

# Testing cheat sheet

## Finders

```
onNode(matcher)
onNodeWithContentDescription
onNodeWithTag
onNodeWithText
onRoot
```

```
onAllNodes(matcher)
onAllNodesWithContentDescription
onAllNodesWithTag
onAllNodesWithText
```

**OPTIONS:** useUnmergedTree: Boolean

## Matchers

```
has[No]ClickAction
hasContentDescription[Exactly]
hasImeAction
hasProgressBarRangeInfo
has[No]ScrollAction
hasSetTextAction
hasStateDescription
hasTestTag
hasText[Exactly]
is[Not]Dialog
is[Not]Enabled
is[Not]Focused
is[Not]Selected
isHeading
isOff
isOn
isPopup
isSelectable
isToggleable
isFocusable
isRoot
```

### HIERARCHICAL

```
hasParent
hasAnyChild
hasAnySibling
hasAnyDescendant
hasAnyAncestor
```

### SELECTORS

```
filter(matcher)
filterToOne(matcher)
onAncestors
onChild
onChildAt
onChildren
onFirst
onLast
onParent
onSibling
onSiblings
```

## Assertions

```
assert(matcher)
assertExists
assertDoesNotExist
assertContentDescriptionContains
assertContentDescriptionEquals
assertIs[Not]Displayed
assertIs[Not]Enabled
assertIs[Not]Selected
assertIs[Not]Focused
assertIsOn
assertIsOff
assertIsToggleable
assertIsSelectable
assertTextEquals
assertTextContains
assertValueEquals
assertRangeInfoEquals
assertHas[No]ClickAction
```

### COLLECTIONS

```
assertAll
assertAny
assertCountEquals(Int)
```

### BOUNDS

```
assertWidthIsEqualTo
assertHeightIsEqualTo
assertWidthIsAtLeast
assertHeightIsAtLeast
assertTopPositionInRootIsEqualTo
assertLeftPositionInRootIsEqualTo
getAlignmentLinePosition(BaseLine)
getUnclippedBoundsInRoot
```

## Actions

```
performClick
performGesture
performScrollTo
performSemanticsAction
performKeyPress
performImeAction
performTextClearance
performTextInput
performTextReplacement
```

### GESTURES

```
click
doubleClick
longClick
pinch
swipe
swipe[Down|Left|Right|Up]
swipeWithVelocity
```

### PARTIAL GESTURES

```
down
moveTo
movePointerTo
moveBy
movePointerBy
move
up
cancel
```

## ComposeTestRule

```
@get:Rule
val testRule =
    createComposeRule()
```

```
setContent { }
density
runOnIdle { }
runOnUiThread { }
waitForIdle()
waitUntil { }
awaitIdle()
[un]registerIdlingResource()
mainClock.autoAdvance
mainClock.currentTime
mainClock.advanceTimeBy()
mainClock.advanceTimeByFrame()
mainClock.advanceTimeUntil { }
```

## AndroidComposeTestRule

```
@get:Rule
val testRule =
    createAndroidComposeRule<Activity>()
```

```
ComposeTestRule.* +
activity
activityRule
```

## Debug

```
onNode(...).*
```

```
printToString()
printToLog()
captureToImage()
```