
at an index

Array version:

```
Feature f = countryArray[0];
```

must have at
least 1 element

ArrayLists are like Arrays

get an element at an index

Array version:

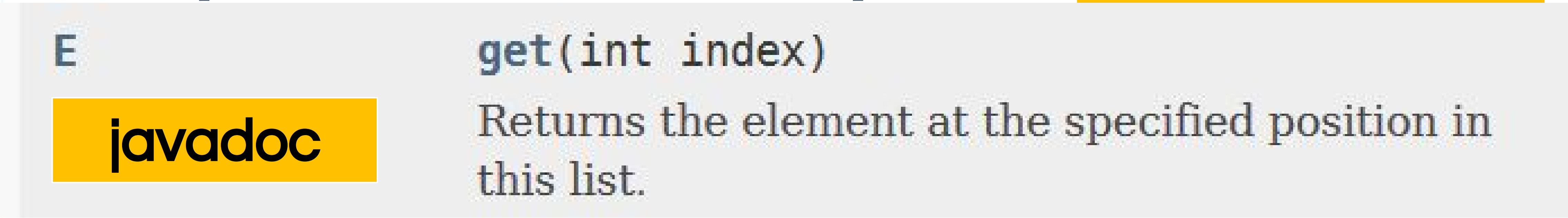
```
Feature f = countryArray[0];
```

```
Feature f = countries.get(0);
```

must have at least 1 element

ArrayLists are like Arrays

get an element at an index



javadoc

E get(int index)

Returns the element at the specified position in this list.

```
Feature f = countries.get(0);
```

ArrayLists are like Arrays

set an element at an index

Array version:

```
countryArray[0] = f;
```

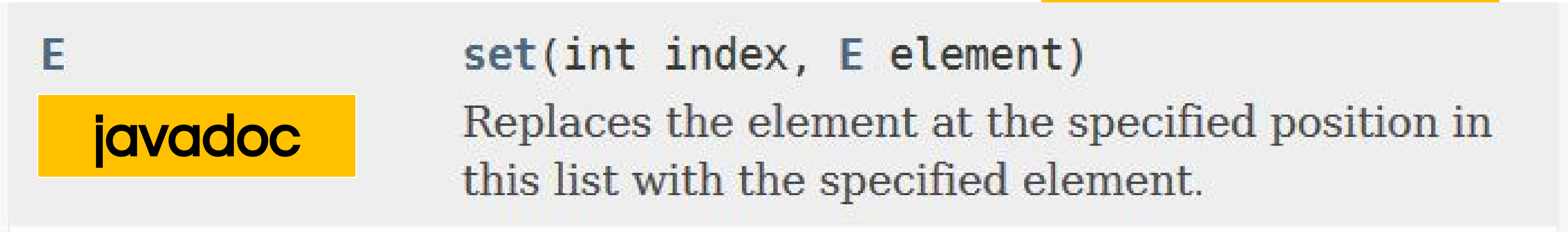
Feature

```
countries.set(0, f);
```

must have at least 1 element

ArrayLists are like Arrays

set an element at an index



E

`set(int index, E element)`

Replaces the element at the specified position in this list with the specified element.

javadoc

```
countries.set(f);
```

ArrayLists are like Arrays

get the number of elements

Array version:

```
int len = countryArray.length;
```

```
int len = countries.size();
```

ArrayLists are like Arrays

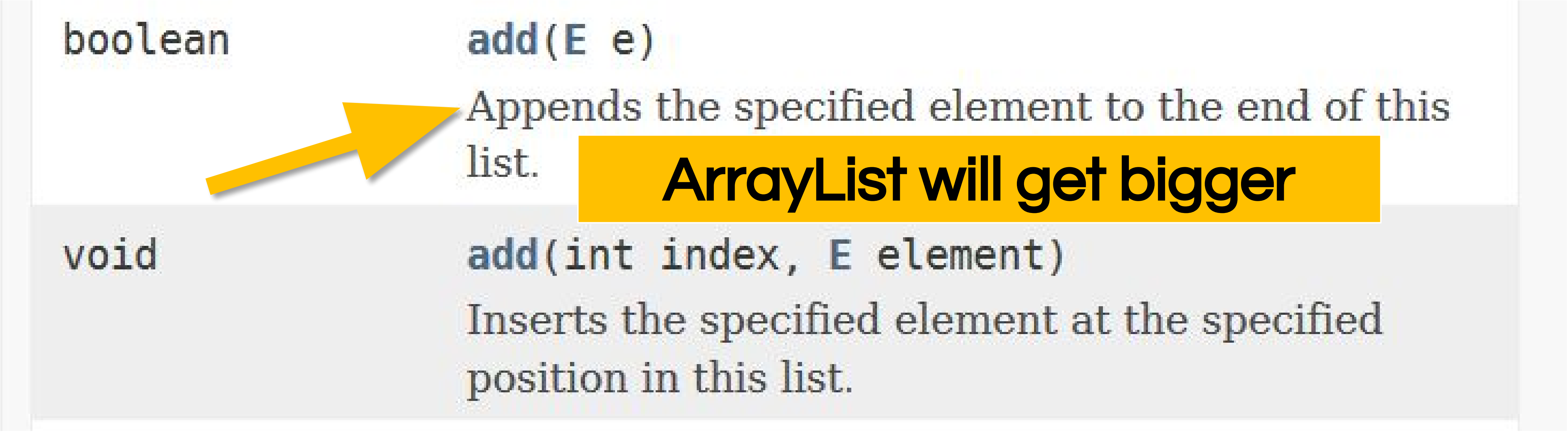
get the number of elements

Array version:

```
int len = countryArray.length;
```

```
int len = countries.size();
```

ArrayLists are resizable (arrays are not)



boolean	add(E e) Appends the specified element to the end of this list.
void	add(int index, E element) Inserts the specified element at the specified position in this list.

ArrayList will get bigger

```
public class LifeExpectancy extends PApplet
{
  UnfoldingMap map;
  Map<String, Float> lifeExpMap;

  List<Feature> countries;
  List<Marker> countryMarkers;
   ...
}
```

Generics

Lists and ArrayLists declare what type they store

```
List<Feature> countries = new ArrayList<Feature>();
List<Marker> countryMarkers = new ArrayList<Marker>();
...
```

Types match

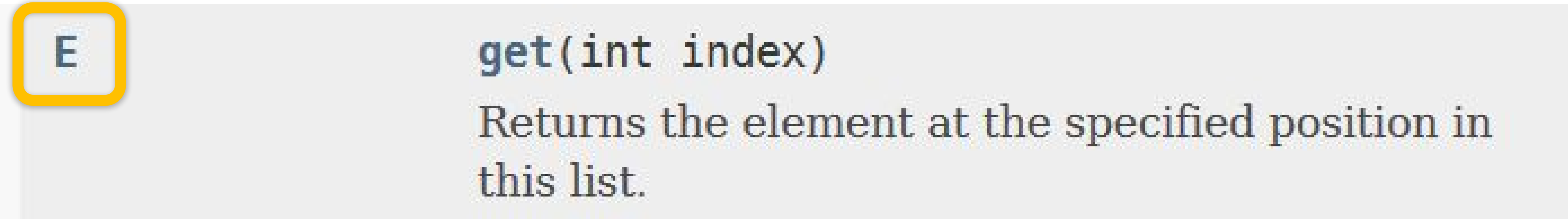
```
List<Feature> countries = new ArrayList<Feature>();
List<Marker> countryMarkers = new ArrayList<Marker>();
...

Feature f = countries.get(0);
Marker m = countryMarkers.get(0);
```

Code to add elements

```
List<Feature> countries = new ArrayList<Feature>();
List<Marker> countryMarkers = new ArrayList<Marker>();
...

Feature f = countries.get(0);
Marker m = countryMarkers.get(0);
```

```
List<Feature> countries = new ArrayList<Feature>();
List<Marker> countryMarkers = new ArrayList<Marker>();
...

Feature f = countries.get(0);
Marker m = countryMarkers.get(0);
```

```
List<Feature> countries = new ArrayList<Feature>();
List<Marker> countryMarkers = new ArrayList<Marker>();
...

Feature f = countries.get(0);
Marker m = countryMarkers.get(0);
```



E

```
get(int index)
```

Returns the element at the specified position in this list.