

Get screen width and height in Android

Asked 11 years, 8 months ago Modified 6 months ago Viewed 708k times



How can I get the screen width and height and use this value in:

595

```
@Override protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {
    Log.e(TAG, "onMeasure" + widthMeasureSpec);
    setMeasuredDimension(SCREEN_WIDTH, SCREEN_HEIGHT -
        game.findViewById(R.id.flag).getHeight());
}
```

148



android

ShareFollow

edited Feb 24, 2019 at 18:20

asked Jan 20, 2011 at 3:17



0xCursor

2,236 4 14 32



rebel-UA

6,097 3 20 19

2 see: stackoverflow.com/questions/1016896/... – Will Tate Jan 20, 2011 at 3:18

2 Does this answer your question? [How to get screen dimensions as pixels in Android](#) – AdamHurwitz Aug 20, 2020 at 0:24

the update correct answer stackoverflow.com/a/67691009/4797289 – Rasoul Miri May 25, 2021 at 15:24

Sorted by:

34 Answers

Highest score (default)

12Next



Using this code, you can get the runtime display's width & height:

1190

```
DisplayMetrics displayMetrics = new DisplayMetrics();
getWindowManager().getDefaultDisplay().getMetrics(displayMetrics);
int height = displayMetrics.heightPixels;
int width = displayMetrics.widthPixels;
```



In a view you need to do something like this:

```
((Activity) getContext()).getWindowManager()
    .getDefaultDisplay()
    .getMetrics(displayMetrics);
```

In some scenarios, where devices have a navigation bar, you have to check at runtime:

```
public boolean showNavigationBar(Resources resources)
{
    int id = resources.getIdentifier("config_showNavigationBar", "bool", "android");
```

```

return id > 0 && resources.getBoolean(id);
}

```

If the device has a navigation bar, then count its height:

```

private int getNavigationBarHeight() {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.JELLY_BEAN_MR1) {
        DisplayMetrics metrics = new DisplayMetrics();
        getWindowManager().getDefaultDisplay().getMetrics(metrics);
        int usableHeight = metrics.heightPixels;
        getWindowManager().getDefaultDisplay().getRealMetrics(metrics);
        int realHeight = metrics.heightPixels;
        if (realHeight > usableHeight)
            return realHeight - usableHeight;
        else
            return 0;
    }
    return 0;
}

```

So the final height of the device is:

```
int height = displayMetrics.heightPixels + getNavigationBarHeight();
```

ShareFollow

edited Jul 9, 2019 at 1:28

answered Jan 20, 2011 at 7:28



OxCursor

2,236 4 14 32



Parag Chauhan

35.3k 13 85 95

40 Are the width and height swapped if the device is rotated? – Madeyedexter Nov 28, 2016 at 6:51

16 @Madedyedexter yes it will. – Parag Chauhan Nov 28, 2016 at 7:08

1 But you can no longer assume you are running on the default display. – satur9nine Jun 5, 2018 at 22:30

4 or just simply use - getResources().getDisplayMetrics().heightPixels / getResources().getDisplayMetrics().widthPixels – Alex Jun 27, 2018 at 17:09 ✎

12 getDefaultDisplay() and getMetrics are deprecated – Denis Aug 14, 2020 at 7:55



There is a very simple answer and without pass context

370

```

public static int getScreenWidth() {
    return Resources.getSystem().getDisplayMetrics().widthPixels;
}

```



```

public static int getScreenHeight() {
    return Resources.getSystem().getDisplayMetrics().heightPixels;
}

```

Note: if you want the height include navigation bar, use method below

```

WindowManager windowManager =
    (WindowManager)
    BaseApplication.getApplication().getSystemService(Context.WINDOW_SERVICE);
    final Display display = windowManager.getDefaultDisplay();
    Point outPoint = new Point();
    if (Build.VERSION.SDK_INT >= 19) {
        // include navigation bar
        display.getRealSize(outPoint);
    } else {
        // exclude navigation bar
        display.getSize(outPoint);
    }
    if (outPoint.y > outPoint.x) {
        mRealSizeHeight = outPoint.y;
        mRealSizeWidth = outPoint.x;
    } else {
        mRealSizeHeight = outPoint.x;
        mRealSizeWidth = outPoint.y;
    }
}

```

ShareFollow

edited Mar 8, 2018 at 12:05

answered Jul 13, 2015 at 7:27



weigan

4,447 1 12 14

3 How to get total screen height? This version of `getScreenHeight` excludes all bars. – Jaydev Dec 30, 2016 at 11:45

1 this works in most cases, but it can't get the real screen height if the navigation bar is visible. the height excludes the bar's height. – L. Swifter Mar 9, 2017 at 6:43

This produced unusual results for me in landscape mode – Carson Holzheimer Nov 13, 2017 at 3:34

1 This will cause trouble on screens with multiple displays (Foldables, Chrome OS laptops)
– EpicPandaForce Nov 12, 2018 at 9:31

The height of the navigation bar is not included in Android 4.4 – Hemant Aggarwal Nov 30, 2018 at 11:56

Just to update the answer by parag and SpK to align with current SDK backward compatibility from deprecated methods:

47

```
int Measuredwidth = 0;
```

```
int Measuredheight = 0;
```

```
Point size = new Point();
```

```
WindowManager w = getWindowManager();
```

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
```

```
    w.getDefaultDisplay().getSize(size);
```

```
    Measuredwidth = size.x;
```

```
    Measuredheight = size.y;
```

```
}else{
```

```
    Display d = w.getDefaultDisplay();
```

```
Measuredwidth = d.getWidth();
Measuredheight = d.getHeight();
}
```

ShareFollow

edited Jul 19, 2013 at 6:28



dakshbhatt21

3,514 3 30 38

answered Jul 24, 2012 at 3:24



digiphd

2,299 3 23 22

7 @digiphd wouldn't Parang's code work on all API versions? All methods were introduced before API level 8 and I didn't find them deprecated at the android dev site. – Sufian May 22, 2013 at 12:52 ✎

@Sufian I had trouble getting non-0 return values from the `getSize` implementation on a Samsung Galaxy Mini (API 10) but the deprecated methods in this answer return correctly. So this is useful for older versions of Android. – Damien Diehl Aug 9, 2016 at 18:44 ✎

@Sufian can we set new dimension to screen width and height? – Srishti Roy Jan 18, 2018 at 5:39

1 this is depreciated now :(– cegprakash Dec 30, 2018 at 12:10

▲ Why not

27

DisplayMetrics displaymetrics = getResources().getDisplayMetrics();



🕒 then use

displayMetrics.widthPixels (heightPixels)

ShareFollow

answered Mar 25, 2014 at 6:47



wangqi060934

1,494 16 23

On some devices `widthPixels` or `heightPixels` returns 0 in a `Fragment`'s `onViewCreated()`. Really annoying if you want to calculate span count for `GridLayoutManager`. – Bram Stoker Sep 15, 2020 at 8:20 ✎

▲ It's very easy to get in Android:

25

```
int width = Resources.getSystem().getDisplayMetrics().widthPixels;
int height = Resources.getSystem().getDisplayMetrics().heightPixels;
```



ShareFollow

edited Feb 11, 2019 at 10:08



Eric Aya

69.2k 34 176 246

answered Feb 11, 2019 at 10:04



Ness Tyagi

1,940 23 18

2 this returns 0 on an emulator – jonadv Feb 9, 2020 at 20:39

▲ • Kotlin Version via Extension Property

20

If you want to know the size of the screen in pixels as well as dp, using these extension properties really helps:



🕒 [DimensionUtils.kt](#)

```
import android.content.Context
import android.content.res.Resources
```

```
import android.graphics.Rect
import android.graphics.RectF
import android.os.Build
import android.util.DisplayMetrics
import android.view.WindowManager
import kotlin.math.roundToInt

/**
 * @author aminography
 */

private val displayMetrics: DisplayMetrics by lazy { Resources.getSystem().displayMetrics }

/**
 * Returns boundary of the screen in pixels (px).
 */
val screenRectPx: Rect
    get() = displayMetrics.run { Rect(0, 0, widthPixels, heightPixels) }

/**
 * Returns boundary of the screen in density independent pixels (dp).
 */
val screenRectDp: RectF
    get() = screenRectPx.run { RectF(0f, 0f, right.px2dp, bottom.px2dp) }

/**
 * Returns boundary of the physical screen including system decor elements (if any) like navigation
 * bar in pixels (px).
 */
val Context.physicalScreenRectPx: Rect
    get() = if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.JELLY_BEAN_MR1) {
        (applicationContext.getSystemService(Context.WINDOW_SERVICE) as
        WindowManager)
            .run { DisplayMetrics().also { defaultDisplay.getRealMetrics(it) } }
            .run { Rect(0, 0, widthPixels, heightPixels) }
    } else screenRectPx

/**
 * Returns boundary of the physical screen including system decor elements (if any) like navigation
 * bar in density independent pixels (dp).
 */
val Context.physicalScreenRectDp: RectF
    get() = physicalScreenRectPx.run { RectF(0f, 0f, right.px2dp, bottom.px2dp) }

/**
 * Converts any given number from pixels (px) into density independent pixels (dp).
 */
val Number.px2dp: Float
    get() = this.toFloat() / displayMetrics.density
```

```
/**
 * Converts any given number from density independent pixels (dp) into pixels (px).
 */
val Number.dp2px: Int
    get() = (this.toFloat() * displayMetrics.density).roundToInt()
```

Usage:

```
class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val widthPx = screenRectPx.width()
        val heightPx = screenRectPx.height()
        println("[PX] screen width: $widthPx , height: $heightPx")

        val widthDp = screenRectDp.width()
        val heightDp = screenRectDp.height()
        println("[DP] screen width: $widthDp , height: $heightDp")

        println()

        val physicalWidthPx = physicalScreenRectPx.width()
        val physicalHeightPx = physicalScreenRectPx.height()
        println("[PX] physical screen width: $physicalWidthPx , height: $physicalHeightPx")

        val physicalWidthDp = physicalScreenRectDp.width()
        val physicalHeightDp = physicalScreenRectDp.height()
        println("[DP] physical screen width: $physicalWidthDp , height: $physicalHeightDp")
    }
}
```

Result:

When the device is in **portrait** orientation:

```
[PX] screen width: 1440 , height: 2392
[DP] screen width: 360.0 , height: 598.0
```

```
[PX] physical screen width: 1440 , height: 2560
[DP] physical screen width: 360.0 , height: 640.0
```

When the device is in **landscape** orientation:

[PX] screen width: 2392 , height: 1440

[DP] screen width: 598.0 , height: 360.0

[PX] physical screen width: 2560 , height: 1440

[DP] physical screen width: 640.0 , height: 360.0

ShareFollow

edited Apr 3, 2021 at 7:44

answered Aug 12, 2020 at 6:23



aminography

21k 13 66 73

1 This code fails to take into account the height of the navigation bar. – Johann Apr 3, 2021 at 6:29

1 @AndroidDev: Thank you for reporting the issue. Please use `Context.physicalScreenRectPx` and `Context.physicalScreenRectDp` to get the real dimension of the screen. – aminography Apr 3, 2021 at 7:24

▲ You can get width and height from **context**

18 **java:**

▼ int width= context.getResources().getDisplayMetrics().widthPixels;
🕒 int height= context.getResources().getDisplayMetrics().heightPixels;

kotlin

```
val width: Int = context.resources.displayMetrics.widthPixels
val height: Int = context.resources.displayMetrics.heightPixels
```

ShareFollow

answered May 25, 2021 at 15:23



Rasoul Miri

8,925 1 61 71

▲ Try below code :-

14 **1.**

▼ Display display = getWindowManager().getDefaultDisplay();
🕒 Point size = new Point();
display.getSize(size);
int width = size.x;
int height = size.y;

2.

```
Display display = getWindowManager().getDefaultDisplay();
int width = display.getWidth(); // deprecated
int height = display.getHeight(); // deprecated
```

or

```
int width = getWindowManager().getDefaultDisplay().getWidth();
int height = getWindowManager().getDefaultDisplay().getHeight();
```

3.

```
DisplayMetrics metrics = new DisplayMetrics();
getWindowManager().getDefaultDisplay().getMetrics(metrics);
```

```
metrics.heightPixels;
metrics.widthPixels;
```

ShareFollow

edited Sep 26, 2013 at 4:43

answered May 7, 2013 at 10:24



duggu

37.4k 12 114 111

3 don't you think 2 = 4 – TrikalDarshiii Aug 15, 2013 at 17:48

Means your method 2 is equal to method 4 – TrikalDarshiii Sep 24, 2013 at 13:54

Is it possible to get the "natural" resolution of the device? meaning that no matter if you rotate the screen, you will get the same values for width&height, and that if it's a tablet that's meant to be used horizontally, it will return the width as the large one instead of the height ? – android developer May 10, 2015 at 12:16

```
DisplayMetrics IDisplayMetrics = getResources().getDisplayMetrics();
```

```
int widthPixels = IDisplayMetrics.widthPixels;
```

10 int heightPixels = IDisplayMetrics.heightPixels;

ShareFollow

edited Dec 17, 2015 at 11:48

answered Dec 17, 2015 at 10:12



Micer

8,355 3 78 70

Shubham

501 4 11

For kotlin user's

```
8 fun Activity.displayMetrics(): DisplayMetrics {
    val displayMetrics = DisplayMetrics()
    windowManager.defaultDisplay.getMetrics(displayMetrics)
    return displayMetrics
}
```

And in Activity you could use it like

```
resources.displayMetrics.let { displayMetrics ->
    val height = displayMetrics.heightPixels
    val width = displayMetrics.widthPixels
}
```

Or in fragment

```
activity?.displayMetrics()?.run {
    val height = heightPixels
    val width = widthPixels
}
```

ShareFollow

edited Nov 25, 2019 at 15:29

answered Nov 20, 2018 at 7:27



Silvia H

7,847 6 30 33



Ali mohammadi

1,701 23 27

Some methods, applicable for retrieving screen size, are deprecated in **API Level 31**, including `Display.getRealMetrics()` and `Display.getRealSize()`. Starting from **API Level 30** we can use `WindowManager#getCurrentWindowMetrics()`. The clean way to get screen size is to create some `Compat` class, e.g.:



```
object ScreenMetricsCompat {
    private val api: Api =
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.R) ApiLevel30()
        else Api()

    /**
     * Returns screen size in pixels.
     */
    fun getScreenSize(context: Context): Size = api.getScreenSize(context)

    @Suppress("DEPRECATION")
    private open class Api {
        open fun getScreenSize(context: Context): Size {
            val display = context.getSystemService(WindowManager::class.java).defaultDisplay
            val metrics = if (display != null) {
                DisplayMetrics().also { display.getRealMetrics(it) }
            } else {
                Resources.getSystem().displayMetrics
            }
            return Size(metrics.widthPixels, metrics.heightPixels)
        }
    }

    @RequiresApi(Build.VERSION_CODES.R)
    private class ApiLevel30 : Api() {
        override fun getScreenSize(context: Context): Size {
            val metrics: WindowMetrics =
                context.getSystemService(WindowManager::class.java).currentWindowMetrics
            return Size(metrics.bounds.width(), metrics.bounds.height())
        }
    }
}
```

Calling `ScreenMetricsCompat.getScreenSize(this).height` in Activity we can get a screen height.

ShareFollow

edited Feb 22 at 14:54

answered Nov 23, 2021 at 20:27



Sergio

22.4k 5 71 96

In class `Api` you can use `ContextCompat.getSystemService(context, WindowManager::class.java)?.defaultDisplay` not `context.getSystemService(WindowManager::class.java).defaultDisplay` to target APIs less than 23 safely – Islam Khaled Jun 15 at 17:40

▲ As `getMetrics` and `getRealMetrics` are deprecated, Google recommends to determine the screen width and height as follows:

7

```
WindowMetrics windowMetrics =
```

▼ `getActivity().getWindowManager().getMaximumWindowMetrics();`

🕒 `Rect bounds = windowMetrics.getBounds();`

```
int widthPixels = bounds.width();
```

```
int heightPixels = bounds.height();
```

However, I've figured out another method that gives me the same results:

```
Configuration configuration = mContext.getResources().getConfiguration();
```

```
Display.Mode mode = display.getMode();
```

```
int widthPixels = mode.getPhysicalWidth();
```

```
int heightPixels = mode.getPhysicalHeight();
```

ShareFollow

answered Jul 9, 2021 at 21:02



Patrick

101 1 9

This is the only answer that works in Android R – SemicolonSpace Feb 6 at 5:52

▲ I suggest you create extension functions.

7

```
/**
```

```
 * Return the width and height of the screen
```

```
 */
```



```
val Context.screenWidth: Int
```



```
get() = resources.displayMetrics.widthPixels
```

```
val Context.screenHeight: Int
```

```
get() = resources.displayMetrics.heightPixels
```

```
/**
```

```
 * Pixel and Dp Conversion
```

```
 */
```

```
val Float.toPx get() = this * Resources.getSystem().displayMetrics.density
```

```
val Float.toDp get() = this / Resources.getSystem().displayMetrics.density
```

```
val Int.toPx get() = (this * Resources.getSystem().displayMetrics.density).toInt()
```

```
val Int.toDp get() = (this / Resources.getSystem().displayMetrics.density).toInt()
```

ShareFollow

answered Mar 9 at 14:25



Mattia Ferigutti

1,458 13 20

▲ `DisplayMetrics dimension = new DisplayMetrics();`

```
getWindowManager().getDefaultDisplay().getMetrics(dimension);
```

6

```
int width = dimension.widthPixels;
```

```
int height = dimension.heightPixels;
```



ShareFollow



answered Dec 18, 2013 at 10:28



Cristiana Chavez

▲ Get the value of screen width and height.

6 Display display = getWindowManager().getDefaultDisplay();

Point size = new Point();

▼ display.getSize(size);

🕒 width = size.X;

height = size.y;

ShareFollow

edited Feb 5, 2018 at 6:58



Pang

9,221

146

85

118

answered Jan 21, 2015 at 0:38



Anil Singhania

749

10

8

▲ None of the answers here work correctly for Chrome OS multiple displays, or soon-to-come Foldables.

6



When looking for the current configuration, always use the configuration from your current activity in `getResources().getConfiguration()`. **Do not use the configuration from your background activity or the one from the system resource.** The background activity does not have a size, and **the system's configuration may contain multiple windows with conflicting sizes and orientations**, so no usable data can be extracted.

So the answer is

```
val config = context.getResources().getConfiguration()
```

```
val (screenWidthPx, screenHeightPx) = config.screenWidthDp.dp to
```

```
config.screenHeightDp.dp
```

ShareFollow

answered Nov 12, 2018 at 9:43



EpicPandaForce

76.9k

27

241

410

▲ As an android official document said for the default display use `Context#getDisplay()` because this method was deprecated in API level 30.

6



```
getWindowManager().getDefaultDisplay().getMetrics(displayMetrics);
```

This code given below is in kotlin and is written accodring to the latest version of Android help you determine width and height:

```
fun getWidth(context: Context): Int {
```

```
    var width:Int = 0
```

```
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.R) {
```

```
        val displayMetrics = DisplayMetrics()
```

```
        val display: Display? = context.getDisplay()
```

```
        display!!.getRealMetrics(displayMetrics)
```

```
        return displayMetrics.widthPixels
```

```
    }else{
```

```
        val displayMetrics = DisplayMetrics()
```

```
        this.windowManager.defaultDisplay.getMetrics(displayMetrics)
```

```

        width = displayMetrics.widthPixels
        return width
    }
}

```

```

fun getHeight(context: Context): Int {
    var height: Int = 0
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.R) {
        val displayMetrics = DisplayMetrics()
        val display = context.display
        display!!.getRealMetrics(displayMetrics)
        return displayMetrics.heightPixels
    } else {
        val displayMetrics = DisplayMetrics()
        this.windowManager.defaultDisplay.getMetrics(displayMetrics)
        height = displayMetrics.heightPixels
        return height
    }
}

```

ShareFollow

edited Sep 15, 2020 at 15:24

answered Sep 15, 2020 at 14:50



Mohit Yadav

138 3 10

1 Can you give a complete example for this – Dilshan Sep 22, 2020 at 3:13

1 u can copy the code and call the functions like `var one= getHeight(this)` and the value of the display's height will be stored in the variable one as the functions return the height and width values when called. – Mohit Yadav Sep 22, 2020 at 4:30

```

▲ fun Activity.getRealScreenSize(): Pair<Int, Int> { //<width, height>
  return if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.R) {
6   val size = Point()
    display?.getRealSize(size)
    Pair(size.x, size.y)
  } else {
  ↻   val size = Point()
    windowManager.defaultDisplay.getRealSize(size)
    Pair(size.x, size.y)

  }}

```

This is an extension function and you can use in your activity in this way:

```

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)

    val pair = getRealScreenSize()
    pair.first //to get width

```

```
pair.second //to get height
}
```

ShareFollow

edited Jul 20, 2021 at 9:40

answered Jul 15, 2021 at 19:50



DevWM

114 1 5

Code-only answers are low quality answers. – Taslim Oseni Jul 16, 2021 at 6:53

@TaslimOseni I updated my answer by example to use – DevWM Jul 20, 2021 at 9:42

▲ Full way to do it, that returns the true resolution:

5
▼
🕒

```
WindowManager wm = (WindowManager)
context.getSystemService(Context.WINDOW_SERVICE);
Point size = new Point();
wm.getDefaultDisplay().getRealSize(size);
final int width = size.x, height = size.y;
```

And since this can change on different orientation, here's a solution (in Kotlin), to get it right no matter the orientation:

```
/**
 * returns the natural orientation of the device: Configuration.ORIENTATION_LANDSCAPE or
 * Configuration.ORIENTATION_PORTRAIT .<br></br>
 * The result should be consistent no matter the orientation of the device
 */
@JvmStatic
fun getScreenNaturalOrientation(context: Context): Int {
    //based on : http://stackoverflow.com/a/9888357/878126
    val windowManager = context.getSystemService(Context.WINDOW_SERVICE) as
    WindowManager
    val config = context.resources.configuration
    val rotation = windowManager.defaultDisplay.rotation
    return if ((rotation == Surface.ROTATION_0 || rotation == Surface.ROTATION_180) &&
    config.orientation == Configuration.ORIENTATION_LANDSCAPE || (rotation ==
    Surface.ROTATION_90 || rotation == Surface.ROTATION_270) && config.orientation ==
    Configuration.ORIENTATION_PORTRAIT)
        Configuration.ORIENTATION_LANDSCAPE
    else
        Configuration.ORIENTATION_PORTRAIT
}

/**
 * returns the natural screen size (in pixels). The result should be consistent no matter the orientation of the device
 */
@JvmStatic
fun getScreenNaturalSize(context: Context): Point {
    val screenNaturalOrientation = getScreenNaturalOrientation(context)
    val wm = context.getSystemService(Context.WINDOW_SERVICE) as WindowManager
    val point = Point()
```

```

wm.defaultDisplay.getRealSize(point)
val currentOrientation = context.resources.configuration.orientation
if (currentOrientation == screenNaturalOrientation)
    return point
else return Point(point.y, point.x)
}

```

ShareFollow

edited Feb 17, 2018 at 22:50

answered Dec 6, 2017 at 22:07



android developer

114k 140 703

1214

▲ I use the following code to get the screen dimensions

5

```

getWindow().getDecorView().getWidth()
getWindow().getDecorView().getHeight()

```

ShareFollow

answered Dec 13, 2019 at 8:46



Harsha

1,400 1 18 21

This is not technically the screen dimensions but this is helpful since these dimensions do not contain stuff like status- and bottom bar – hordurh Apr 13, 2021 at 7:49

▲ Display display = ((WindowManager)
this.getSystemService(Context.WINDOW_SERVICE)).getDefaultDisplay();

4

```

int mWidthScreen = display.getWidth();
int mHeightScreen = display.getHeight();

```

ShareFollow

edited Mar 20, 2016 at 14:35

answered Mar 20, 2016 at 14:21



Tunaki

128k 45 322 405

Rahul Yadav

143 6

Deprecated, use DisplayMetrics() – deyanm Apr 22, 2019 at 16:39

▲ public class DisplayInfo {
int screen_height=0, screen_width=0;
4 WindowManager wm;
DisplayMetrics displaymetrics;

⌚ DisplayInfo(Context context) {
getdisplayheightWidth(context);
}

```

void getdisplayheightWidth(Context context) {
    wm = (WindowManager) context.getSystemService(Context.WINDOW_SERVICE);
    displaymetrics = new DisplayMetrics();
    wm.getDefaultDisplay().getMetrics(displaymetrics);
    screen_height = displaymetrics.heightPixels;
    screen_width = displaymetrics.widthPixels;
}

```

```

public int getScreen_height() {
    return screen_height;
}

```

```

    }

    public int getScreen_width() {
        return screen_width;
    }
}

```

ShareFollow

edited Nov 29, 2017 at 5:29



Satendra

6,515 4 25 45

answered Jan 31, 2017 at 14:53



Habibur Rahman Ovie

291 2 6

▲ Methods shown here are deprecated/outdated but this is still working. Require API 13

3 check it out

▼ Display disp= getWindowManager().getDefaultDisplay();

🕒 Point dimensions = new Point();

```
disp.getSize(size);
```

```
int width = size.X;
```

```
int height = size.y;
```

ShareFollow

edited May 13, 2015 at 12:09

answered Jan 19, 2015 at 6:11



Hemant Shori

2,443 1 21 20

▲ As an android official document said for the default display use Context#getDisplay() because this method was deprecated in API level 30.

3

▼ getWindowManager().getDefaultDisplay().getMetrics(displayMetrics);



This bowl of code help to determine width and height.

```

public static int getWidth(Context context) {
    DisplayMetrics displayMetrics = new DisplayMetrics();
    Display display = context.getDisplay();
    if (display != null) {
        display.getRealMetrics(displayMetrics);
        return displayMetrics.widthPixels;
    }
    return -1;
}

```

For the Height:

```

public static int getHeight(Context context) {
    DisplayMetrics displayMetrics = new DisplayMetrics();
    Display display = context.getDisplay();
    if (display != null) {
        display.getRealMetrics(displayMetrics);
        return displayMetrics.heightPixels;
    }
}

```

```
    return -1;
}
ShareFollow
```

answered Aug 20, 2020 at 11:17



Kishan Donga

2,503 2 20 34

Does this require special permission in AndroidManifest.xml – Dilshan Sep 21, 2020 at 14:40

1 No there is not any special permission required for the same. – Kishan Donga Sep 21, 2020 at 14:51

@Krishan Donga Thank you so much for your quick reply.... – Dilshan Sep 21, 2020 at 14:52

▲ Seems like all these answers fail for my Galaxy M51 with Android 11. After doing some research around I found this code :

3

```
WindowMetrics windowmetrics =
```

▼ MainActivity.getWindowManager().getCurrentWindowMetrics();

🕒 Rect rect = windowmetrics.getBounds();

```
int width = rect.right;
```

```
int height = rect.bottom;
```

shows my true device resolution of 1080x2400, the rest only return 810x1800.

ShareFollow

answered Jun 2, 2021 at 9:36



RelativeGames

1,583 2 18 27

Requires API 30 – user1608385 Apr 15 at 21:33

▲ Try this code for **Kotlin**

2

```
val display = windowManager.defaultDisplay
```

```
val size = Point()
```

▼ display.getSize(size)

🕒 var DEVICE_WIDTH = size.x

```
var DEVICE_HEIGHT = size.y
```

ShareFollow

answered Jun 5, 2018 at 8:44



Najib.Nj

3,608 1 24 36

▲ Just use the function below that returns width and height of the screen size as an array of integers

1

```
private int[] getScreenSize(){
```

▼ DisplayMetrics displaymetrics = new DisplayMetrics();

🕒 getWindowManager().getDefaultDisplay().getMetrics(displaymetrics);

```
int h = displaymetrics.heightPixels;
```

```
int w = displaymetrics.widthPixels;
```

```
int[] size={w,h};
```

```
return size;
```

```
}
```


On your onCreate function or button click add the following code to output the screen sizes as shown below

```
int[] screenSize= getScreenSize();
int width=screenSize[0];
int height=screenSize[1];
screenSizes.setText("Phone Screen sizes \n\n width = "+width+" \n Height = "+height);
```

ShareFollow

edited Aug 18, 2017 at 10:32

answered Jan 20, 2017 at 7:44



Tim

40.4k

18

126

142



Daniel Nyamasyo

2,350

1

23

23

I updated answer for Kotlin language!

1

For Kotlin: You should call Window Manager and get metrics. After that easy way.



```
val displayMetrics = DisplayMetrics()
windowManager.defaultDisplay.getMetrics(displayMetrics)
```

```
var width = displayMetrics.widthPixels
```

```
var height = displayMetrics.heightPixels
```

How can we use it effectively in independent activity way with Kotlin language?

Here, I created a method in general Kotlin class. You can use it in all activities.

```
private val T_GET_SCREEN_WIDTH:String = "screen_width"
private val T_GET_SCREEN_HEIGHT:String = "screen_height"
```

```
private fun getDeviceSizes(activity:Activity, whichSize:String):Int{
```

```
    val displayMetrics = DisplayMetrics()
    activity.windowManager.defaultDisplay.getMetrics(displayMetrics)
```

```
    return when (whichSize){
        T_GET_SCREEN_WIDTH -> displayMetrics.widthPixels
        T_GET_SCREEN_HEIGHT -> displayMetrics.heightPixels
        else -> 0 // Error
    }
```

```
}
```

ShareFollow

edited May 21, 2021 at 17:56

answered Feb 24, 2020 at 14:53



ChrisF ♦

132k

30

252

323



nonstopdevlife

4,836

36

33

¹ Maybe even better solution would be to create an extension function for Activity class. – Micer May 12, 2020 at 13:55

I found **weigan**'s answer best one in this page, here is how you can use that in Xamarin.Android:

0
▼

```
public int GetScreenWidth()
{
    return Resources.System.DisplayMetrics.WidthPixels;
}
```

```
public int GetScreenHeight()
{
    return Resources.System.DisplayMetrics.HeightPixels;
}
```

ShareFollow

answered Feb 5, 2018 at 6:56



Mehdi Dehghani

10k 6 56 58

▲ Screen resolution is total no of pixel in screen. Following program will extract the screen resolution of the device. It will print screen width and height. Those values are in pixel.

0
▼

```
public static Point getScreenResolution(Context context) {
    // get window managers
    WindowManager manager =
        (WindowManager)context.getSystemService(Context.WINDOW_SERVICE);
    Display display = manager.getDefaultDisplay();
    Point point = new Point();
    display.getSize(point);

    // get width and height
    int width = point.x;
    int height = point.y;

    return point;
}
```

ShareFollow

answered Feb 13, 2018 at 20:03



Fakhridin Abdullaev

3,560 2 29 33