



ΑΝΔΡΟΙΔ

Ανδρέας Π. Πλαγεράς
Ph.D. in Applied Informatics

XML – eXtensible Markup Language & Java

Περιεχόμενα Μαθήματος

- ❑ Εργαλεία Προγραμματισμού
 - Eclipse IDE – Android SDK – ADT Plugin – AppInventor
- ❑ Εισαγωγή στο Android Studio και εγκατάστασή του.
- ❑ Δομή Android Project
 - Βασικά Δομικά Στοιχεία
- ❑ Δημιουργία Απλής Εφαρμογής
 - Γραφική Διεπαφή – XML – Layouts

Βασικά Στάδια Ανάπτυξης Εφαρμογών

- Εγκατάσταση Λογισμικού
 - Ανάπτυξη Πηγαίου Κώδικα Εφαρμογής
 - Αποσφαλμάτωση (Debugging)
 - Τελειοποίηση και Διάθεση της Εφαρμογής
- 

Παράμετροι για την Ανάπτυξη μιας Εφαρμογής

- ❑ Λογισμικό Ανάπτυξης – Android SDK
- ❑ Περιβάλλοντα Ανάπτυξης Λογισμικού
 - IDE (Integrated Development Environment)
 - Eclipse IDE
- ❑ Εργαλεία Ανάπτυξης, Δοκιμής και Αποσφαλμάτωσης
 - Android Debug Bridge (ADB)
 - Dalvik Debug Monitor Server (DDMS)
 - Logcat – Εργαλείο Καταγραφής Συμβάντων
 - Κλάση Log
 - Android Asset Packaging Tool (AAPT)
 - Android Interface Description Language (AIDL)
 - Εικονικές Συσκευές Android (Android Virtual Devices, AVD)
 - Βιβλιοθήκη Acra του Android (Application Crash Reporter for Android)

Βήματα για την Ανάπτυξη Android Εφαρμογών



Βήμα 1^ο

- Επιλογή των Views δηλαδή του περιεχομένου της εφαρμογής (π.χ. Text, Image, Button, κλπ).
 - Δηλαδή ότι μπορούμε να δούμε στην οθόνη.
- Όλα αυτά τα Views μαζί συνθέτουν αυτό που λέμε Layout of the screen.
 - Ότι βλέπουμε και με ότι μπορούμε να αλληλεπιδράσουμε αποτελεί το Περιβάλλον Χρήστη (User Interface, UI).
- Κάθε View είναι ένα τετράγωνο στην οθόνη, το οποίο όμως είναι αόρατο.

Παραδείγματα Διαφορετικών Τύπων Views

□ <TextView

```
android:text="Happy Birthday"  
android:background="@android:color/darker_gray"  
android:layout_width="150dp"  
android:layout_height="75dp" />
```

□ <ImageView

```
android:src="@drawable/όνομα_εικόνας"  
android:layout_width="150dp"  
android:layout_height="75dp"  
android:scaleType="center" ή "centerCrop" />
```

Βήμα 2^ο

□ Μορφοποίηση (Style) των Views

✓ `android:background="@android:color/darker_gray"`

ή

✓ `android:background="@android:color/#00AACC"`

✓ `android:textSize="45sp"` →

sp = Scale-independent
Pixels (Μόνο για Fonts)

ή

✓ `android:textAppearance="?android:textAppearanceLarge`

✓ `android:textAppearance="?android:textAppearanceMedium`

✓ `android:textAppearance="?android:textAppearanceSmall` }

✓ `android:scaleType="center"` → centered but too big

✓ `android:scaleType="centerCrop"` → fit height and width

✓ `android:textAllCaps="true"` (or "false")

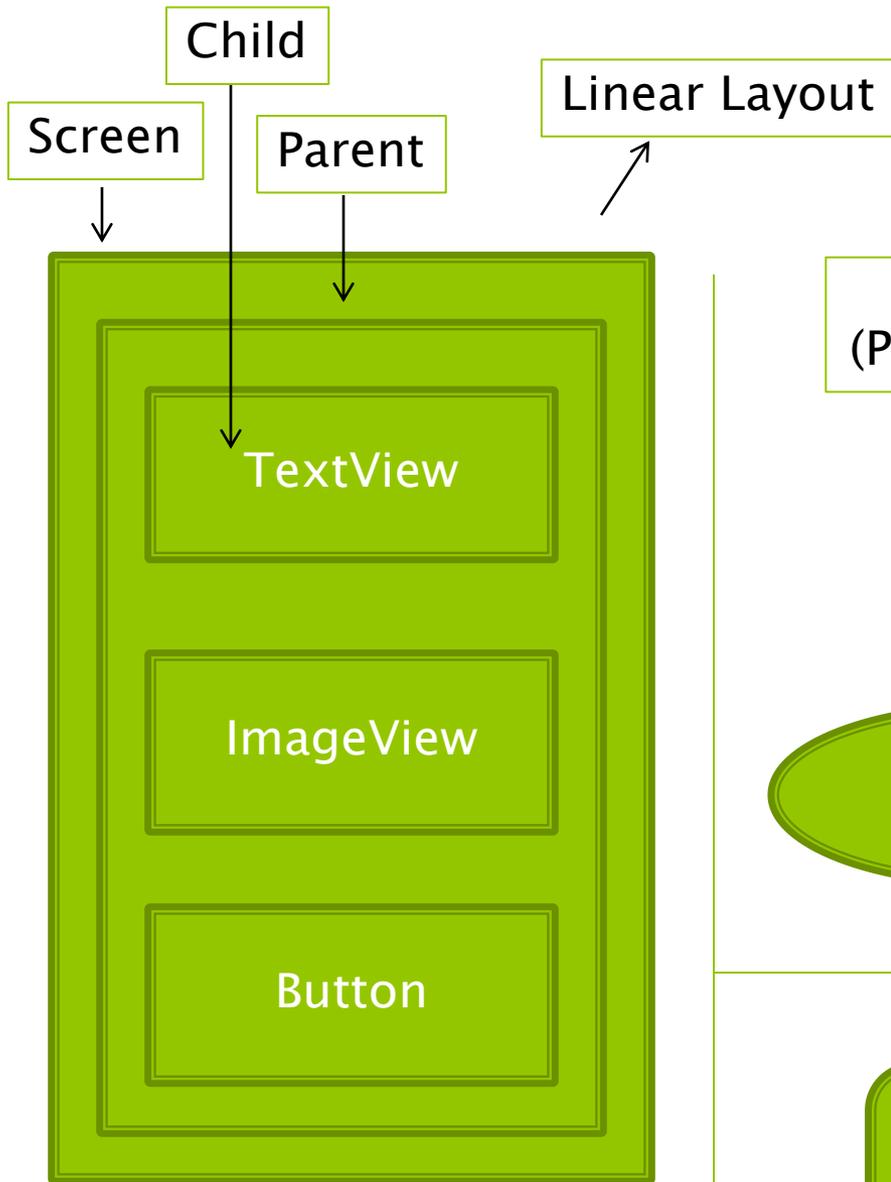
✓ `android:textStyle="bold"` (or "italic" or "normal")

Βήμα 3^ο

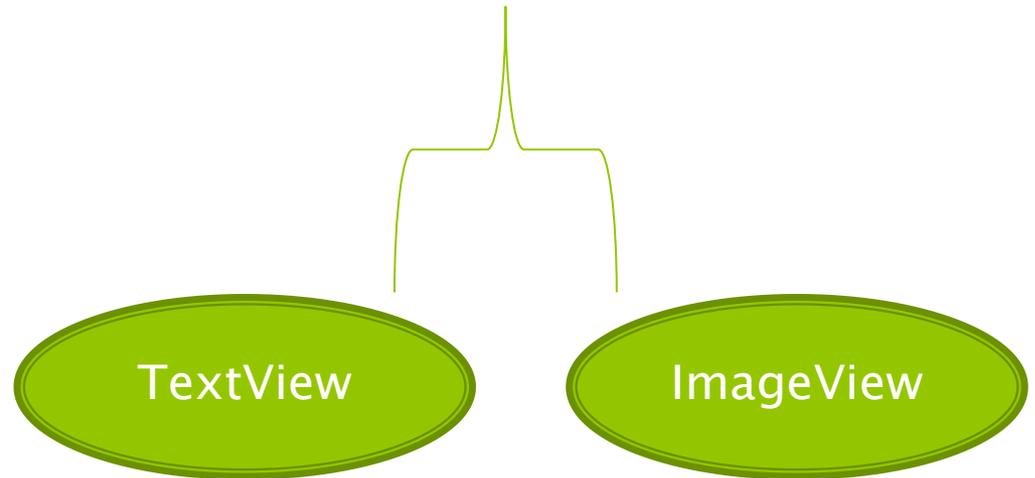
□ Group of Views (Positioning the Views)

- Αν έχω ένα TextView σκέτο και το αντιγράψω, τότε θα βγει ένα error το οποίο θα λέει ότι δεν μπορούμε να έχουμε περισσότερα από ένα root view. Άρα αυτό που πρέπει να κάνουμε είναι να τα βάλουμε μαζί σε ένα View Group, έτσι ώστε το μοναδικό root view να είναι το View Group.
- Όταν έχουμε ένα View Group τα Views που είναι μέσα σε αυτό το group είναι τα παιδιά του View Group. Το View Group είναι το Parent View.
- Όταν έχουμε 2 ή περισσότερα παιδιά (child) σε ένα View Group τότε αυτά είναι siblings μεταξύ τους.
- Το View Group είναι και αυτό ένα View.

Παράδειγμα



Children Views - Siblings
(Positioned based on their parents)

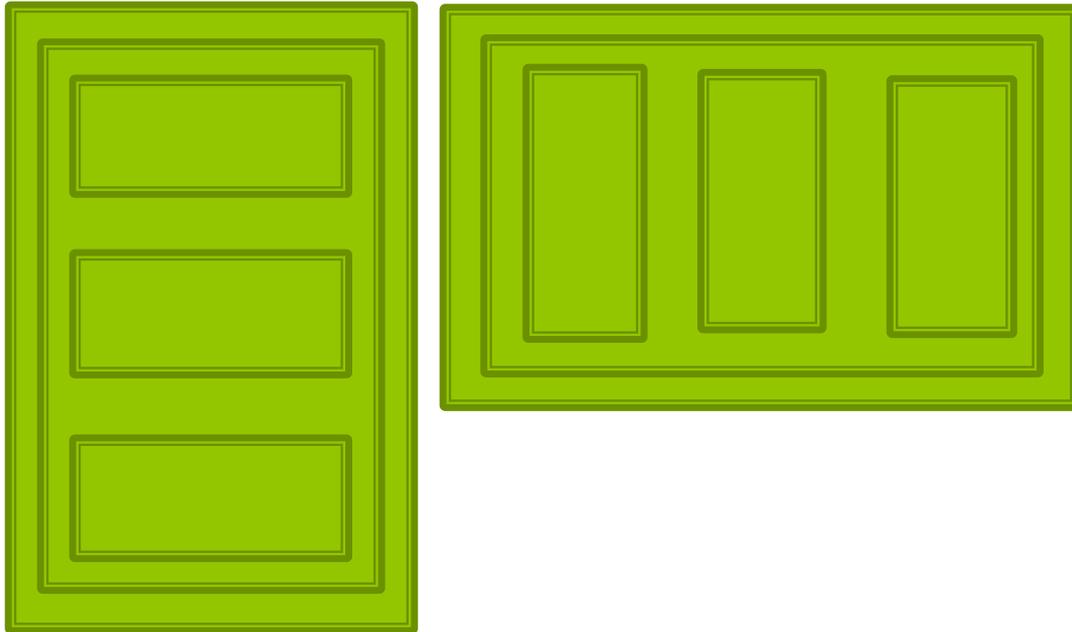


Parents

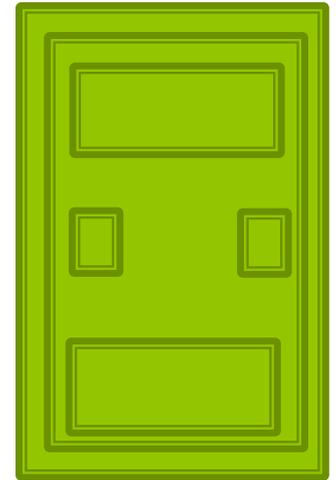


Μέθοδοι που χρησιμοποιούνται για την Οργάνωση Διεπαφών

Linear Layout



Relative to the parent Layout



Relative to other children Layout

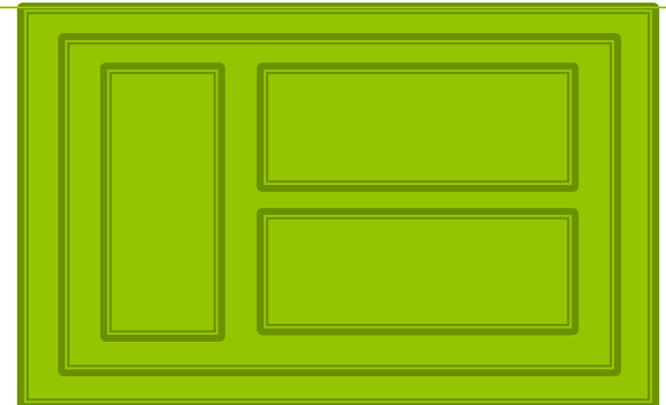


Table Layout
&
List Layout

Linear Layout

□ <LinearLayout

```
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" ή "horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
    <TextView
        android:text="Happy Birthday"
        android:background="@android:color/darker_gray"
        android:layout_width="150dp"
        android:layout_height="75dp" />
    <ImageView
        android:src="@drawable/όνομα_εικόνας"
        android:layout_width="150dp"
        android:layout_height="75dp"
        android:scaleType="center" ή "centerCrop" />
</LinearLayout>
```

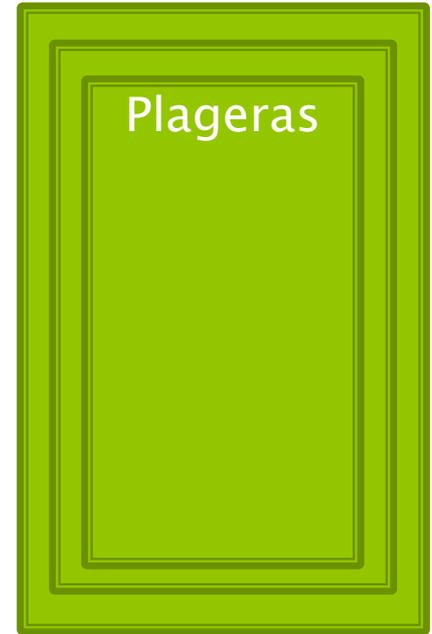
Linear Layout (width and height)



200 dp



wrap_content

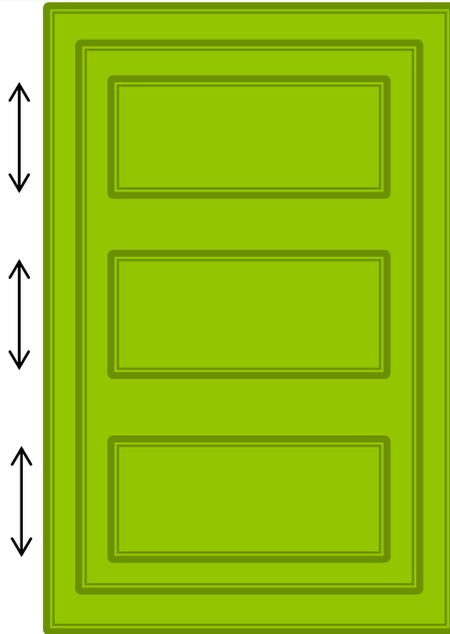


match_parent

Linear Layout – Weight Attribute

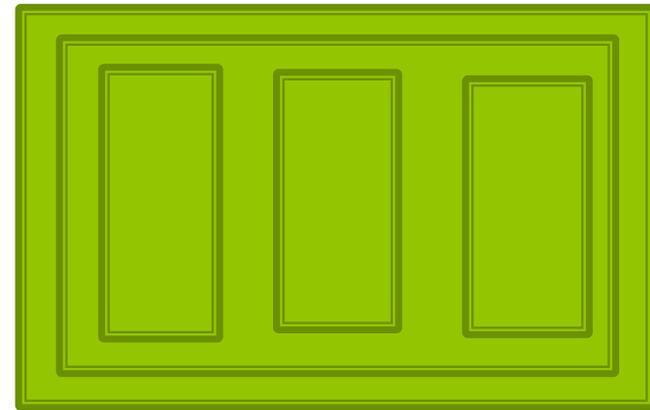
Equal Weights means equal Heights

3 child views
Height=0 dp
Weight=1



Equal Weights means equal Widths

3 child views
Width=0 dp
Weight=1



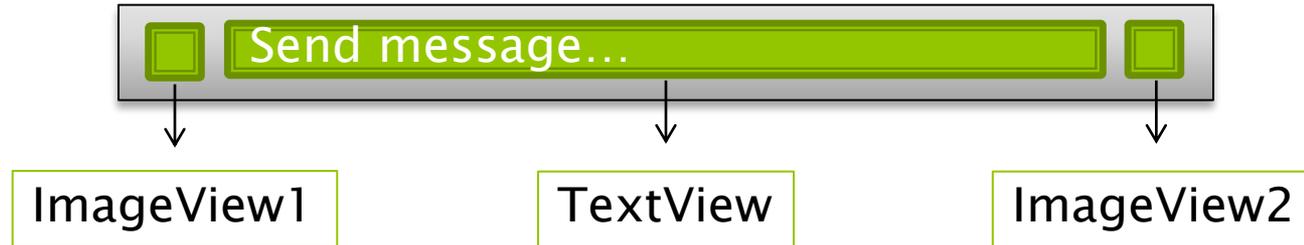
Δίνουμε τιμές 0 και 1
καθώς δεν ξέρουμε το
ακριβές ύψος ή πλάτος

Linear Layout – Responsive Design

□ layout_weight

- (just search in google for **linearlayout equally spaced children**).
- `android:layout_height="0 dp"` (*vertical*)
- `android:layout_width="0 dp"` (*horizontal*)
- `android:layout_weight="1"` (*όσο μεγαλύτερο αριθμό βάζω τόσο μεγαλύτερο χώρο θα δώσει στο view*)
- Μπορώ να δηλώσω `width=""wrap_content"` και `height="200 dp"`. Στην συσκευή μπορεί να φαίνεται εντάξει αλλά αν πάμε σε μεγαλύτερη οθόνη τα Views θα φαίνονται πάλι πάνω αριστερά.
- Αν βάλω `match_parent` τότε τα υπόλοιπα views θα «πεταχτούν έξω από τα όρια της οθόνης».

Παράδειγμα 1



Views	Width	Height	Weight
ImageView1	match_parent	wrap_content	0
TextView	match_parent	0 dp	1
ImageView2	match_parent	wrap_content	0

Παράδειγμα 2

ImageView

TextView1

TextView2

Views	Width	Height	Weight
ImageView	match_parent	0 dp	1
TextView1	match_parent	wrap_content	0
TextView2	match_parent	wrap_content	0

Relative Layout

□ <RelativeLayout>

```
xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
```

```
    <TextView
```

```
        android:text="Happy"
```

```
        android:background="@android:color/darker_gray"
```

```
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
```

```
        android:layout_alignParentRight="true"
```

```
        android:layout_alignParentTop="true"
```

```
        android:textAppearance="?android:textAppearanceLarge"/>
```

```
    <TextView
```

```
        android:text="Birthday"
```

```
        android:background="@android:color/darker_gray"
```

```
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
```

```
        android:layout_alignParentBottom="true"
```

```
        android:layout_alignParentLeft="true"
```

```
        android:textAppearance="?android:textAppearanceLarge"/>
```

```
    <TextView
```

```
        android:text="To You"
```

```
        android:background="@android:color/darker_gray"
```

```
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
```

```
        android:textAppearance="?android:textAppearanceLarge"/>
```

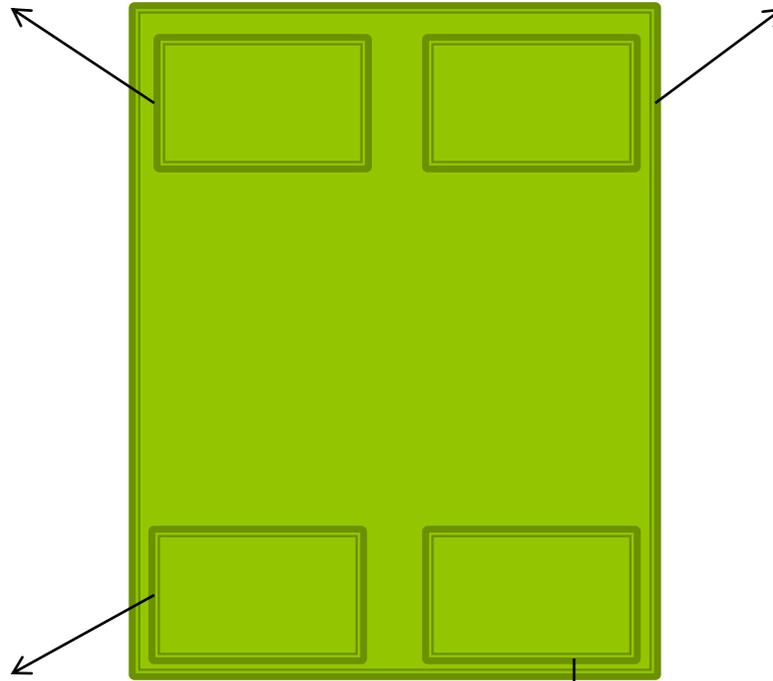
```
</RelativeLayout>
```

Στο τελευταίο View δεν ορίζω alignment γιατί μου κάνουν οι default τιμές (Top-Left)

AlignParent 1

`android:layout_alignParentTop="true"`
`android:layout_alignParentLeft="true"`

`android:layout_alignParentTop="true"`
`android:layout_alignParentRight="true"`



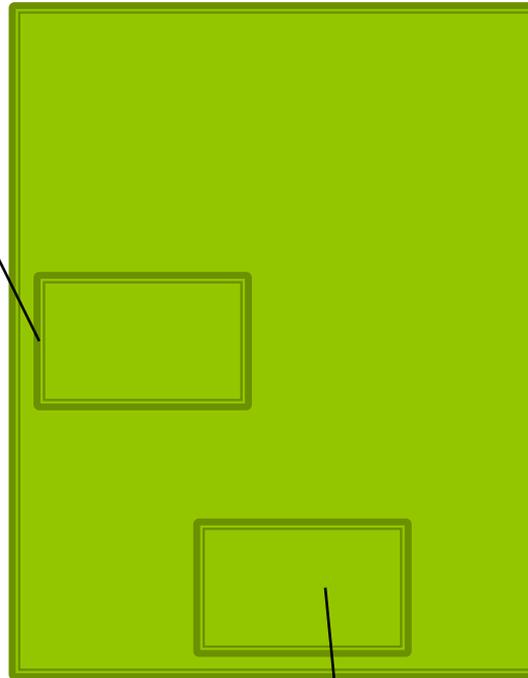
`android:layout_alignParentBottom="true"`
`android:layout_alignParentLeft="true"`

`android:layout_alignParentBottom="true"`
`android:layout_alignParentRight="true"`

AlignParent 2

`android:layout_centerVertical="true"`

Και εδώ δεν προσθέσαμε `alignParentLeft` γιατί μας κάνει η default τιμή

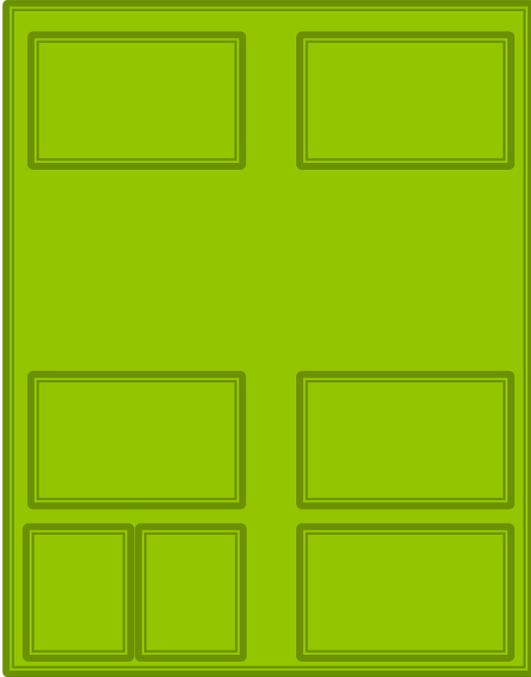


`android:layout_centerHorizontal="true"`
`android:layout_alignParentBottom="true"`

Relative Layout: Positioning Children Relative to other Views / Children



Παράδειγμα 1



Το parent φαίνεται να είναι full width του screen, οπότε έχουμε `match_parent` στο width και στο height.

Ο τρόπος για να προσδιορίσουμε κάποιο view είναι να χρησιμοποιήσουμε View IDs για “positioning children relative to other view”.

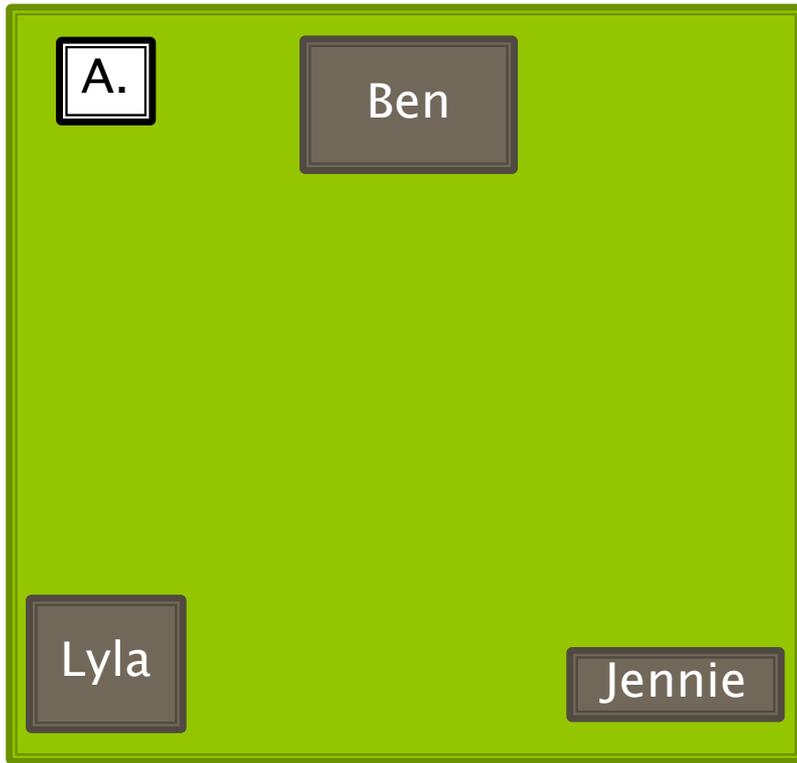
Π.χ. `android:layout_toLeftOf="@id/ben_text_view”`
Το `ben_text_view` είναι το όνομα που δώσαμε στο text view.

`android:id="@+id/ben_text_view”`

Από το παράδειγμα το + σημαίνει ότι ορίζουμε πρώτη φορά αυτό το ID.

Παράδειγμα 1 (Συνέχεια)

Αρχικά τοποθετούμε τον Ben, την Lyla και την Jennie όπως ξέρουμε.



Τελικό

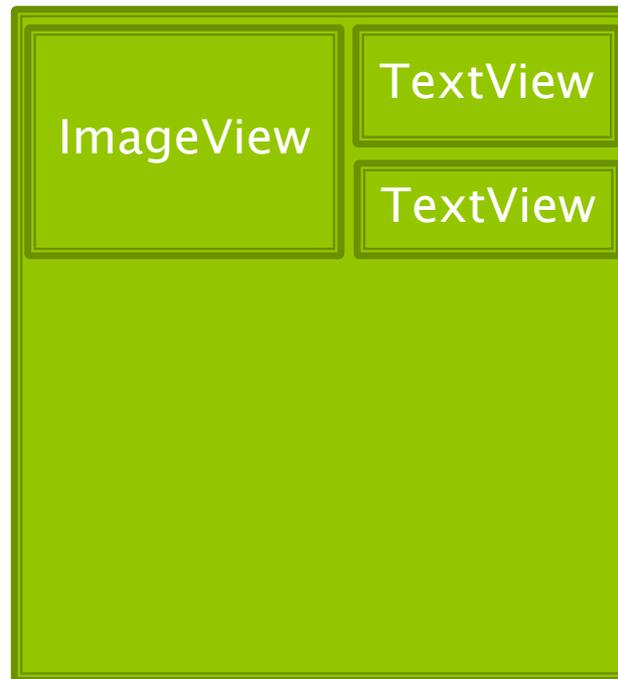


Παράδειγμα 1 (ο κώδικας)

- Ο κώδικας βρίσκεται στο Relative Layout αρχείο .txt.
- Στο παρακάτω link θα βρούμε τα parameters για το positioning των views στο layout. (Google Search:RelativeLayout params)
 - <https://developer.android.com/reference/android/widget/RelativeLayout.LayoutParams.html>

Παράδειγμα 2

Τελικό



Τους κώδικες θα τους βρείτε
στα .txt αρχεία Linear &
Relative Layout Example 2

Use some Whitespace

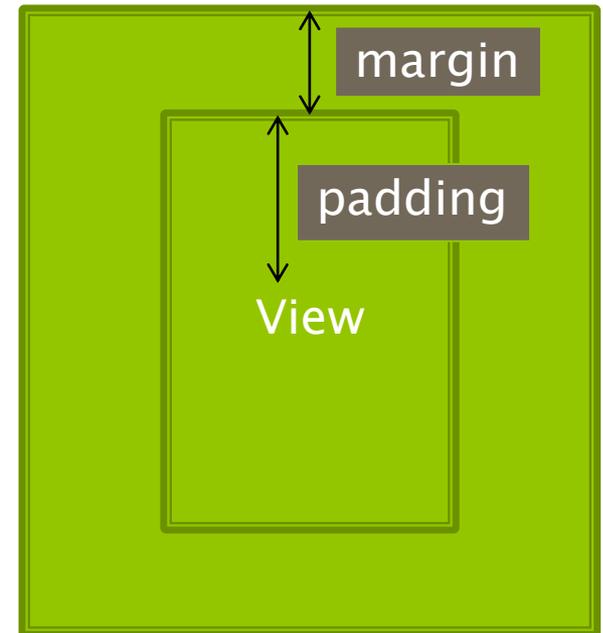
□ Padding

- `android:padding="8dp"`
- ή
- `android:paddingLeft="8dp"`
- `android:paddingRight="8dp"`
- `android:paddingTop="8dp"`
- `android:paddingBottom="8dp"`

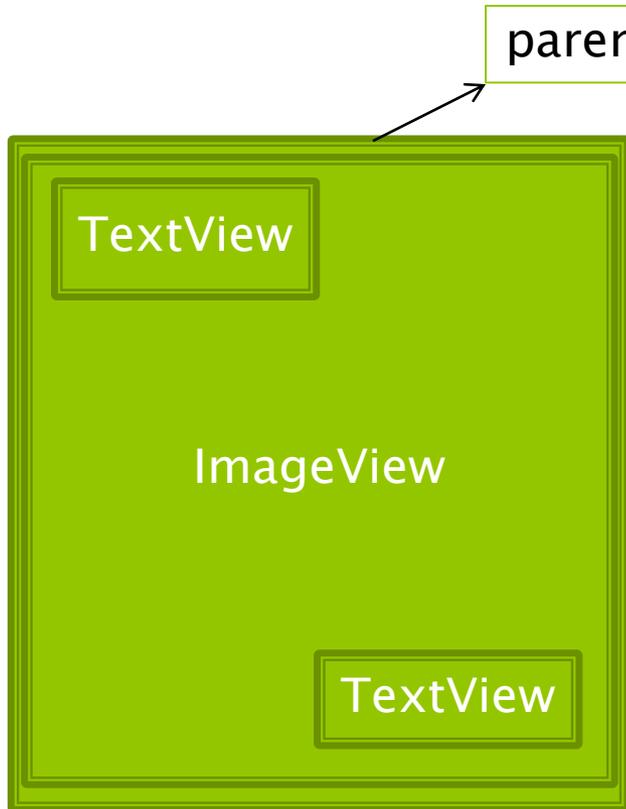
□ Margin

- `android:layout_margin="8dp"`
- ή
- `android:layout_marginLeft="8dp"`
- `android:layout_marginRight="8dp"`
- `android:layout_marginTop="8dp"`
- `android:layout_marginBottom="8dp"`

Η τιμή By Default
είναι μηδέν (0)



Παράδειγμα



Εφόσον τα TextView επικαλύπτουν (overlap) την εικόνα (ImageView), άρα θα χρειαστεί να κάνουμε position τα views με TopLeft, BottomRight, etc., οπότε θα χρησιμοποιήσουμε Relative Layout Positioning των Views.

Ο κώδικας βρίσκεται στο .txt αρχείο ExampleOverlap

Build the Layout

- Στην καρτέλα Design με Drag&Drop μπορούμε να σείρουμε ότι θέλουμε να έχει το Layout. Φυσικά όπως έχουμε πει για κάθε τι που εισάγουμε μπαίνει και ο κώδικας αυτόματα στο Text.
- Στην καρτέλα Text στον κώδικα μπορούμε να επιλέξουμε τον κώδικα και να τον τροποποιήσουμε (Reformat). Από το μενού: Android Studio → Code Reformat → Code και έτσι ο κώδικας έχει παραμετροποιηθεί έτσι ώστε να έχει κενά μεταξύ των Tabs.
- Επίσης, μπορώ να επιλέξω από το μενού: Android Studio → Code → Rearrange Code και ο κώδικας γίνεται πιο εύκολα αναγνώσιμος.

Create a Project

- ❑ App name: Plageras
- ❑ Company Name: android.example.com
- ❑ Package Name: com.example.android.Plageras

- ❑ Πως θα χτίσουμε το layout της εφαρμογής μας;
 - ❑ Βήμα 1^ο: Επιλογή των Views
 - ❑ Βήμα 2^ο: Τοποθέτηση των Views (positioning)
 - ❑ Βήμα 3^ο: Μορφοποίηση των Views (styling)

Βήματα και Αποτέλεσμα



TextView1
TextView2
TextView3
TextView4
TextView5

BUTTON

Andreas P. Plageras

QUANTITY

2

PRICE

\$10

ORDER

Andreas P. Plageras

QUANTITY

+

2

-

PRICE

\$10

ORDER

Andreas P. Plageras

QUANTITY

- 2 +

PRICE

\$10

ORDER

Στα Buttons το κείμενο γίνεται με κεφαλαία αυτόματα

Plan! – Πως θα χτίσουμε το Layout της εφαρμογής μας;

- ❑ Βήμα 1^ο : Ποια Views θα διαλέξω;
- ❑ Βήμα 2^ο : Ποια ViewGroup θα είναι Root View; (Positioning)
- ❑ Βήμα 3^ο : Style the Views

- ❑ Απαντήσεις:
 - Θα χρησιμοποιήσω 2 TextView και 1 Button.
 - Χρησιμοποιούμε Linear Layout (Vertical) ως Parent View Group για τα 3 children Views.
 - Το QUANTITY πρέπει να είναι AllCaps.
 - Θα χρειαστεί να βάλουμε spacing μεταξύ των Views (padding, margin).
 - QUANTITY value should be in black font.

activity_main.xml

```
1. <LinearLayout
2. xmlns:android="http://schemas.android.com/apk/res/android"
3. xmlns:tools="http://schemas.android.com/tools"
4.     android:layout_width="match_parent"
5.     android:layout_height="match_parent"
6.     android:paddingBottom="@dimen/activity_vertical"
7.     android:paddingLeft="@dimen/activity_horizontal"
8.     android:paddingTop="@dimen/activity_horizontal"
9.     android:paddingRight="@dimen/activity_vertical"
10.    android:orientation="vertical"
11.    tools:context=".MainActivity">
12.    <TextView
13.        android:text="Quantity"
14.        android:layout_width="wrap_content"
15.        android:layout_height="wrap_content"
16.        android:textAllCaps="true" />
17.    <TextView
18.        android:text="0"
19.        android:layout_width="wrap_content"
20.        android:layout_height="wrap_content"
21.        android:textSize="16sp
22.        android:textColor="@android:color/black" />
23.    <Button
24.        android:text="Order"
25.        android:layout_width="wrap_content"
26.        android:layout_height="wrap_content" />
27. </LinearLayout>
```



Ο κώδικας είναι
στο .txt αρχείο
PlagerasExample

Do Sth when the Button is Clicked!

- Όταν επιλέξαμε Empty Activity το πρόγραμμα έφτιαξε αυτόματα ένα Activity για μας και εμείς απλά επιλέξαμε το Default Name: MainActivity.
- Ένα Activity αναπαριστά ένα από πράγμα εστίασης που θα έχει κάποιος. Μερικές φορές το Activity είναι ένα παράθυρο της εφαρμογής. Άλλες φορές ένα Activity μπορεί να έχει πολλαπλά (multiple) παράθυρα.
- Στην ουσία το Activity είναι πως θα προβάλω το layout και πως θα αλληλεπιδράσω με το χρήστη.
- Επίσης, ένα XML αρχείο δημιουργήται αυτόματα και είναι το activity_main.xml

Οπότε το MainActivity προβάλει το
layout activity_main

Do Sth when the Button is Clicked!

- Το κάθε Activity έχει έναν συγκεκριμένο σκοπό, μια συγκεκριμένη λειτουργία (είναι στοχευμένο δηλαδή το τι θα κάνει).
- Με την δημιουργία ενός νέου project δημιουργείται αυτόματα και το αρχείο «MainActivity.java». Μέσα σε αυτό το αρχείο έχει δημιουργηθεί αυτόματα και ένα κομμάτι του κώδικα Java που προσδίδει στην εφαρμογή μας την απαραίτητη λειτουργικότητα.

Ο αρχικός κώδικας για την MainActivity βρίσκεται στο αρχείο JavaFile.txt

activity_main.xml (ProjectJava)

```
> <?xml version="1.0" encoding="utf-8"?>
  <LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"

    tools:context="com.example.andreas.projectjavaprice.MainActi
    vity">
    <TextView
      android:text="Quantity"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:textAllCaps="true" />
    <TextView
      android:id="@+id/quantity_text_view"
      android:text="0"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:textSize="16sp"
      android:textColor="@android:color/black" />
    <Button
      android:text="Order"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:onClick="submitOrder"/>
  </LinearLayout>
```



Ο κώδικας είναι
στο .txt αρχείο
ProjectJavaPrice_xml

Αρχική MainActivity.java

```
1. package com.example.andreas.plageras;  
2. import android.support.v7.app.AppCompatActivity;  
3. import android.os.Bundle;  
4. public class MainActivity extends AppCompatActivity {  
5.     @Override  
6.     protected void onCreate(Bundle savedInstanceState) {  
7.         super.onCreate(savedInstanceState);  
8.         setContentView(R.layout.activity_main);  
9.     }  
10. }
```

Ο αρχικός κώδικας για
την MainActivity
βρίσκεται στο αρχείο
JavaFile.txt

MainActivity.java (ProjectJava)

```
1. package com.example.andreas.projectjava;
2.
3. import android.support.v7.app.AppCompatActivity;
4. import android.os.Bundle;
5. import android.view.View;
6. import android.widget.TextView;
7.
8. public class MainActivity extends AppCompatActivity {
9.
10.     @Override
11.     protected void onCreate(Bundle savedInstanceState) {
12.         super.onCreate (savedInstanceState);
13.         setContentView (R.layout.activity_main);
14.     }
15.     public void submitOrder(View view) {
16.         display (1);
17.     }
18.
19.     private void display(int number) {
20.         TextView quantityTextView=(TextView) findViewById(R.id.quantity_text_view);
21.         quantityTextView.setText(""+number);
22.     }
23. }
```

Ο κώδικας για την
MainActivity
βρίσκεται στο αρχείο
ProjectJava.txt

activity_main.xml (ProjectJavaPrice)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.example.andreas.projectjavaprice.MainActivity">

    <TextView
        android:text="Quantity"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAllCaps="true" />

    <TextView
        android:id="@+id/quantity_text_view"
        android:text="0"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="16sp"
        android:textColor="@android:color/black" />

    <TextView
        android:id="@+id/price_text_view"
        android:text="Price"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="16sp"
        android:textAllCaps="true" />

    <TextView
        android:text="$0"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="16sp"
        android:textColor="@android:color/black" />

    <Button
        android:text="Order"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="submitOrder"/>

</LinearLayout>
```

Ο κώδικας είναι
στο .txt αρχείο
ProjectJavaPrice_xml

MainActivity.java (+PRICE)

□ **package** com.example.andreas.projectjavaprice;

```
import android.annotation.SuppressLint;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.TextView;
```

```
import com.example.andreas.projectjavaprice.R;
```

```
import java.text.NumberFormat;
```

```
public class MainActivity extends AppCompatActivity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }
```

```
    public void submitOrder(View view) {  
        display(1);  
    }
```

```
    private void display(int number) {  
        TextView quantityTextView = (TextView) findViewById(R.id.quantity_text_view);  
        quantityTextView.setText("" + number);  
    }
```

```
    private void displayPrice(int number) {  
        TextView priceTextView = (TextView) findViewById(R.id.price_text_view);  
        priceTextView.setText(NumberFormat.getCurrencyInstance ().format(number));  
    }
```

```
}
```

Ο κώδικας για την
MainActivity
βρίσκεται στο αρχείο
ProjectJavaPrice.txt

Literals vs Variables

▶ Quantity is:

- 2

▶ Price is:

- $2 * 5$

▶ Paper cup charge is:

- $2 * 2$

▶ Quantity is:

- `numberOfCoffees`

▶ Price is:

- `numberOfCoffees * 5`

▶ Paper cup charge is:

- `numberOfCoffees * 2`

▶ *Set `numberOfCoffees` to be 2*

Literals

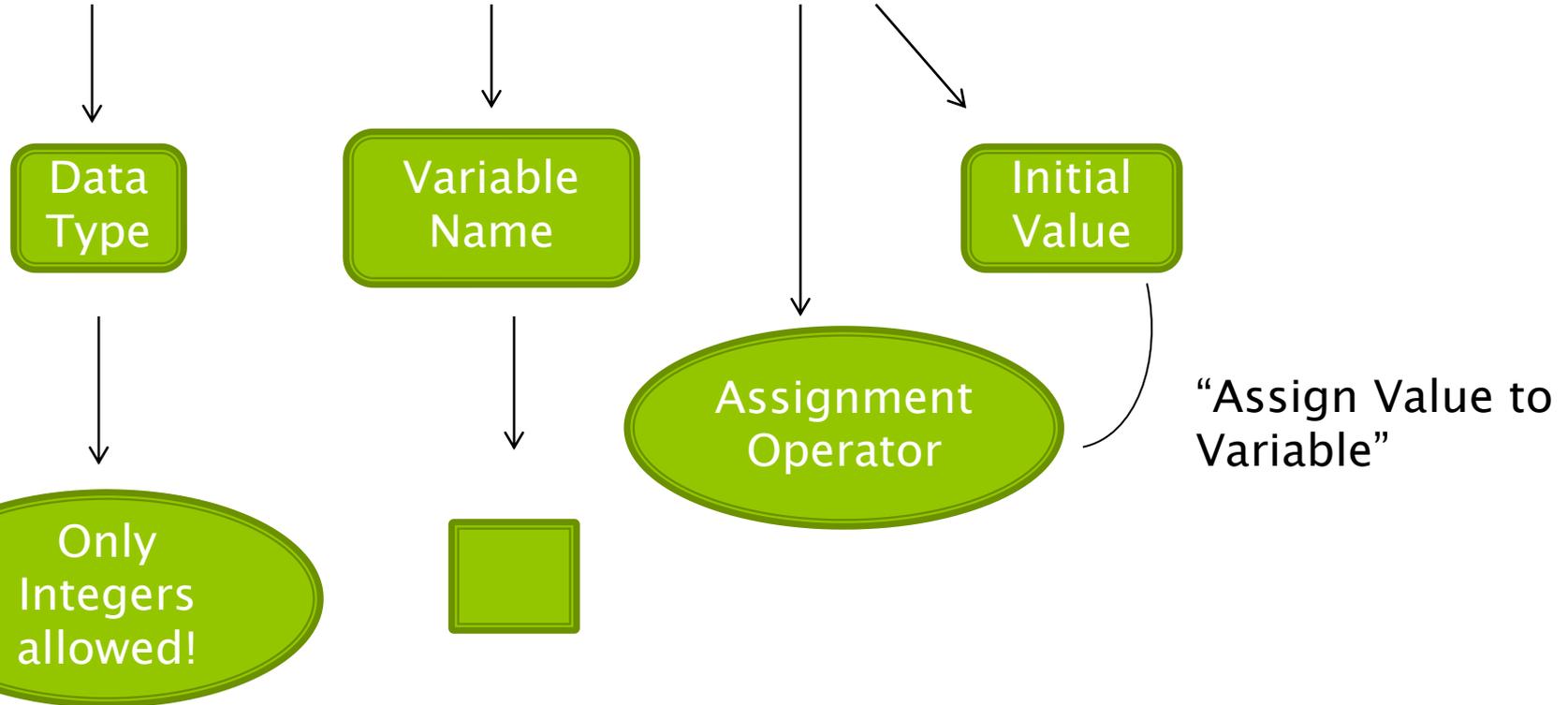
Variables

Using a Variable

- ❑ `int numberOfCoffees = 2;`
- ❑ `display(numberOfCoffees);`
- ❑ `displayPrice(numberOfCoffees*5);`

Declare (Define/Create) a Variable

```
int numberOfCoffees = 2;
```



Java Rules for Variables

- ❑ Google Search: variable names java
 - <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/variables.html>
 - Look for Naming
- ❑ `int numberOfCoffees = 2;`
- ❑ `int quantity = 2;`
- ❑ `display(numberOfCoffees);`
- ❑ `quantity);`
- ❑ `displayPrice(numberOfCoffees*5);`
- ❑ `displayPrice(quantity*5);`

Thank you for your attention!

