Mit Android Studio Seiten mit Text, Bildern Schaltknöpfen generieren, um zwischen den Seiten und zum Internet navigieren zu können.

## Schritt 1:

Gehe zu <u>http://developer.android.com/sdk/index.html</u>, lade "Android Studio" herunter, installiere das Programm ...



Android Studio provides the fastest tools for building apps on every type of Android device.

021.1.1 Patch 1 for W

s 64-bit (872 Mi8)

... und starte ein neues Projekt.

	*	Eile	Edit	⊻iew	<u>N</u> aviga	te <u>C</u> ode	Analy <u>z</u> e	Befa	tor B	uild R Project	yn Iool	s VCS	Window		
	-	-	Open						Impo	t Proje	ct				
New Project															>
Templates									÷		Ŧ		÷	÷	
Phone and Table	et						1								
Wear OS															
Android TV Automotive															
						No Activit	y		Basi	c Activity		В	ottom Navigat	ion Activity	
					÷	Ô			÷				¢	0	
										D					
					Empt	y Compose	Activity		Emp	ty Activit	/		Fullscreen A	activity	
						interstitial A	3		¢	•	•		* 	•	
										Previo	bus	Next	Cancel	Fini	ish

Übernimm die voreingestellte "Empty Activity" mit einem Klick auf "Next".

Empty Activity		
Creates a new e	mpty activity	
Name	test	
Package name	mintunttest	
Save location	C:\Users\gstei\AndroidStudioProjects\test	50
Language	Java	¥
Minimum SDK	API 21: Android 5.0 (Lollipop)	×
	Your app will run on approximately 98,0% of devices. Help me choose     Use legacy android support libraries     Using legacy android support libraries     Using legacy android support libraries	
<ol> <li>The applic</li> </ol>	ation name for most apps begins with an uppercase letter	

Gib dem Projekt einen Namen, beispielsweise "test". Der Package-Name ist beliebig. Der Speicherort kann übernommen werden. Als Sprache wird hier "Java" (statt Kotlin) der Vorzug gegeben. Als "Minimum SDK" ist eine frühe Android-Version (Lollipop) eingestellt worden, um auf möglichst allen Android-Smartphones lauffähig zu sein. Nach einem Klick auf "Finish" zeigt sich nach kurzer Zeit der Startbildschirm:



### Schritt 2: Die Willkommensanzeige ändern

Wähle in der vierten Zeile im Startfenster statt "MainActivity.java" ...



den Reiter "activity\_main.xml" aus.

st	app src m	ain 👌 res 🤇 layout 👌 🏭 activity_n	nain.xml				
	activity_main.xr	ml 🔀 😉 MainActivity.java 🛛					
					≣ Code	💵 Split 🔜 Des	sig
100	Palette	Q ¢ -	- 🧶 🔍 🔍 🛛	Pixel 🗸 🛋 31 🗸 🔘 Test 🤟	⑤ Default (en-us) ∨	0	107
in mate	Common	Ab TextView	⊙ U _0dp _ J	: × I.		0	MUL
-	Text	Button					COAN
-	Buttons	ImageView     Recusted/iew					
	Widgets	FragmentContainerView		Ŷ	Y		
-	Layouts	ScrollView			\$		
0	Containers	- Switch			l ŝ		
1	Helpers						
	Google				\$		
	Legacy				Š		
					3		
		La .			1 N N	+	
					Ę.		
				11		1:1	

Klicke im Design-Fenster auf das Kästchen "Hello World". Wähle dann in der 5. Zeile des Startfensters rechts statt "Design" den Reiter "Split" aus.



Es öffnet sich ein Fenster mit Programmcode. Der gesuchte Abschnitt ist mit "<TextView" überschrieben. In der dritten Zeile darunter findet sich "Hello World!". Ergänze dort beispielsweise "Welcome to my App!"

Klicke in die veränderte Zeile und führe anschließend den Cursor zu dem auf der linken Seite erscheinenden Symbol (eine Lampe). Wenn sich kurz darauf der Hinweis "Click or press Alt + Eingabe" zeigt, folge dem, um folgendes Fenster zu öffnen:



Mit der Eingabe-Taste wird eine weitere Anzeige aktiviert:



Ein Klick auf "OK" erstellt einen "String" und wandelt die Textzeile "Hello World! Welcome to my app" in "@string/hello\_world\_welcome\_to\_my\_app" um. Die so erzeugten Daten werden unter >res >values in der Datei "strings.xml" gespeichert.

Das Zurückschalten in den Design-Modus zeigt nun in der Mitte des virtuellen Screens "Hello World! Welcome to my App!

# Schritt 3: Einen Umschalt-Knopf in der "MainActivity" einrichten

Im Design-Modus der "activity\_main.xml" findet sich links neben der Programmieroberfläche eine "Palette" von vorgefertigten Objekten.



Ziehe aus dieser Palette einen "Button" in die linke obere Ecke des virtuellen App-Screens.

Palette	Q ‡ —					
Common	Ab TextView					
Text	Button ImageView RecyclerView					
Buttons						
Widgets	FragmentContainerView					
Layouts	ScrollView					
Containers	<ul> <li>Switch</li> </ul>					
Helpers						
Google						
Legacy						

Klicke - wie oben bereits ausgeführt - auf den Reiter "Split" und suche unter "<Button" nach der Textzeile "Button". Benenne den Begriff um in "Next" und erstelle einen weiteren String.

NEXT
Hello World! Wekcome to my App!

Wenn alles richtiggemacht wurde, steht nun im Design-Modus "NEXT" in dem violetten Feld.

## Schritt 4: Eine zweite Seite einrichten

Klicke mit der **rechten Maustaste** auf "app" in der dritten Zeile des Startfenstersund navigiere über >New > Activity zu > EmptyActivity.

-		
-	New Android Activity	~
		~
	Empty Activity	
	Creates a new empty activity	
	Activity name	
	SecondActivity	
	Generate Layout File	
	Layout name	
	activity_second	
	Generate Layout File	
	C:\Users\gstei\AndroidStudioProjects\test	
	Package name	
	test	
	Communication of the Communica	
	Source Language	
	Java	
	Target Source Set	
	main	
	Previous Next Cancel	Finish

Gib in dem sich öffnenden Fenster der neuen Seite (z.B.) den Namen "SecondActivity" und klicke auf "Finish".

Über der Programmoberfläche werden zwei weitere Reiter angezeigt: "SecondActivity.java" und "activity\_second.xml". Sie erfüllen die gleichen Funktionen wie "activity\_main.xml" und "MainActivity.java".

Auf der zweiten Seite soll nun ein Text platziert werden. Dafür wird in der linken Spalte der Palette "Text" angewählt und dann aus der rechten Spalte das Objekt "AbTextView" auf den virtuellen App-Screen gezogen. Dort bildet sich dann ein Kästchen mit dem Eintrag "TextView" ab.

Ein Klick auf "Split" öffnet das Programmierfenster, in dem im Abschnitt "<TextView" der Eintrag "TextView" durch beispielsweise "Welcome to the second page!" ersetzt werden kann.

Wie oben erläutert kann daraus ein weiterer String gemacht werden.

Auf der zweiten Seite steht nun der gewünschte Text.

### Schritt 5: Aktivieren des Schaltknopfs auf der ersten Seite

Um den Schaltknopf auf der ersten Seite zu aktivieren, muss der Programmcode unter dem Reiter "MainActivity.java" nach den folgenden Vorgaben erweitert werden:

```
package test;
import androidx.appcompat.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.os.Bundle;
```

```
import android.content.Intent;
import android.net.Uri;
import android.widget.Button;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button button = (Button) findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                int ce = view.getId();
                if(ce == R.id.button){
                    Intent intent = new Intent(MainActivity.this, SecondActivity.class);
                    startActivity(intent);
                }
     }
}

}
```

Wenn auf Seite zwei weitere Schaltknöpfe eingerichtet werden sollen, mit denen beispielsweise zur ersten Seite zurückgesprungen (erstes @Override) und zusätzlich über einen "ImageButton" (zweites @Override) eine Internetadresse aufgerufen werden kann, dann lautet der passende Code wie folgt:

```
package test;
import androidx.appcompat.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.os.Bundle;
import android.content.Intent;
import android.net.Uri;
import android.widget.ImageButton;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button button = (Button) findViewById(R.id.button2);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                int ce = view.getId();
                if(ce == R.id.button2){
                   Intent intent = new Intent(SecondActivity.this, MainActivity.class);
                    startActivity(intent);
                }
            }
        });
        ImageButton imageButton = (ImageButton) findViewById(R.id.imageButton);
        imageButton.setOnClickListener() {
            @Override
           public void onClick(View view) {
                int cc = view.getId();
                if (cc == R.id.imageButton) {
                   Intent intent2 = new Intent(Intent.ACTION_VIEW,
                            Uri.parse("http://www.mint-unt.de"));
                   startActivity(intent2);
               }
      }
});
    }
}
```

Ein "ImageButton" ist in der Design-Palette unter "Buttons" zu finden. Ist er auf dem App-Screen platziert, kann auf der rechten Seite "Attributes" angeklickt werden.

Attributes		Q	\$	1
ImageButton		ima	geB	utt
id	imageButton			
V Declared Attrib	utes		+ •	-
layout_width	wrap_content		٠	
layout_height	wrap_content		¥	
layout_constraint	0.5			
layout_constraintE	parent		٠	
layout_constraintS	parent		٣	
layout_constraintT	parent		٠	
layout_marginTop	92dp			
background	oid:color/tra	nspar	ent	L
id	imageButton			
srcCompat	- @drawable/n	mint_	unt	ſ

In diesem Fenster kann durch einen Klick auf das kleine Kästchen rechts neben "srcCompat" ein weiteres Fenster geöffnet werden.

+ G Module neu-j.app = Q-		
Drawable Color Mip Map	Preview	
neu-j.app (4)	- 2000000	
01n Dravable / 1 version	6	mint-unt.de
ic_launcher_background Dravable / 1 version	e e	2
ic_launcher_foreground Drawable   1 version	Names	mint yet
Survey mint unt	Reference:	@drawable/mint_unt
and the second s	Type	GIF
android (174)	Value:	mint_unt.gif
alert_dark_frame Dravable   1 version		
alert_light_frame Drawable   1 version	D.	
III III   -> ->		
alerUptrifere Drastic i 1 verson	D.	OK C

Mit einem weiteren Klick auf das ",+" unter "Pick a Resource", kann ein Bild vom eigenen Rechner heruntergeladen werden, das dann mit obigem Code zum Schaltknopf gemacht werden kann.

# Schritt 6: Objekte auf dem App-Screen verankern.

Klickt man im Design-Modus einen Text oder Schaltknopf im an, werden an den Ecken blaue Quadrate (zum Vergrößern) und auf den Kantenmitten Kreise sichtbar. Diese sind zum Platzieren relativ zu anderen Objekten oder den Bildschirmrändern gedacht. Zieht man einen solchen Kreis zum Bildschirmrand, folgt das Objekt nach. Zieht man danach den auf dem Objekt gegenüberliegenden Kreis zum gegenüberliegenden Bildschirmrand, wird das Objekt auf dieser Achse zentriert usw. Welche Fixierungen eingerichtet worden sind, wird auf dem blauen Screen mit Pfeilen und Zick-zack-Linien angezeigt. Näheres findet sich hier.

