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

Stufe 1:

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



Gib dem Projekt (z.B.) den Namen "MyApplication" und klicke auf "Next". Der Domain-Name ist beliebig.

👳 Create New Proje	xct	×
New Android	/ Project ^{studio}	
Configure your	r new project	
Application name:	My Application	
<u>C</u> ompany domain:	mint-untde	
Package name:	de.mint_unt.myapplication	<u>Edit</u>
Project location:	C:\Users\gstei\AndroidStudioProjects\MyApplication	
	Previous Next Cancel Finish	

Achte bei der Wahl einer Plattform auf den Haken vor "App und Tablet". Klicke "Next".

🕏 Create New Project			×
Target Android De	evices		
Select the form factors your app wil	I run on		
Different platforms may require separate SDKs			
✓ Phone and Tab	let		
Minimum SDK	API 15: Android 4.0.3 (IceCreamSandwich)		•
	Lower API levels target more devices, but have fewer features available.		_
	By targeting API 15 and later, your app will run on approximately 97,4% of the devices		
	that are active on the Google Play Store. Help me choose		
🗌 Wear			
Minimum SDK	API 21: Android 5.0 (Lollipop)		•
□ TV			_
Minimum SDK	API 21: Android 5.0 (Lollipop)		•
Android Auto			
		Previous Next	Cancel Finish

Wähle "Empty Activity", klicke "Next".

🕏 Create New Project				×
Add an Activity	/ to Mobile			
Add No Activity	€ : Basic Activity	Bottom Navigation Activity	← Empty Activity	Fullscreen Activity
Google AdMob Ads Activity	Google Maps Activity	Login Activity	Master/Detail How	Next Cancel Finish

Lass die Einstellungen im nächsten Fenster unverändert und klicke "Finish".

👳 Create New Project		×
Customize the	ne Activity	
←	Creates a new empty activity	
	Activity Name: MainActivity	
	Generate Layout File Layout Name: activity_main	
	Backwards Compatibility (AppCompat)	
Empty Activity		
	The name of the activity class to create	
	Previous	Cancel Finish

Startbildschirm:



Stufe 2: Die Willkommens-Anzeige verändern

Wähle statt "MainActivity.java" den Reiter "activity_main.xml" über der Programmieroberfläche an.



Wähle (unten links) statt "Design" den Reiter "Text" aus und suche unter "<TextView" nach "Hello World!". Ergänze dort "Welcome to my app!"



Klick 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 ihm, um folgendes Fenster zu öffnen:

i n
nt"

Mit der Eingabe-Taste wird eine weitere Anzeige aktiviert:



"OK" erstellt einen "String" und wandelt die Textzeile "Hello World! Welcome to my app" in "@string/hello_world_welcome_to_my_app" um. Die Daten werden (unter >res >values) in der Datei "strings.xml" gespeichert.

Im Design-Modus lautet der Text in der Mitte des virtuellen Screens nun "Hello World! Welcome to my app!

Hinweis: Das Widget "TextView" ist zum Erstellen jeglicher Form von informellen Texten das richtige Werkzeug.

Stufe 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 den "OK Button" in die linke obere Ecke des virtuellen App-Screens.



Klicke (unten links) auf den Reiter "Text" und suche unter "<Button" nach der Textzeile "Button". Benenne den Begriff um in "Next" und erstelle wie gehabt einen weiteren String.

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



Stufe 4: Eine zweite Seite einrichten

Klicke mit der rechten Maustaste auf "app" in der Zeile direkt unterhalb des Hauptmenüs und navigiere über >New > Activity zu > EmptyActivity.



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

👳 New Android Activity		×
Configure Act	tivity	
	Creates a new empty a	activity
	Activity Name:	SecondActivity
←		✓ Generate Layout File
	Layout Name:	activity_second
		Launcher Activity
		Backwards Compatibility (AppCompat)
	Package name:	de.mint_unt.myapplication
	Target Source Set:	main
	The name of the activi	ity class to create
		Previous Next Cancel Finish

Über der Programmeroberfläche werden zwei neue Dateien angezeigt: "SecondActivity.java" und "activity_second.xml". Wähle letztere aus und schalte in die Design-Ansicht.

MyApplication - [C/Users\gstei\AndroidStudioProjects\MyApplicatio	n] - {app]\app\uc\(main/yes\Jayouf\activity_second.x	ml - Android	Studio 2.3.1			- 1	1	×
File Edit View Navigate Code Analyze Refactor Build Run I	ools VCS Window	Help							
	9-▶ 4 幕 ⊪		?					Q	(FI
MyApplication) 📑 app) 🗂 src) 🗂 main) 🕞 res) 🗔 layout)	activity_second.xm	1)							
ng 🖗 Android → 😳 🕂 🕸 - !**	ectivity_main.xml	× eactivity_second.xml ×	SecondA	tivityjava × G MainActivityjava ×					0
Y Discrete Y Discrete Ø MainActivity Ø SecondActivity Ø SecondActivity Ø SecondActivity Ø SecondActivity Ø SecondActivity Ø G Ø G Ø G Ø G Ø G Ø G Ø G Ø G Ø G Ø G Ø G Ø G	Palette Average Widgets Text Layouts Containers Images Date Transitions Advanced Google Design AppCompat	As TextView Sutton Sutton CrockBar CrockBar SeeKBar South SeeKBar South SeeKBar South SeeKBar South SeeKBar South South SeeKBar South South	8 8 8 8 8	Original Control Contro Control Control Control Control Control Control Control Control C		Proper.	Q 27	¢!	Grade 🏘 Android
4	Design Text		G.						lodel
📕 g: Messages 📓 Terminal 🐥 😒 Android Monitor 😤 TODO	erenges text					lvent Log	Gradle	Console	
Gradle build finished in 5s 367ms (6 minutes ago)					n/a n/a	Content -	no contexto	'n	0

Ziehe aus der Palette das Objekt "Ab TextView" in die Mitte des virtuellen App-Screens.

MyApplication - [C\Users\pstei\AndroidStudioProjects\MyApplication File Edit View Navigate Code Analyze Refactor Build Run Tr	h] - [app]\app\sr	C\main\res\layout\activity_second.x	ni - Android	Studio 2.3.1			- o x
	p- ▶ 4 #		?				Q, F
MyApplication	activity_second.	ami					
y 🗣 Android → 🔿 🕂 🗘 - !*	activity_main.xn	nl × 🧧 activity_second.xml ×	SecondAc	tivityjava × 🙆 MainActivityjava ×			3
app	· Palette	Q. Ø- !		📲 🛇 = 🗌 Nexus 4 = 🗯 25 = 🛈 AppTheme 🌐 L	anguage - 🔁 -	Proper	Q = + + -1 8
P □ manifests	All	TextView	@ 1J	:× + * * ::- = · I ·	⊖ 39% ⊕ 🖸 🍓 🚺	ID	textView
T demint untimvapplication	Widgets	Button		a 100 100 100	400 500 400		
© To MainActivity	Lavouts	CheckBox					
E C in SecondActivity	Containers	RadioButton	-9	₹8,700			8
V le de.mint_unt.myapplication (androidTest)	Images	CheckedTextView		The Application			33 46
demint_unt.myapplication (test)	Transitions	C ProgressBar		my oppression			8
Gradle Scripts	Advanced	 ProgressBar (Horizontal) 	8				
10	Google	 SeekBar 					
	AppCompat	SeekBar (Discrete)				Incruit wi	contant 2
		* RatingBar	8			hayoucina	
		Switch				layout_he.	content M
		In Space				textview	Textilian
		B Password	1	TextView		fr anna	Texteren
		* Password (Numeric)	- 2			> text	
		@ E-mail				contentD_	
		A Postal Address				► textAp_	sterial.Small
2		E Multiline Text	R.			Favorite A	ttributes
orte		© Time	÷			visibility	none
2 M							
*	d Tre	TextVew	8				
2	Joner			1			
Varia	Com						adro
pand	10	TextView	8			View all pr	roperties + [→]
*	Design Text						del.
🚝 👷 Messages 📓 Terminal 🛛 🖶 👰 Android Monitor 🛛 🔮 TODO					2 Q	vent Log	Gradle Console
Gradle build finished in 5s 367ms (11 minutes ago)					n/a n/a	Context <no< td=""><td>e context - Te 📲</td></no<>	e context - Te 📲

Schalte wieder über den Reiter unten links um in die Textansicht, wandele "TextView" um in "Welcome to the second page!" und erstelle wie oben erläutert einen weiteren String. Auf der zweiten Seite steht nun der gewünschte Text.

MyApplication - [C:\Users\gstel\AndroidStudioProjects\MyApplication] - [app]\app\src\main\res\)ayout\activity_second.xml - Android Studio 2.3.1	- D X
Eile Edit View Navigate Code Analyze Befactor Build Run I	ols VCS Window Help	
	··▶ + # # B # 8 8 Ca & ?	Q. E
MyApplication 🕞 app 🗇 src 🖸 main 🕞 res 🔁 layout	activity_second.xml	
मु 🟺 Android 🔹 😳 🕂 🕸 - !**	😑 activity_main.xml × 🧧 activity_second.xml × 🌀 SecondActivity.java × 🌀 MainActivity.java	X Preview Q1 @
Y Comparing and the second Activity Y Commit untravpapplication Y Commit untravpapplication (Individ/Test) Y Commit untravpapplication (Individ/Test)	android.support.constraint.ConstraintLayout TextView *2ml version="1.0" encoding="uft-8-7> *2ml version="1.0" encoding="uft-8-7> *2mlns:app="http://schemas.android.com/apk/res-auto" mins:tools="thtp://schemas.android.com/apk/res-auto" android.layout.uidth="match_parent" android.layout.paight="match_parent" android.layout.paight="match_parent" android.layout.paight="match_parent" android.layout.paight="match_parent" android.layout.paight="match_parent" android.layout.paight="match_parent" android.layout.paight="match_parent" android.layout.paight="match_parent" tools:layout.paight="match_parent" tools:layout.paight="match_parent" tools:layout.dailor_aboututs="247dp" /> 	Image: Second page Image: Second page
	Design Text	
🗮 Q: Messages 📓 Terminal 🐥 §: Android Monitor 🔮 TODO		💷 Event Log 🖉 Gradie Console
Gradie build finished in 5s 367ms (20 minutes ago)		1357 CRLF= UTF-B Context <no-context> 'a 悟</no-context>

Stufe 5: Aktivieren des Schaltknopfs

Wähle den "MainActivity.java" Tabulator und ergänze folgende Zeilen zwischen der letzten Textzeile und den beiden geschweiften Klammern:

```
Button button = (Button) findViewById(R.id.button);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        int ce = v.getId();
        if(ce == R.id.button) {
            Intent intent = new Intent(MainActivity.this, SecondActivity.class);
            startActivity(intent);
        }
```

} });

Zusätzlich müssen weiter oben im Programmcode folgenden Import-Funktionen ergänzt

```
import android.view.View;
import android.widget.Button;
```

und danach im Code nach rot eingefärbten Begriffen suchen, in diesem Fall "Intend". Setze mit einem Klick den Cursor auf einen der beiden eingefärbten Begriffe und betätige die Tastenkombination "Alt + Entfernen". Die Anpassung erfolgt automatisch.

Hier das vollständige Java-Script:



Auf der zweiten Seite lassen sich weitere Schaltknöpfe einrichten, mit denen man z.B. zur ersten zurückkehren oder eine Internetverbindung aufrufen kann. Hier der passende Code zum Script "SecondActivity.java":

```
import android.content.Intent;
import android.net.Uri;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
public class SecondActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);
        Button button = (Button) findViewById(R.id.button3);
        //Button button2 = (Button) findViewById(R.id.button2);
        button.setOnClickListener(new View.OnClickListener() {
```

package de.mint_unt.myapplication;

```
@Override
            public void onClick(View view)
                int ce = view.getId();
                    if (ce == R.id.button3) {
                    Intent intent = new Intent(Intent.ACTION_VIEW,
                            Uri.parse("http://www.mint-unt.de"));
                    startActivity(intent);
                    }
                }
        });
        Button button1 = (Button) findViewById(R.id.button2);
        button1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                int cc = v.getId();
                if(cc == R.id.button2){
                    Intent intent = new Intent(SecondActivity.this, MainActivity.class);
                    startActivity(intent);
                }
            }
        });
    }
}
```

Stufe 6: Hintergrundfarben ändern

- Zum Ändern der Hintergrundfarbe bei den **Schaltknöpfen** im Code der jeweiligen Seite unter "<Button" folgende Zeile ergänzen: "android:background="#FFFFFF". Die HTML-Codes der Farben finden sich im Internet. #FFFFFF ist ein sehr helles Grau (grey100, fast weiß).
- Zum Ändern der Hintergrundfarbe des **App-Screens** wird die gleiche Befehlszeile unter *"*<android.support" eingefügt.

Stufe 7: Eigene Bilder zu platzieren

Zur Bezeichnung von Bilddateien dürfen nur Unterstriche, Kleinbuchstaben und Zahlen verwendet werden. In der Design-Ansicht kann über >images ein >imageView auf dem virtuellen App-Screen platziert werden. In dem sich öffnenden Dialog kann das Bild ausgewählt werden, sofern es unter dem voreingestellten Pfad gespeichert wurde.

Speicherort: C:\Users\gstei\AndroidStudioProjects\MyApplication\app\src\main\res\drawable

Zum Speichern von Icons funktioniert folgender Weg:

- 1. Rechtsklick auf res, new Image Asset
- 2. In Icon Type wähle Action Bar and Tab Icons
- 3. In Asset type wähle Image
- 4. In Path wähle den Pfad zum Bild auf deinem Rechner
- 5. Next->Finish

Stufe 7: 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 Zick-zack-Linien angezeigt.



Objekte platziert man relativ zueinander, indem man an passenden Seiten die Kreise übereinander zieht. Ein rotes Aufleuchten zeigt die Verbindung an.

Löschen kann man die Verbindungen durch einen Klick im blauen Screen auf das Kreuz links unterhalb des Objekts, das beim Anwählen eingeblendet wird. Näheres findet sich <u>hier</u>.