Hamcrest 1.3 *Quick Reference*

Core Library

General purpose

```
is(T)
equalTo(T)
not(T)                                              : Matcher<T>
```

```
anything()
anything(String)                               : Matcher<Object>
```

```
any(Class<T>)
instanceOf(Class<?>)
isA(Class<T>)                                       : Matcher<T>
```

```
nullValue()                                    : Matcher<Object>
nullValue(Class<T>)                                 : Matcher<T>
notNullValue()                                 : Matcher<Object>
notNullValue(Class<T>)                              : Matcher<T>
```

```
sameInstance(T)
theInstance(T)                                      : Matcher<T>
```

```
isIn(Collection<T>)
isIn(T[])
isOneOf(T...)
hasToString(String)
hasToString(Matcher<? super String>)                : Matcher<T>
```

Combining multiple matchers

```
is(Matcher<T>)
not(Matcher<T>)                                     : Matcher<T>
```

```
allOf(Matcher<? super T>...)
allOf(Iterable<Matcher<? super T>>)
anyOf(Matcher<? super T>...)
anyOf(Iterable<Matcher<? super T>>)                 : Matcher<T>
```

```
both(Matcher<? super LHS>)
either(Matcher<? super LHS>)                      : Matcher<LHS>
```

```
describedAs(String, Matcher<T>, Object...)          : Matcher<T>
```

Strings

```
containsString(String)
startsWith(String)
endsWith(String)                               : Matcher<String>
```

```
equalToIgnoringCase(String)
equalToIgnoringWhiteSpace(String)              : Matcher<String>
```

```
isEmptyString()
isEmptyOrNullString()                          : Matcher<String>
```

```
stringContainsInOrder(Iterable<String>)        : Matcher<String>
```

Iterables

```
everyItem(Matcher<U>)                     : Matcher<Iterable<U>>
```

```
hasItem(T)
hasItem(Matcher<? super T>)  : Matcher<Iterable<? super T>>
```

```
hasItems(T...)
hasItems(Matcher<? super T>...)           : Matcher<Iterable<T>>
```

```
emptyIterable()                 : Matcher<Iterable<? extends E>>
emptyIterableOf(Class<E>)                 : Matcher<Iterable<E>>
```

```
contains(E...)
contains(Matcher<? super E>...)
contains(Matcher<? super E>)
contains(List<Matcher<? super E>>)
                                : Matcher<Iterable<? extends E>>
```

```
containsInAnyOrder(T...)
containsInAnyOrder(Collection<Matcher<? super T>>)
containsInAnyOrder(Matcher<? super T>...)
containsInAnyOrder(Matcher<? super E>)
                                : Matcher<Iterable<? extends E>>
```

```
iterableWithSize(Matcher<? super Integer>)
iterableWithSize(int)                     : Matcher<Iterable<E>>
```

Collections

```
hasSize(int)
hasSize(Matcher<? super Integer>)
                              : Matcher<Collection<? extends E>>
```

```
empty()                       : Matcher<Collection<? extends E>>
emptyCollectionOf(Class<E>)             : Matcher<Collection<E>>
```

Arrays

```
array(Matcher<? super T>...)                      : Matcher<T[]>
```

```
hasItemInArray(T)
hasItemInArray(Matcher<? super T>)                : Matcher<T[]>
```

```
arrayContaining(E...)
arrayContaining(List<Matcher<? super E>>)
arrayContaining(Matcher<? super E>...)            : Matcher<E[]>
```

```
arrayContainingInAnyOrder(E...)
arrayContainingInAnyOrder(Matcher<? super E>...)
arrayContainingInAnyOrder(Collection<Matcher<? super E>>)
                                                  : Matcher<E[]>
```

```
arrayWithSize(int)
arrayWithSize(Matcher<? super Integer>)
emptyArray()                                      : Matcher<E[]>
```

Maps

```
hasEntry(K, V)
hasEntry(Matcher<? super K>, Matcher<? super V>)
                    : Matcher<Map<? extends K, ? extends V>>
```

```
hasKey(K)
hasKey(Matcher<? super K>)   : Matcher<Map<? extends K, ?>>
```

```
hasValue(V)
hasValue(Matcher<? super V>) : Matcher<Map<?, ? extends V>>
```

Beans

```
hasProperty(String)
hasProperty(String, Matcher<?>)
samePropertyValuesAs(T)                             : Matcher<T>
```

Comparables

```
comparesEqualTo(T extends Comparable<T>)
greaterThan(T extends Comparable<T>)
greaterThanOrEqualTo(T extends Comparable<T>)
lessThan(T extends Comparable<T>)
lessThanOrEqualTo(T extends Comparable<T>)          : Matcher<T>
```

Numbers

```
closeTo(double, double)                        : Matcher<Double>
closeTo(BigDecimal, BigDecimal)            : Matcher<BigDecimal>
```

Classes

```
typeCompatibleWith(Class<T>)  : Matcher<java.lang.Class<?>>
```

EventObjects

```
eventFrom(Object)
eventFrom(Class<? extends EventObject>, Object)
                                          : Matcher<EventObject>
```

DOM

```
hasXPath(String)
hasXPath(String, NamespaceContext)
hasXPath(String, Matcher<String>)
hasXPath(String, NamespaceContext, Matcher<String>)
                                        : Matcher<org.w3c.dom.Node>
```

Created by Marc Philipp, http://www.marcphilipp.de