

# **GRAPHICAL USER INTERFACE (GUI)**

# WEB GUI

← → C (

---

Hello world

# MOBILE APPS GUI

## ANDROID



# DESKTOP GUI

.NET



# NATIVE APPS

- Developed for a specific platform (iOS, Android, etc.)
- Pros: performance, user experience, APIs access
- Cons: separate code per platform
- Example: Kotlin/Java for Android

# CROSS-PLATFORM

- Write once, run anywhere philosophy
- Pros: shared code for all supported platforms
- Cons: harder to achieve the pros of native apps
- Example: .NET MAUI

# GUI PROGRAMMING

- Visual items are essentially ... OBJECTs!
  - Window, button, label, etc.
- Have special members for drawing the object
- We can use these members to customize our objects

# ANDROID (JAVA/KOTLIN)

```
1 public class MainActivity extends Activity {  
2  
3     @Override  
4     protected void onCreate(Bundle savedInstanceState) {  
5         super.onCreate(savedInstanceState);  
6  
7         TextView textView = new TextView(this);  
8         textView.setText("Hello World!");  
9  
10        LinearLayout linearLayout = new LinearLayout(this);  
11  
12        linearLayout.addView(textView);  
13        setContentView(linearLayout);  
14    }  
15 }
```



# ANDROID (JAVA/KOTLIN)

```
1 public class MainActivity extends Activity {  
2  
3     @Override  
4     protected void onCreate(Bundle savedInstanceState) {  
5         super.onCreate(savedInstanceState);  
6  
7         TextView textView = new TextView(this);  
8         textView.setText("Hello World!");  
9  
10        LinearLayout linearLayout = new LinearLayout(this);  
11  
12        linearLayout.addView(textView);  
13        setContentView(linearLayout);  
14    }  
15 }
```



# ANDROID (JAVA/KOTLIN)

```
1 public class MainActivity extends Activity {  
2  
3     @Override  
4     protected void onCreate(Bundle savedInstanceState) {  
5         super.onCreate(savedInstanceState);  
6  
7         TextView textView = new TextView(this);  
8         textView.setText("Hello World!");  
9  
10        LinearLayout linearLayout = new LinearLayout(this);  
11  
12        linearLayout.addView(textView);  
13        setContentView(linearLayout);  
14    }  
15 }
```



# ANDROID (JAVA/KOTLIN)

```
1 public class MainActivity extends Activity {  
2  
3     @Override  
4     protected void onCreate(Bundle savedInstanceState) {  
5         super.onCreate(savedInstanceState);  
6  
7         TextView textView = new TextView(this);  
8         textView.setText("Hello World!");  
9  
10        LinearLayout linearLayout = new LinearLayout(this);  
11  
12        linearLayout.addView(textView);  
13        setContentView(linearLayout);  
14    }  
15 }
```



# ANDROID (JAVA/KOTLIN)

```
1 public class MainActivity extends Activity {  
2  
3     @Override  
4     protected void onCreate(Bundle savedInstanceState) {  
5         super.onCreate(savedInstanceState);  
6  
7         TextView textView = new TextView(this);  
8         textView.setText("Hello World!");  
9  
10        LinearLayout linearLayout = new LinearLayout(this);  
11  
12        linearLayout.addView(textView);  
13        setContentView(linearLayout);  
14    }  
15 }
```



# ANDROID (JAVA/KOTLIN)

```
1 public class MainActivity extends Activity {  
2  
3     @Override  
4     protected void onCreate(Bundle savedInstanceState) {  
5         super.onCreate(savedInstanceState);  
6  
7         TextView textView = new TextView(this);  
8         textView.setText("Hello World!");  
9  
10        LinearLayout linearLayout = new LinearLayout(this);  
11  
12        linearLayout.addView(textView);  
13        setContentView(linearLayout);  
14    }  
15 }
```



# WEB (JAVASCRIPT)

← → C |

---

Hello world

```
1 var txt = document.createTextNode("Hello world");
2 document.body.appendChild(txt);
```

# DEVELOPING GUI VIA CODE

- Doable, but usually not a best practice
- Hard to read and maintain
- Requires knowledge of programming
  - Especially of the specific programming language

# DEVELOPING GUI VIA CODE

- Instead, separate UI code from logic code
- Use a **markup** language
  - Markup language describes the objects to be created in a readable and flexible format
  - But, there will still be objects - so you can still also work with the objects if needed

# XML

## EXTENSIBLE MARKUP LANGUAGE

- Useful for storing and transmitting data
- The data is structured using **tags**
- Tag names can be whatever you want

# XML

## EXTENSIBLE MARKUP LANGUAGE

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <foo>
3   <bar>Hello, World!</bar>
4 </foo>
```

# XML

## EXTENSIBLE MARKUP LANGUAGE

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <foo>
3   <bar>Hello, World!</bar>
4 </foo>
```

# XML

## EXTENSIBLE MARKUP LANGUAGE

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <foo>
3   <bar>Hello, World!</bar>
4 </foo>
```

# XML

## EXTENSIBLE MARKUP LANGUAGE

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <foo>
3   <bar>Hello, World!</bar>
4 </foo>
```

# XML

## EXTENSIBLE MARKUP LANGUAGE

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <news_articles>
3   <article>
4     <headline>World's Largest Foo Created in Foobar Event</headline>
5     <author>Baz Caz</author>
6     <category>Mystery</category>
7     <publication_date>2024-01-11</publication_date>
8   </article>
9   <article>
10    <headline>Unicorn Called Bar in West of Foobar</headline>
11    <author>Foo foo</author>
12    <category>Nature</category>
13    <publication_date>2024-01-10</publication_date>
14  </article>
15 </news_articles>
```

# XML

## EXTENSIBLE MARKUP LANGUAGE

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <news_articles>
3   <article>
4     <headline>World's Largest Foo Created in Foobar Event</headline>
5     <author>Baz Caz</author>
6     <category>Mystery</category>
7     <publication_date>2024-01-11</publication_date>
8   </article>
9   <article>
10    <headline>Unicorn Called Bar in West of Foobar</headline>
11    <author>Foo foo</author>
12    <category>Nature</category>
13    <publication_date>2024-01-10</publication_date>
14  </article>
15 </news_articles>
```

# XML

## EXTENSIBLE MARKUP LANGUAGE

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <news_articles>
3   <article>
4     <headline>World's Largest Foo Created in Foobar Event</headline>
5     <author>Baz Caz</author>
6     <category>Mystery</category>
7     <publication_date>2024-01-11</publication_date>
8   </article>
9   <article>
10    <headline>Unicorn Called Bar in West of Foobar</headline>
11    <author>Foo foo</author>
12    <category>Nature</category>
13    <publication_date>2024-01-10</publication_date>
14  </article>
15 </news_articles>
```

# XML

## EXTENSIBLE MARKUP LANGUAGE

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <news_articles>
3   <article>
4     <headline>World's Largest Foo Created in Foobar Event</headline>
5     <author>Baz Caz</author>
6     <category>Mystery</category>
7     <publication_date>2024-01-11</publication_date>
8   </article>
9   <article>
10    <headline>Unicorn Called Bar in West of Foobar</headline>
11    <author>Foo foo</author>
12    <category>Nature</category>
13    <publication_date>2024-01-10</publication_date>
14  </article>
15 </news_articles>
```

# XML

## EXTENSIBLE MARKUP LANGUAGE

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <news_articles>
3   <article>
4     <headline>World's Largest Foo Created in Foobar Event</headline>
5     <author>Baz Caz</author>
6     <category>Mystery</category>
7     <publication_date>2024-01-11</publication_date>
8   </article>
9   <article>
10    <headline>Unicorn Called Bar in West of Foobar</headline>
11    <author>Foo foo</author>
12    <category>Nature</category>
13    <publication_date>2024-01-10</publication_date>
14  </article>
15 </news_articles>
```

# XML IN ANDROID

## RESOURCE FILES

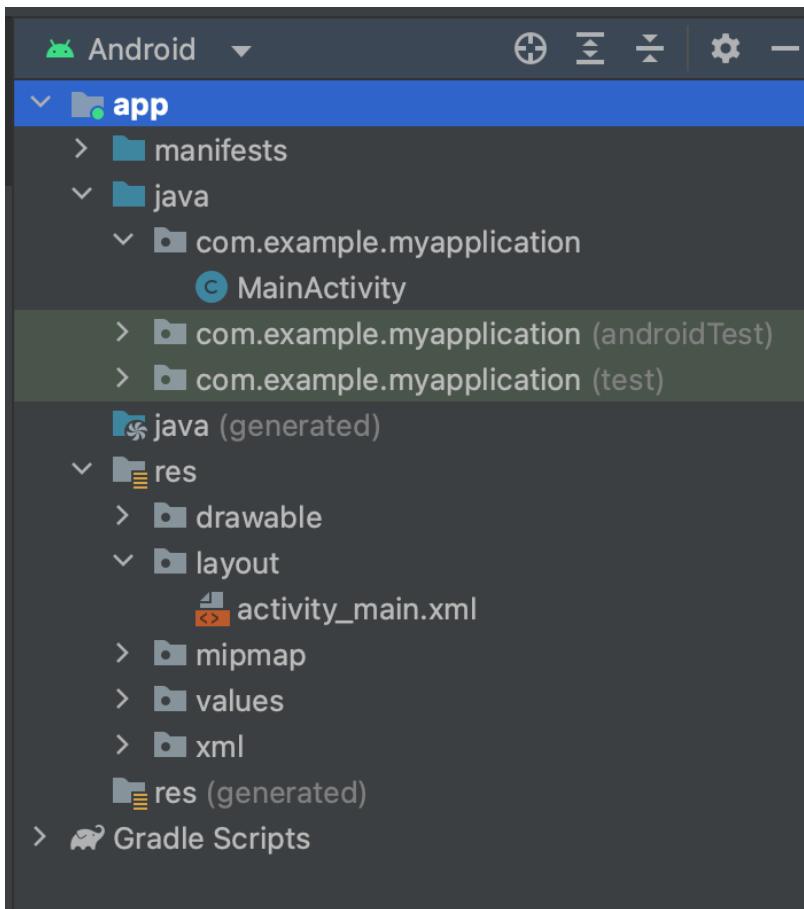
- In Android, XML is used a lot
- Examples:
  - To define UI, i.e., **layouts**
  - To define resources

# GUI VIA XML

```
1 public class MainActivity extends Activity {  
2  
3     @Override  
4     protected void onCreate(Bundle savedInstanceState) {  
5         super.onCreate(savedInstanceState);  
6  
7         TextView textView = new TextView(this);  
8         textView.setText("Hello World!");  
9  
10        LinearLayout linearLayout = new LinearLayout(this);  
11  
12        linearLayout.addView(textView);  
13        setContentView(linearLayout);  
14    }  
15 }
```



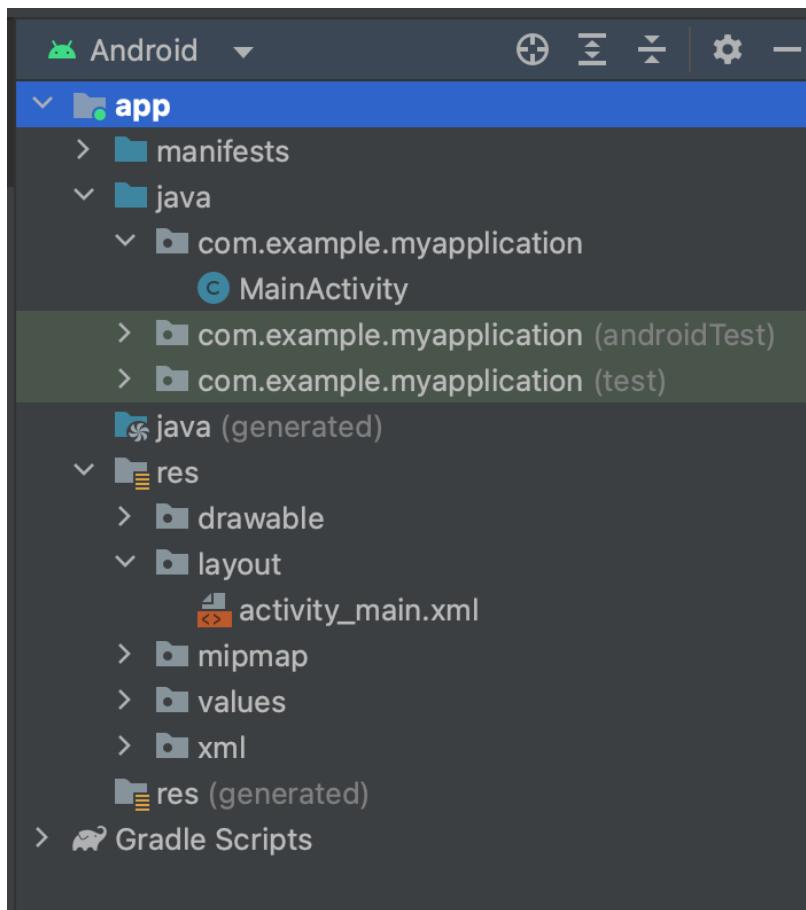
# GUI VIA XML



```
1 public class MainActivity extends Activity {  
2  
3     @Override  
4     protected void onCreate(Bundle savedInstanceState) {  
5         super.onCreate(savedInstanceState);  
6         setContentView(R.layout.activity_main);  
7     }  
9 }
```

```
1 <?xml version="1.0" encoding="utf-8"?>  
2  
3 <LinearLayout  
4     xmlns:android="http://schemas.android.com/apk/res/android"  
5     android:layout_width="match_parent"  
6     android:layout_height="match_parent"  
7     android:orientation="vertical">  
8  
9     <TextView  
10        android:layout_width="match_parent"  
11        android:layout_height="wrap_content"  
12        android:text="Hello world" />  
13  
14 </LinearLayout>
```

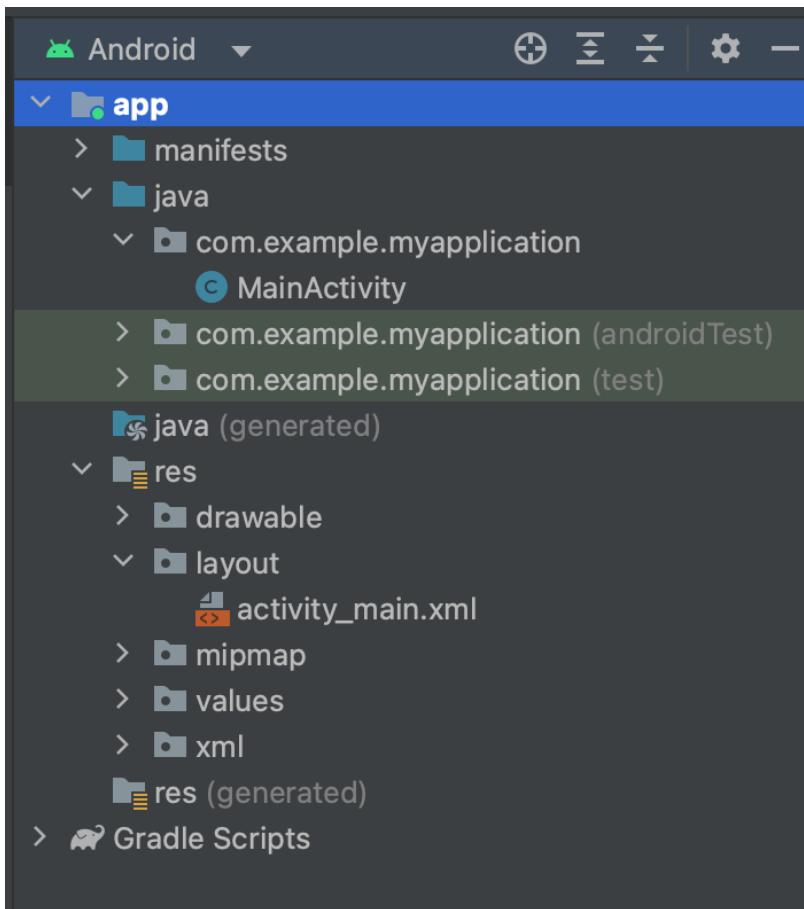
# GUI VIA XML



```
1 public class MainActivity extends Activity {  
2  
3     @Override  
4     protected void onCreate(Bundle savedInstanceState) {  
5         super.onCreate(savedInstanceState);  
6         setContentView(R.layout.activity_main);  
7     }  
9 }
```

```
1 <?xml version="1.0" encoding="utf-8"?>  
2  
3 <LinearLayout  
4     xmlns:android="http://schemas.android.com/apk/res/android"  
5     android:layout_width="match_parent"  
6     android:layout_height="match_parent"  
7     android:orientation="vertical">  
8  
9     <TextView  
10        android:layout_width="match_parent"  
11        android:layout_height="wrap_content"  
12        android:text="Hello world" />  
13  
14 </LinearLayout>
```

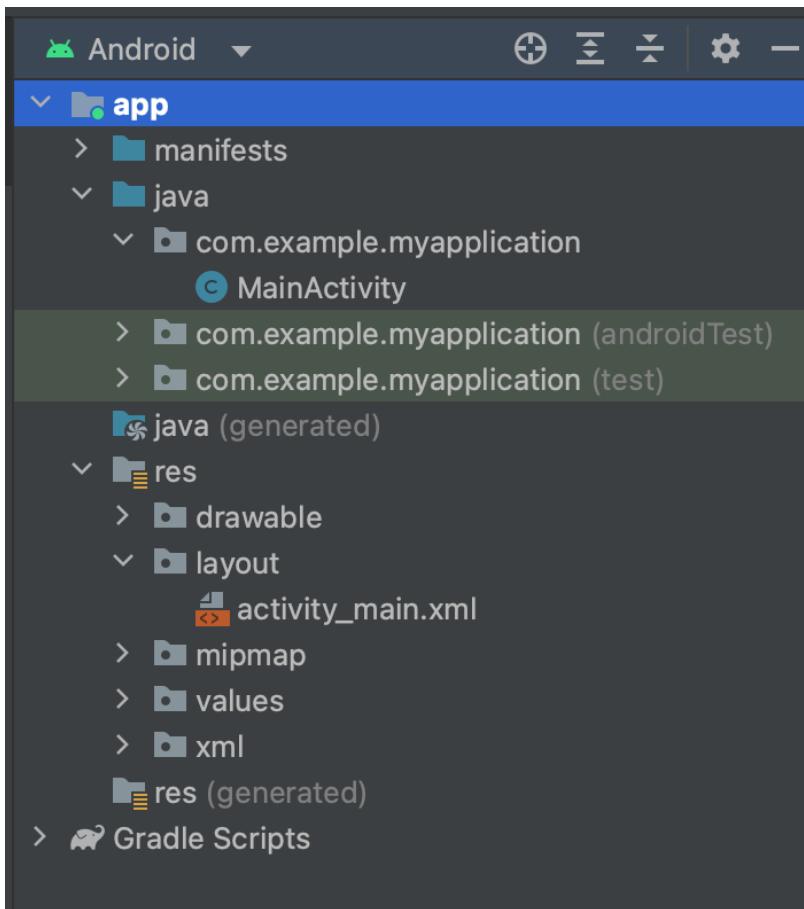
# GUI VIA XML



```
1 public class MainActivity extends Activity {  
2  
3     @Override  
4     protected void onCreate(Bundle savedInstanceState) {  
5         super.onCreate(savedInstanceState);  
6         setContentView(R.layout.activity_main);  
7     }  
9 }
```

```
1 <?xml version="1.0" encoding="utf-8"?>  
2  
3 <LinearLayout  
4     xmlns:android="http://schemas.android.com/apk/res/android"  
5     android:layout_width="match_parent"  
6     android:layout_height="match_parent"  
7     android:orientation="vertical">  
8  
9     <TextView  
10        android:layout_width="match_parent"  
11        android:layout_height="wrap_content"  
12        android:text="Hello world" />  
13  
14 </LinearLayout>
```

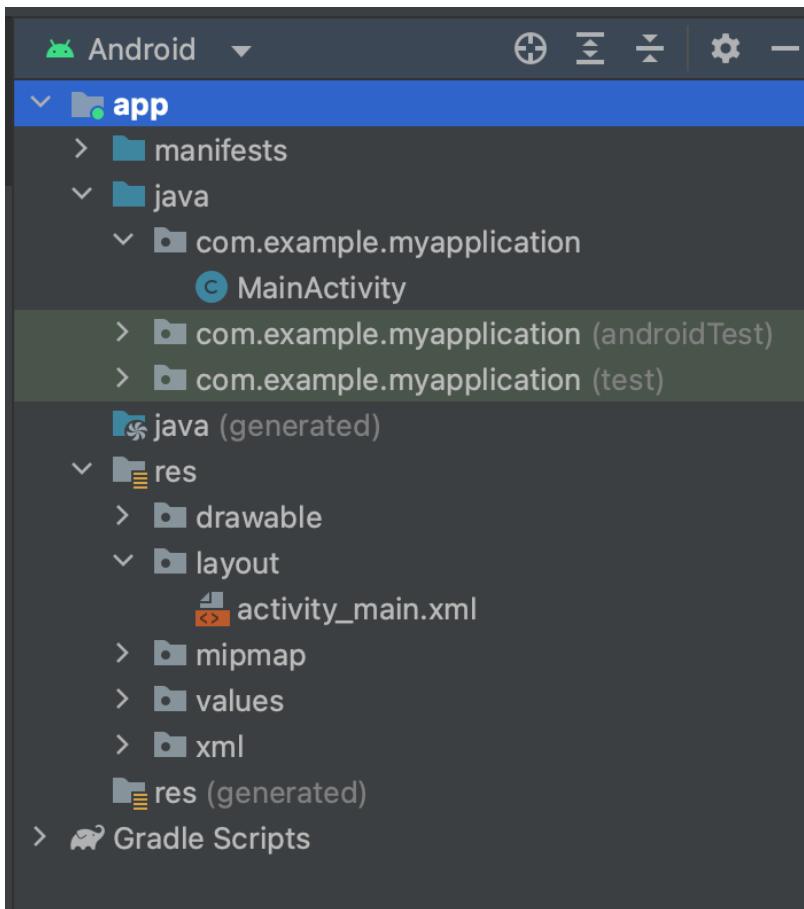
# GUI VIA XML



```
1 public class MainActivity extends Activity {  
2  
3     @Override  
4     protected void onCreate(Bundle savedInstanceState) {  
5         super.onCreate(savedInstanceState);  
6         setContentView(R.layout.activity_main);  
7     }  
9 }
```

```
1 <?xml version="1.0" encoding="utf-8"?>  
2  
3 <LinearLayout  
4     xmlns:android="http://schemas.android.com/apk/res/android"  
5     android:layout_width="match_parent"  
6     android:layout_height="match_parent"  
7     android:orientation="vertical">  
8  
9     <TextView  
10        android:layout_width="match_parent"  
11        android:layout_height="wrap_content"  
12        android:text="Hello world" />  
13  
14 </LinearLayout>
```

# GUI VIA XML



```
1 public class MainActivity extends Activity {  
2  
3     @Override  
4     protected void onCreate(Bundle savedInstanceState) {  
5         super.onCreate(savedInstanceState);  
6         setContentView(R.layout.activity_main);  
7     }  
9 }
```

```
1 <?xml version="1.0" encoding="utf-8"?>  
2  
3 <LinearLayout  
4     xmlns:android="http://schemas.android.com/apk/res/android"  
5     android:layout_width="match_parent"  
6     android:layout_height="match_parent"  
7     android:orientation="vertical">  
8  
9     <TextView  
10        android:layout_width="match_parent"  
11        android:layout_height="wrap_content"  
12        android:text="Hello world" />  
13  
14 </LinearLayout>
```

# GUI VIA HTML

## HYPertext Markup Language

- HTML describes the structure and content of a web page
  - Can also contain design elements, but better not to
- HTML uses **tags** to describe HTML **elements**
- The browser takes HTML file and displays it visually

# GUI VIA HTML

```
1 var txt = document.createTextNode("Hello world");
2 document.body.appendChild(txt);
```

```
1 <!DOCTYPE html>
2 <html>
3   <head></head>
4   <body>
5     Hello world
6   </body>
7 </html>
```

# GUI VIA HTML

```
1 var txt = document.createTextNode("Hello world");
2 document.body.appendChild(txt);
```

```
1 <!DOCTYPE html>
2 <html>
3   <head></head>
4   <body>
5     Hello world
6   </body>
7 </html>
```

# GUI VIA HTML

```
1 var txt = document.createTextNode("Hello world");
2 document.body.appendChild(txt);
```

```
1 <!DOCTYPE html>
2 <html>
3   <head></head>
4   <body>
5     Hello world
6   </body>
7 </html>
```

# GUI VIA HTML

```
1 var txt = document.createTextNode("Hello world");
2 document.body.appendChild(txt);
```

```
1 <!DOCTYPE html>
2 <html>
3   <head></head>
4   <body>
5     Hello world
6   </body>
7 </html>
```

# GUI VIA HTML

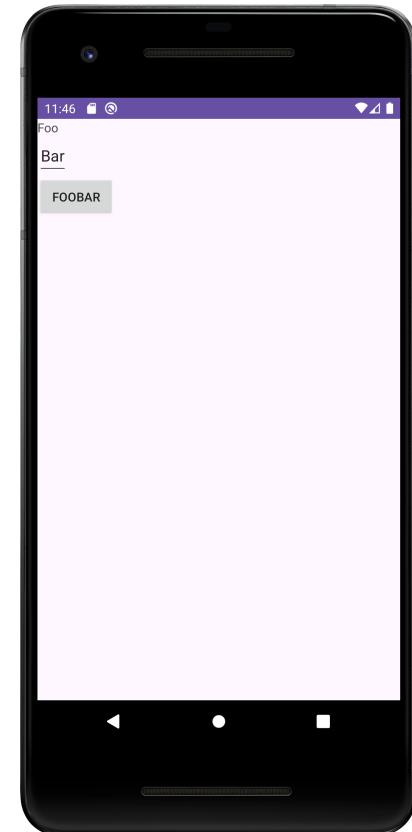
```
1 var txt = document.createTextNode("Hello world");
2 document.body.appendChild(txt);
```

```
1 <!DOCTYPE html>
2 <html>
3   <head></head>
4   <body>
5     Hello world
6   </body>
7 </html>
```

# GUI ELEMENTS

## ANDROID

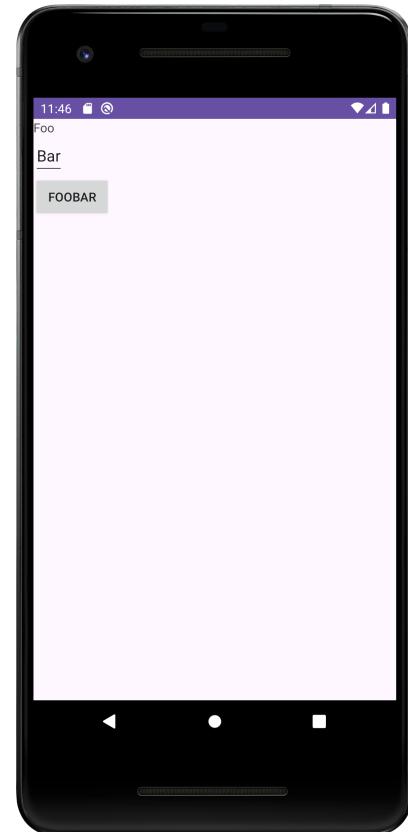
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical" >
7     <TextView
8         android:layout_width="wrap_content"
9         android:layout_height="wrap_content"
10        android:text="Foo" />
11     <EditText
12         android:layout_width="wrap_content"
13         android:layout_height="wrap_content"
14         android:text="Bar" />
15     <Button
16         android:layout_width="wrap_content"
17         android:layout_height="wrap_content"
18         android:text="Foobar" />
19 </LinearLayout>
```



# GUI ELEMENTS

## ANDROID

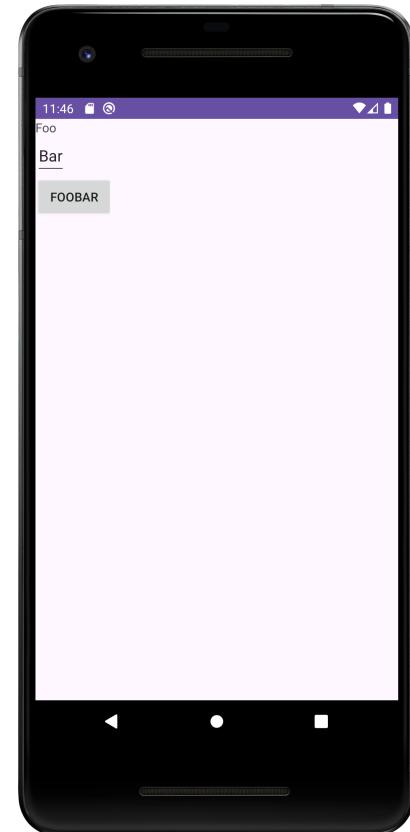
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical" >
7     <TextView
8         android:layout_width="wrap_content"
9         android:layout_height="wrap_content"
10        android:text="Foo" />
11     <EditText
12         android:layout_width="wrap_content"
13         android:layout_height="wrap_content"
14         android:text="Bar" />
15     <Button
16         android:layout_width="wrap_content"
17         android:layout_height="wrap_content"
18         android:text="Foobar" />
19 </LinearLayout>
```



# GUI ELEMENTS

## ANDROID

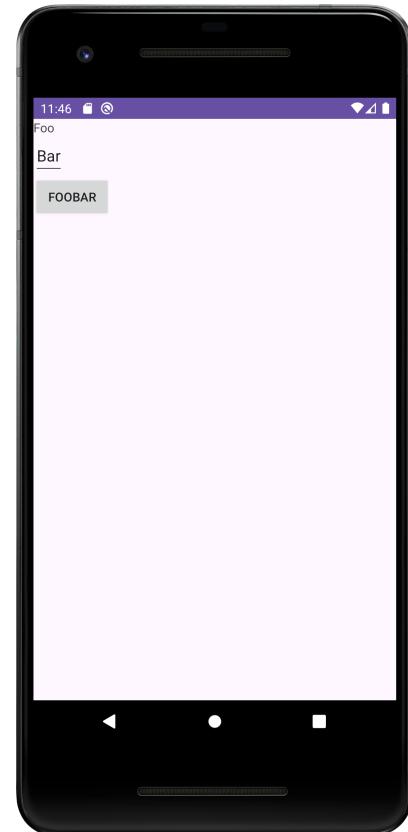
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical" >
7     <TextView
8         android:layout_width="wrap_content"
9         android:layout_height="wrap_content"
10        android:text="Foo" />
11     <EditText
12         android:layout_width="wrap_content"
13         android:layout_height="wrap_content"
14         android:text="Bar" />
15     <Button
16         android:layout_width="wrap_content"
17         android:layout_height="wrap_content"
18         android:text="Foobar" />
19 </LinearLayout>
```



# GUI ELEMENTS

## ANDROID

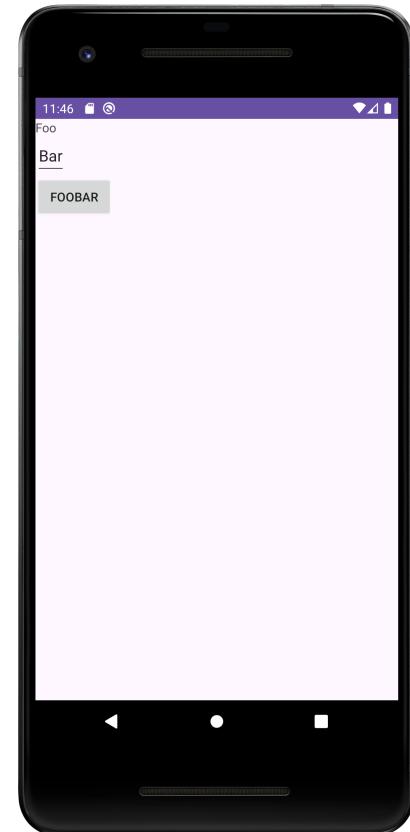
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical" >
7     <TextView
8         android:layout_width="wrap_content"
9         android:layout_height="wrap_content"
10        android:text="Foo" />
11     <EditText
12         android:layout_width="wrap_content"
13         android:layout_height="wrap_content"
14         android:text="Bar" />
15     <Button
16         android:layout_width="wrap_content"
17         android:layout_height="wrap_content"
18         android:text="Foobar" />
19 </LinearLayout>
```



# GUI ELEMENTS

## ANDROID

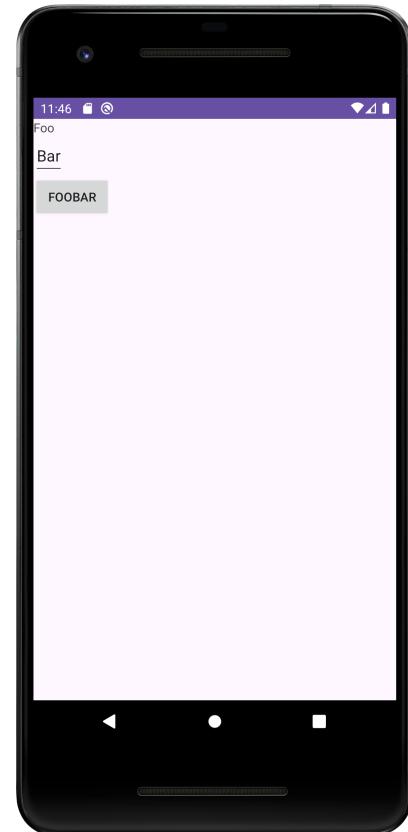
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical" >
7     <TextView
8         android:layout_width="wrap_content"
9         android:layout_height="wrap_content"
10        android:text="Foo" />
11     <EditText
12         android:layout_width="wrap_content"
13         android:layout_height="wrap_content"
14         android:text="Bar" />
15     <Button
16         android:layout_width="wrap_content"
17         android:layout_height="wrap_content"
18         android:text="Foobar" />
19 </LinearLayout>
```



# GUI ELEMENTS

## ANDROID

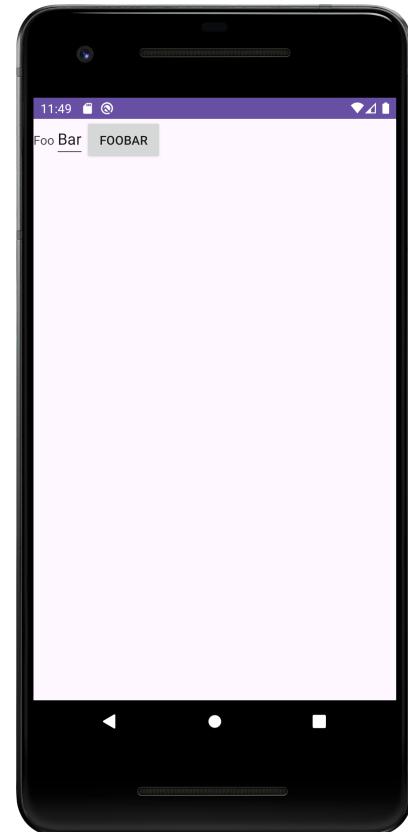
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical" >
7     <TextView
8         android:layout_width="wrap_content"
9         android:layout_height="wrap_content"
10        android:text="Foo" />
11     <EditText
12         android:layout_width="wrap_content"
13         android:layout_height="wrap_content"
14         android:text="Bar" />
15     <Button
16         android:layout_width="wrap_content"
17         android:layout_height="wrap_content"
18         android:text="Foobar" />
19 </LinearLayout>
```



# GUI ELEMENTS

## ANDROID

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="horizontal" >
7     <TextView
8         android:layout_width="wrap_content"
9         android:layout_height="wrap_content"
10        android:text="Foo" />
11     <EditText
12         android:layout_width="wrap_content"
13         android:layout_height="wrap_content"
14         android:text="Bar" />
15     <Button
16         android:layout_width="wrap_content"
17         android:layout_height="wrap_content"
18         android:text="Foobar" />
19 </LinearLayout>
```



# GUI ELEMENTS

## WEB

```
1 <html>
2   <head></head>
3   <body>
4     <span>Foo</span>
5     <input value="Bar"></input>
6     <button>Foobar</button>
7   </body>
8 </html>
```

Foo  Foobar

# GUI ELEMENTS

## WEB

```
1 <html>
2   <head></head>
3   <body>
4     <span>Foo</span>
5     <input value="Bar"></input>
6     <button>Foobar</button>
7   </body>
8 </html>
```

Foo  Foobar

# GUI ELEMENTS

## WEB

```
1 <html>
2   <head></head>
3   <body>
4     <span>Foo</span>
5     <input value="Bar"></input>
6     <button>Foobar</button>
7   </body>
8 </html>
```

Foo  Foobar

# GUI ELEMENTS

## WEB

```
1 <html>
2   <head></head>
3   <body>
4     <span>Foo</span>
5     <input value="Bar"></input>
6     <button>Foobar</button>
7   </body>
8 </html>
```

Foo  Foobar

# GUI ELEMENTS

## WEB

```
1 <html>
2   <head></head>
3   <body>
4     <span>Foo</span>
5     <input value="Bar"></input>
6     <button>Foobar</button>
7   </body>
8 </html>
```

Foo  Foobar

# GUI ELEMENTS

## WEB

```
1 <html>
2   <head></head>
3   <body>
4     <span>Foo</span>
5     <input value="Bar"></input>
6     <button>Foobar</button>
7   </body>
8 </html>
```

Foo  Foobar

# GUI ELEMENTS

## WEB

```
1 <html>
2   <head></head>
3   <body>
4     <span>Foo</span>
5     <input value="Bar"></input>
6     <button>Foobar</button>
7   </body>
8 </html>
```

Foo  Foobar

# STYLING

# ANDROID

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical">
7
8     <Button
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:text="Foo"
12        android:background="#00FF00"
13        android:textColor="#FF0000"
14        android:textSize="18pt"
15        android:layout_margin="10dp" />
16
17     <Button
18         android:layout_width="wrap_content"
19         android:layout_height="wrap_content"
20         android:text="Bar" />
21
22     <Button
23         android:layout_width="wrap_content"
24         android:layout_height="wrap_content"
25         android:text="Foobar" />
26 </LinearLayout>
```



# STYLING

# ANDROID

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical">
7
8     <Button
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:text="Foo"
12        android:background="#00FF00"
13        android:textColor="#FF0000"
14        android:textSize="18pt"
15        android:layout_margin="10dp" />
16
17     <Button
18         android:layout_width="wrap_content"
19         android:layout_height="wrap_content"
20         android:text="Bar" />
21
22     <Button
23         android:layout_width="wrap_content"
24         android:layout_height="wrap_content"
25         android:text="Foobar" />
26 </LinearLayout>
```



# STYLING

## WEB

```
1 <html>
2   <head></head>
3   <body>
4     <button style="background-color:#00FF00;
5                   color:#FF0000;
6                   font-size:18pt;
7                   margin:10px">Foo</button>
8     <button>Bar</button>
9     <button>Foobar</button>
10    </body>
11 </html>
```



# STYLING

## WEB

```
1 <html>
2   <head></head>
3   <body>
4     <button style="background-color:#00FF00;
5                   color:#FF0000;
6                   font-size:18pt;
7                   margin:10px">Foo</button>
8     <button>Bar</button>
9     <button>Foobar</button>
10    </body>
11 </html>
```



# STYLING

## WEB

```
1 <html>
2   <head></head>
3   <body>
4     <button style="background-color:#00FF00;
5                   color:#FF0000;
6                   font-size:18pt;
7                   margin:10px">Foo</button>
8     <button style="background-color:#00FF00;
9                   color:#FF0000;
10                  font-size:18pt;
11                  margin:10px">Bar</button>
12    <button style="background-color:#00FF00;
13                   color:#FF0000;
14                   font-size:18pt;
15                   margin:10px">Foobar</button>
16  </body>
17 </html>
```



# STYLING

## USING CSS

```
1 <html>
2   <head>
3     <style>
4       button{
5         background-color:#00FF00;
6         color:#FF0000;
7         font-size:18pt;
8         margin:10px
9       }
10    </style>
11  </head>
12  <body>
13    <button>Foo</button>
14    <button>Bar</button>
15    <button>Foobar</button>
16  </body>
17 </html>
```



# STYLING

## USING CSS

```
1 <html>
2   <head>
3     <style>
4       button{
5         background-color:#00FF00;
6         color:#FF0000;
7         font-size:18pt;
8         margin:10px
9       }
10    </style>
11  </head>
12  <body>
13    <button>Foo</button>
14    <button>Bar</button>
15    <button>Foobar</button>
16  </body>
17 </html>
```



# STYLING

## USING CSS

```
1 <html>
2   <head>
3     <style>
4       button{
5         background-color:#00FF00;
6         color:#FF0000;
7         font-size:18pt;
8         margin:10px
9       }
10    </style>
11  </head>
12  <body>
13    <button>Foo</button>
14    <button>Bar</button>
15    <button>Foobar</button>
16  </body>
17 </html>
```



# STYLING

## USING CSS

```
1 <html>
2   <head>
3     <style>
4       button{
5         background-color:#00FF00;
6         color:#FF0000;
7         font-size:18pt;
8         margin:10px
9       }
10    </style>
11  </head>
12  <body>
13    <button>Foo</button>
14    <button>Bar</button>
15    <button>Foobar</button>
16  </body>
17 </html>
```



# STYLING

## USING CSS

```
1 <html>
2   <head>
3     <style>
4       button{
5         background-color:#00FF00;
6         color:#FF0000;
7         font-size:18pt;
8         margin:10px
9       }
10    </style>
11  </head>
12  <body>
13    <button>Foo</button>
14    <button>Bar</button>
15    <button>Foobar</button>
16  </body>
17 </html>
```



# STYLING

## USING CSS

```
1 <html>
2   <head>
3     <style>
4       button{
5         background-color:#00FF00;
6         color:#FF0000;
7         font-size:18pt;
8         margin:10px
9       }
10    </style>
11  </head>
12  <body>
13    <button>Foo</button>
14    <button>Bar</button>
15    <button>Foobar</button>
16  </body>
17 </html>
```



# STYLING

## USING CSS

```
1 <html>
2   <head>
3     <style>
4       button{
5         background-color:#00FF00;
6         color:#FF0000;
7         font-size:18pt;
8         margin:10px
9       }
10    </style>
11  </head>
12  <body>
13    <button>Foo</button>
14    <button>Bar</button>
15    <button>Foobar</button>
16  </body>
17 </html>
```



# STYLING

## USING CSS

### INDEX.CSS:

```
1 button{  
2   margin:10px;  
3   background-color:#00FF00;  
4   color:#FF0000;  
5   font-size:18pt  
6 }
```

### INDEX.HTML:

```
1 <html>  
2   <head>  
3     <link href="index.css" rel="stylesheet">  
4   </head>  
5   <body>  
6     <button>Foo</button>  
7     <button>Bar</button>  
8     <button>Foobar</button>  
9   </body>  
10 </html>
```



# STYLING

## USING CSS

### INDEX.CSS:

```
1 button{  
2   margin:10px;  
3   background-color:#00FF00;  
4   color:#FF0000;  
5   font-size:18pt  
6 }
```

### INDEX.HTML:

```
1 <html>  
2   <head>  
3     <link href="index.css" rel="stylesheet">  
4   </head>  
5   <body>  
6     <button>Foo</button>  
7     <button>Bar</button>  
8     <button>Foobar</button>  
9   </body>  
10 </html>
```



# CSS

## CASCADING STYLESHEET

### INDEX.CSS:

```
1 button{
2   background-color:#00FF00;
3   color:#FF0000;
4   font-size:18pt;
5   margin:10px;
6 }
7 #danger-btn{
8   background-color:#0000FF;
9 }
10 .green-bg{
11   background-color:#000000;
12 }
```

### INDEX.HTML:

```
1 <html>
2   <head>
3     <link href="index.css" rel="stylesheet">
4   </head>
5   <body>
6     <button>Foo</button>
7     <button class="green-bg">Bar</button>
8     <button class="green-bg" id="danger-btn">Foobar</button>
9   </body>
10 </html>
```



# CSS

## CASCADING STYLESHEET

### INDEX.CSS:

```
1 button{
2   background-color:#00FF00;
3   color:#FF0000;
4   font-size:18pt;
5   margin:10px;
6 }
7 #danger-btn{
8   background-color:#0000FF;
9 }
10 .green-bg{
11   background-color:#000000;
12 }
```

### INDEX.HTML:

```
1 <html>
2   <head>
3     <link href="index.css" rel="stylesheet">
4   </head>
5   <body>
6     <button>Foo</button>
7     <button class="green-bg">Bar</button>
8     <button class="green-bg" id="danger-btn">Foobar</button>
9   </body>
10 </html>
```



# CSS

## CASCADING STYLESHEET

### INDEX.CSS:

```
1 button{
2   background-color:#00FF00;
3   color:#FF0000;
4   font-size:18pt;
5   margin:10px;
6 }
7 #danger-btn{
8   background-color:#0000FF;
9 }
10 .green-bg{
11   background-color:#000000;
12 }
```

### INDEX.HTML:

```
1 <html>
2   <head>
3     <link href="index.css" rel="stylesheet">
4   </head>
5   <body>
6     <button>Foo</button>
7     <button class="green-bg">Bar</button>
8     <button class="green-bg" id="danger-btn">Foobar</button>
9   </body>
10 </html>
```



# CSS

## CASCADING STYLESHEET

### INDEX.CSS:

```
1 button{
2   background-color:#00FF00;
3   color:#FF0000;
4   font-size:18pt;
5   margin:10px;
6 }
7 #danger-btn{
8   background-color:#0000FF;
9 }
10 .green-bg{
11   background-color:#000000;
12 }
```

### INDEX.HTML:

```
1 <html>
2   <head>
3     <link href="index.css" rel="stylesheet">
4   </head>
5   <body>
6     <button>Foo</button>
7     <button class="green-bg">Bar</button>
8     <button class="green-bg" id="danger-btn">Foobar</button>
9   </body>
10 </html>
```



# CSS

## CASCADING STYLESHEET

### INDEX.CSS:

```
1 button{
2   background-color:#00FF00;
3   color:#FF0000;
4   font-size:18pt;
5   margin:10px;
6 }
7 #danger-btn{
8   background-color:#0000FF;
9 }
10 .green-bg{
11   background-color:#000000;
12 }
```

### INDEX.HTML:

```
1 <html>
2   <head>
3     <link href="index.css" rel="stylesheet">
4   </head>
5   <body>
6     <button>Foo</button>
7     <button class="green-bg">Bar</button>
8     <button class="green-bg" id="danger-btn">Foobar</button>
9   </body>
10 </html>
```



# CSS SPECIFICITY ORDER

1. Inline Styles: (1, 0, 0, 0)
  2. ID Selectors: (0, 1, 0, 0)
  3. Class Selectors, attribute Selectors, pseudo-Classes: (0, 0, 1, 0)
  4. Type Selectors, pseudo-Elements: (0, 0, 0, 1)
  5. \*: (0, 0, 0, 0)
- 
- !important is like a joker
  - when more than one definitions with the same specificity: last one wins

# CSS SPECIFICITY ORDER

## EXAMPLES

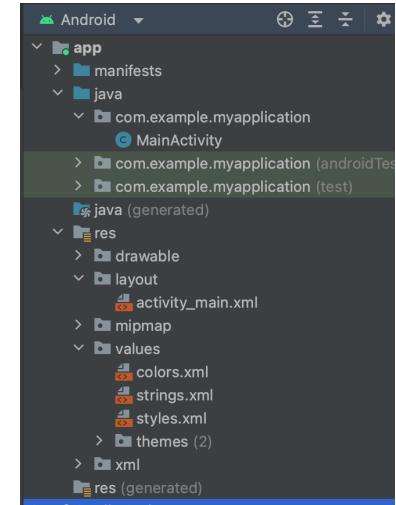
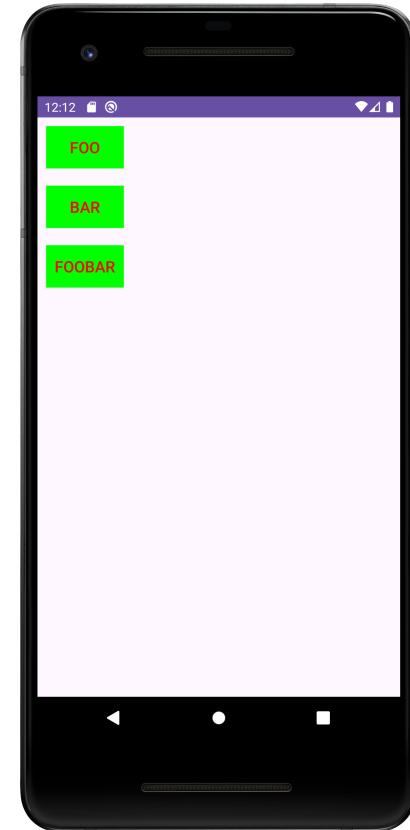
	Score	Explanation
button	1	type selector
.green-bg	10	class selector
#danger-btn	100	id selector
button.green-bg	11	type + class selectors

# STYLING

# ANDROID

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3     <style name="myBtn">
4         <item name="android:background">#00FF00</item>
5         <item name="android:textColor">#FF0000</item>
6         <item name="android:textSize">18sp</item>
7         <item name="android:layout_margin">10dp</item>
8     </style>
9 </resources>
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical">
7
8     <Button
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:text="Foo"
12        style="@style/myBtn" />
13
14    <Button
15        android:layout_width="wrap_content"
16        android:layout_height="wrap_content"
17        android:text="Bar"
18        style="@style/myBtn" />
19
20    <Button
21        android:layout_width="wrap_content"
22        android:layout_height="wrap_content"
23        android:text="Foobar"
24        style="@style/myBtn" />
25
26 </LinearLayout>
```

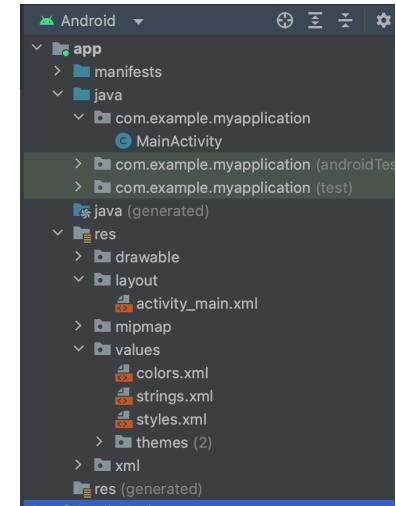
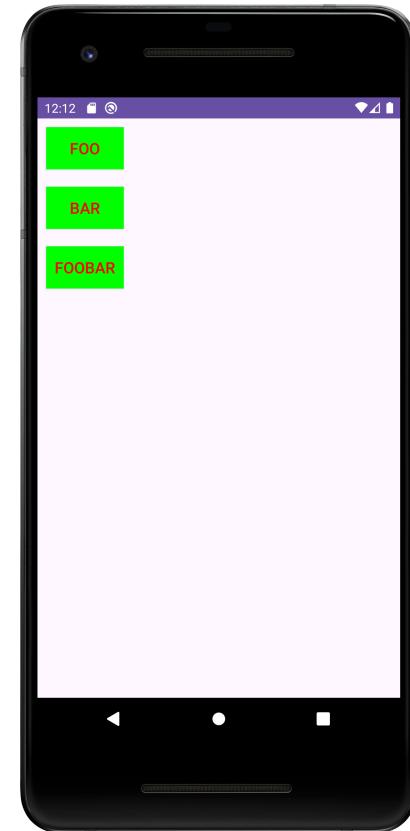


# STYLING

# ANDROID

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3     <style name="myBtn">
4         <item name="android:background">#00FF00</item>
5         <item name="android:textColor">#FF0000</item>
6         <item name="android:textSize">18sp</item>
7         <item name="android:layout_margin">10dp</item>
8     </style>
9 </resources>
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical">
7
8     <Button
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:text="Foo"
12        style="@style/myBtn" />
13
14    <Button
15        android:layout_width="wrap_content"
16        android:layout_height="wrap_content"
17        android:text="Bar"
18        style="@style/myBtn" />
19
20    <Button
21        android:layout_width="wrap_content"
22        android:layout_height="wrap_content"
23        android:text="Foobar"
24        style="@style/myBtn" />
25
26 </LinearLayout>
```

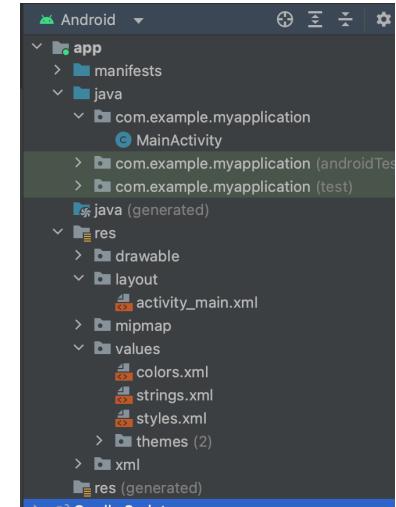
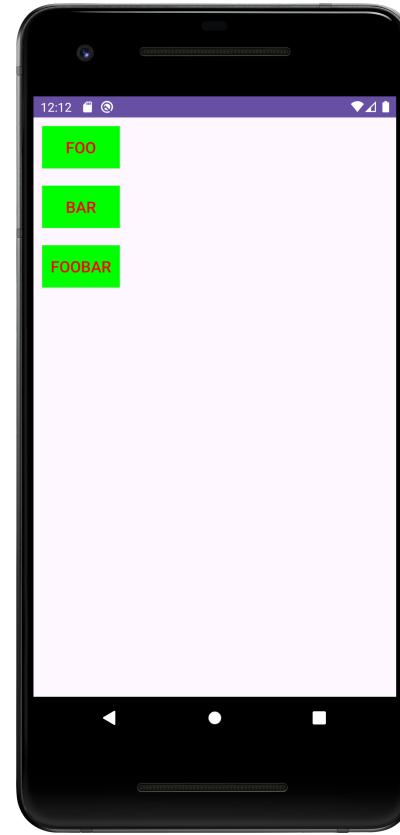


# STYLING

# ANDROID

```
1 <resources>
2     <string name="app_name">My Application</string>
3     <string name="foo">Foo</string>
4     <string name="bar">Bar</string>
5     <string name="foobar">Foobar</string>
6 </resources>
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical">
7
8     <Button
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:text="@string/foo"
12        style="@style/myBtn" />
13
14     <Button
15         android:layout_width="wrap_content"
16         android:layout_height="wrap_content"
17         android:text="@string/bar"
18         style="@style/myBtn" />
19
20     <Button
21         android:layout_width="wrap_content"
22         android:layout_height="wrap_content"
23         android:text="@string/foobar"
24         style="@style/myBtn" />
25 </LinearLayout>
```

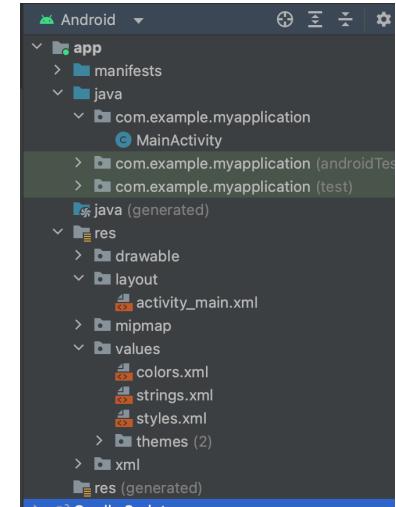
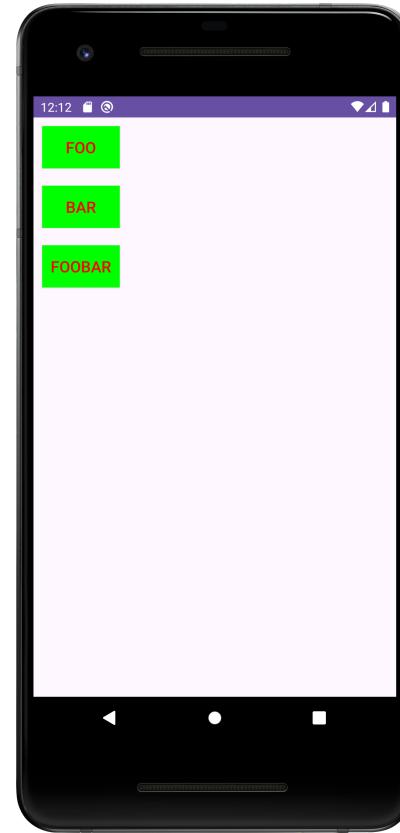


# STYLING

# ANDROID

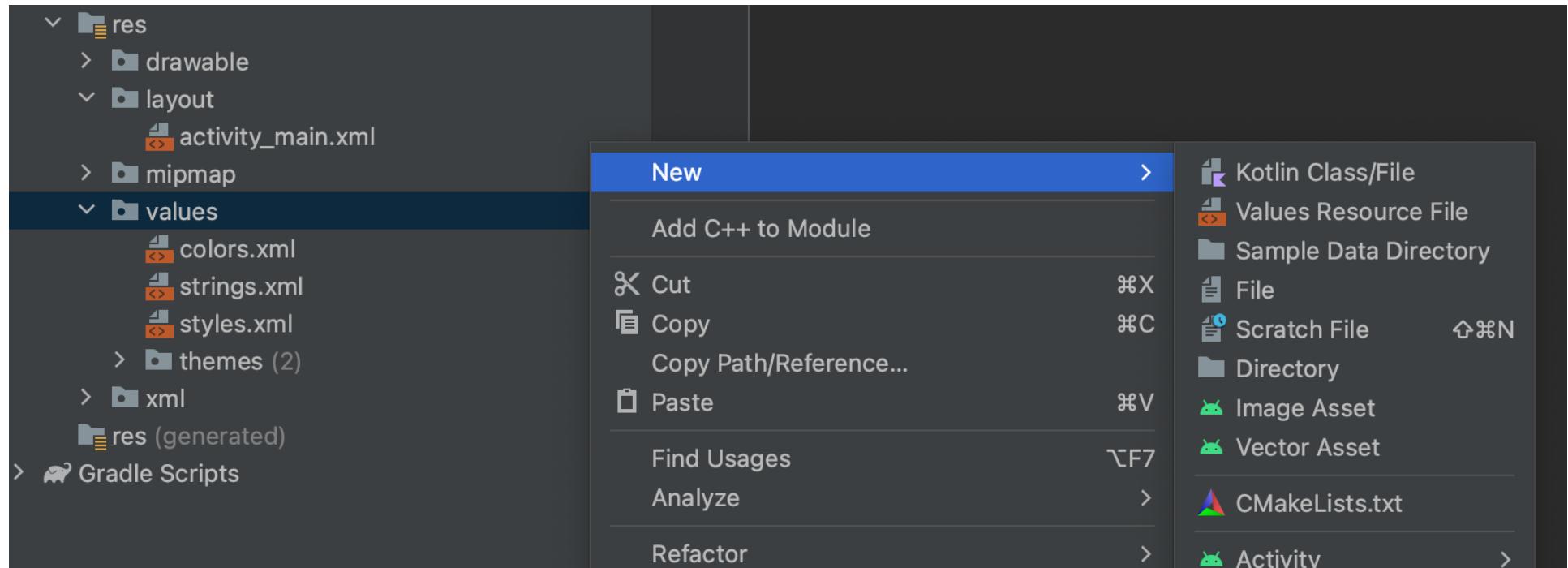
```
1 <resources>
2     <string name="app_name">My Application</string>
3     <string name="foo">Foo</string>
4     <string name="bar">Bar</string>
5     <string name="foobar">Foobar</string>
6 </resources>
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical">
7
8     <Button
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:text="@string/foo"
12        style="@style/myBtn" />
13
14     <Button
15         android:layout_width="wrap_content"
16         android:layout_height="wrap_content"
17         android:text="@string/bar"
18         style="@style/myBtn" />
19
20     <Button
21         android:layout_width="wrap_content"
22         android:layout_height="wrap_content"
23         android:text="@string/foobar"
24         style="@style/myBtn" />
25 </LinearLayout>
```



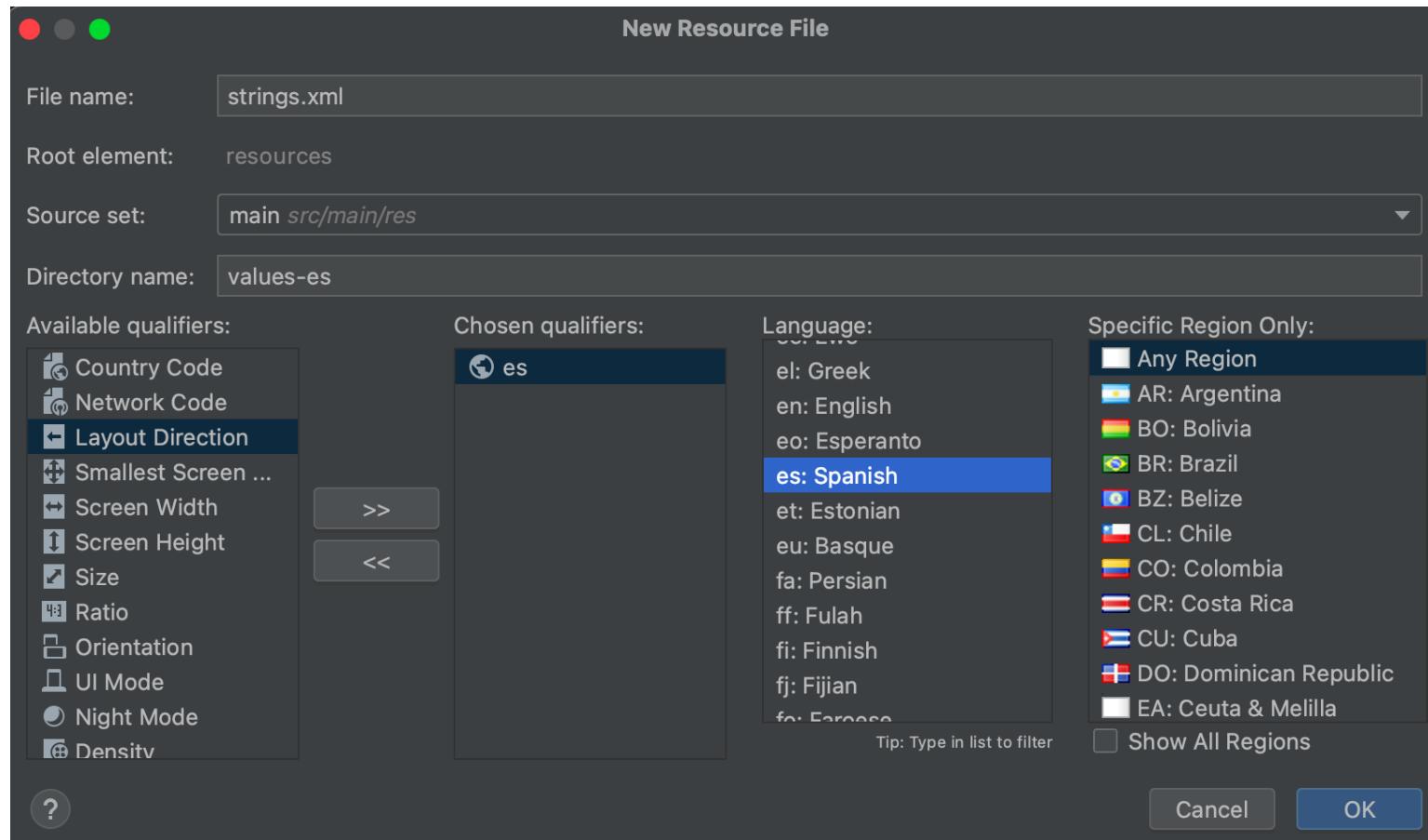
# STYLING

# ANDROID



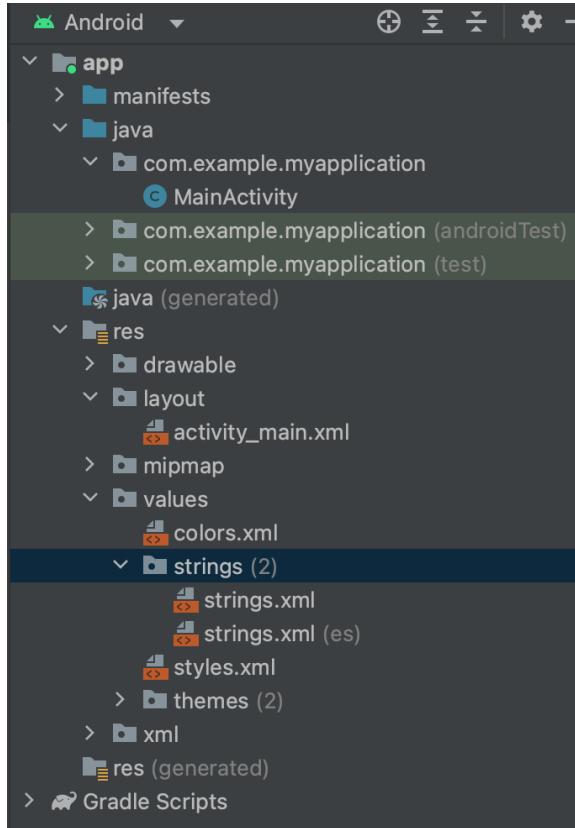
# STYLING

## ANDROID



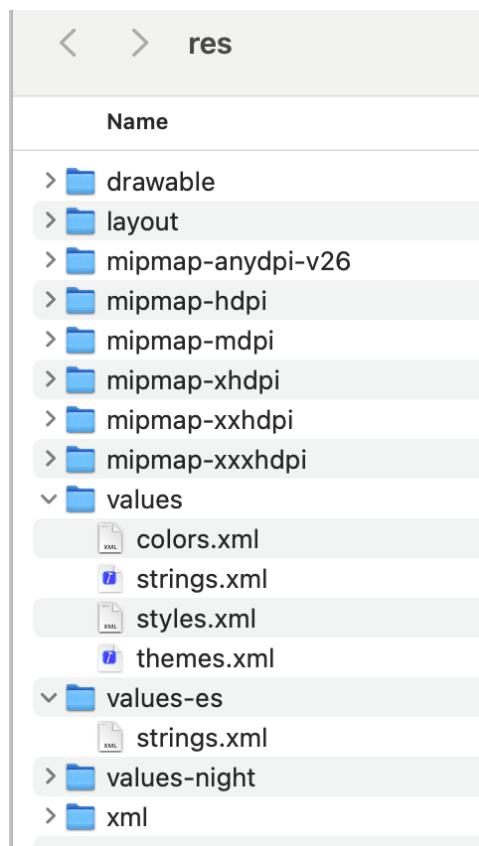
# STYLING

# ANDROID



# STYLING

# ANDROID

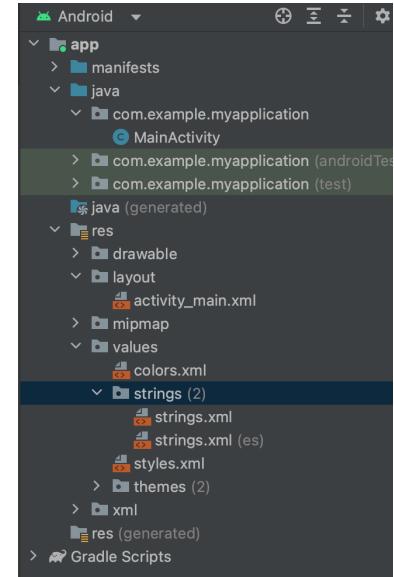
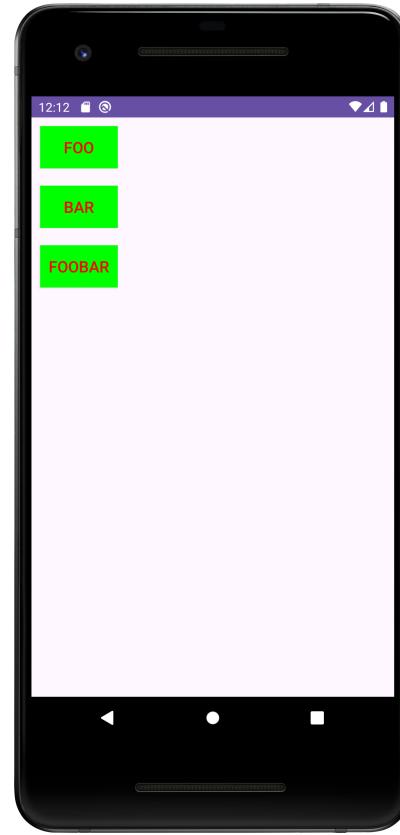


# STYLING

# ANDROID

```
1 <resources>
2     <string name="app_name">My Application</string>
3     <string name="foo">Hola</string>
4     <string name="bar">Cómo</string>
5     <string name="foobar">estás</string>
6 </resources>
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical">
7
8     <Button
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:text="@string/foo"
12        style="@style/myBtn" />
13     <Button
14         android:layout_width="wrap_content"
15         android:layout_height="wrap_content"
16         android:text="@string/bar"
17         style="@style/myBtn" />
18     <Button
19         android:layout_width="wrap_content"
20         android:layout_height="wrap_content"
21         android:text="@string/foobar"
22         style="@style/myBtn" />
23 </LinearLayout>
```

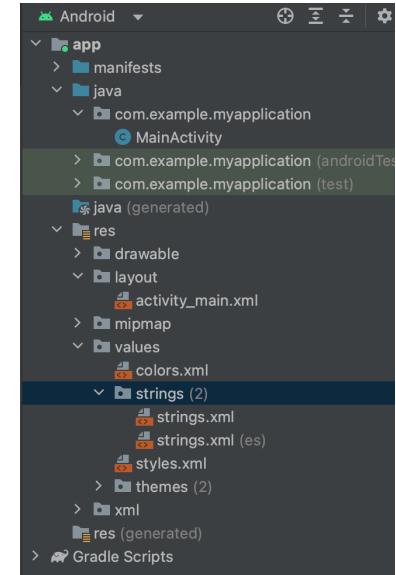
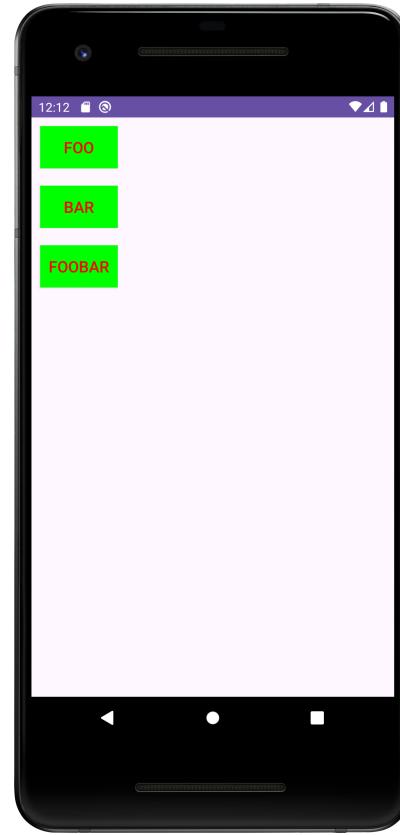


# STYLING

# ANDROID

```
1 <resources>
2     <string name="app_name">My Application</string>
3     <string name="foo">Hola</string>
4     <string name="bar">Cómo</string>
5     <string name="foobar">estás</string>
6 </resources>
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical">
7
8     <Button
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:text="@string/foo"
12        style="@style/myBtn" />
13
14     <Button
15        android:layout_width="wrap_content"
16        android:layout_height="wrap_content"
17        android:text="@string/bar"
18        style="@style/myBtn" />
19
20     <Button
21        android:layout_width="wrap_content"
22        android:layout_height="wrap_content"
23        android:text="@string/foobar"
24        style="@style/myBtn" />
25 </LinearLayout>
```

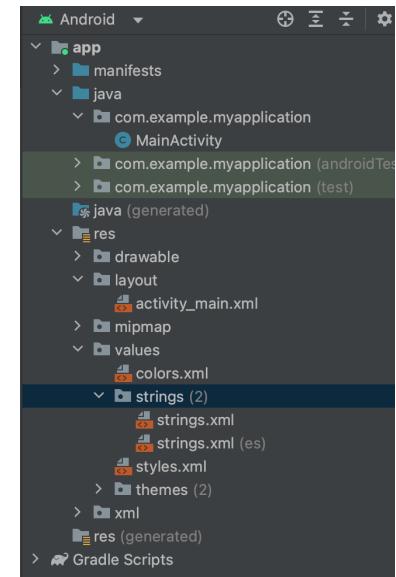
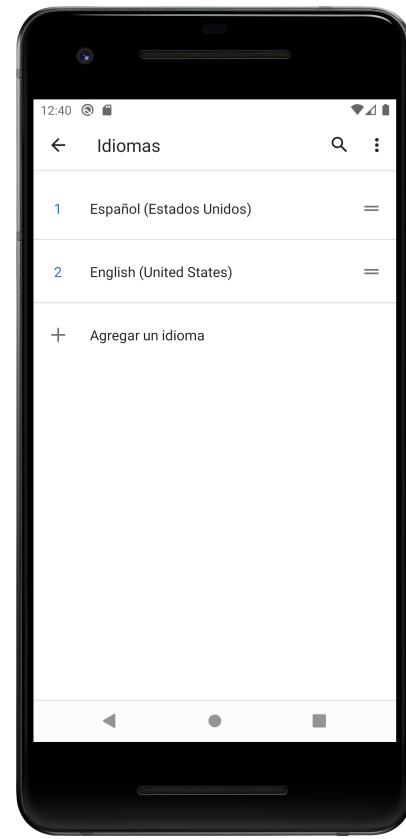


# STYLING

# ANDROID

```
1 <resources>
2     <string name="app_name">My Application</string>
3     <string name="foo">Hola</string>
4     <string name="bar">Cómo</string>
5     <string name="foobar">estás</string>
6 </resources>
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical">
7
8     <Button
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:text="@string/foo"
12        style="@style/myBtn" />
13     <Button
14         android:layout_width="wrap_content"
15         android:layout_height="wrap_content"
16         android:text="@string/bar"
17         style="@style/myBtn" />
18     <Button
19         android:layout_width="wrap_content"
20         android:layout_height="wrap_content"
21         android:text="@string/foobar"
22         style="@style/myBtn" />
23 </LinearLayout>
```

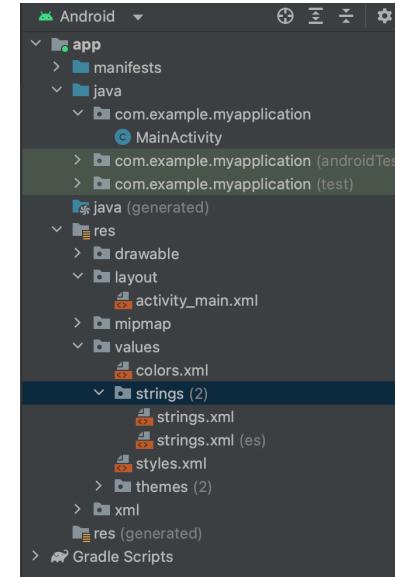
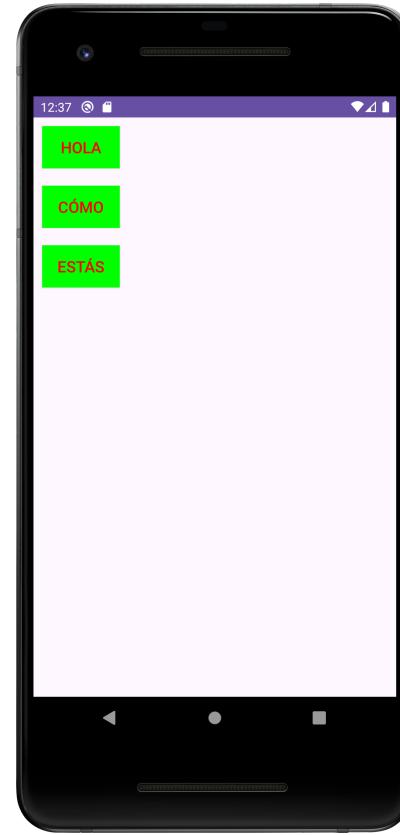


# STYLING

# ANDROID

```
1 <resources>
2     <string name="app_name">My Application</string>
3     <string name="foo">Hola</string>
4     <string name="bar">Cómo</string>
5     <string name="foobar">estás</string>
6 </resources>
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical">
7
8     <Button
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:text="@string/foo"
12        style="@style/myBtn" />
13     <Button
14         android:layout_width="wrap_content"
15         android:layout_height="wrap_content"
16         android:text="@string/bar"
17         style="@style/myBtn" />
18     <Button
19         android:layout_width="wrap_content"
20         android:layout_height="wrap_content"
21         android:text="@string/foobar"
22         style="@style/myBtn" />
23 </LinearLayout>
```

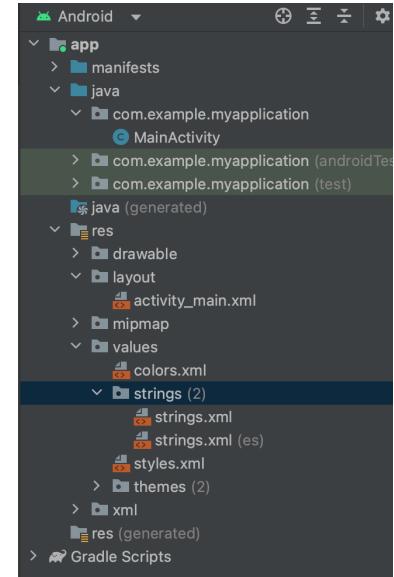
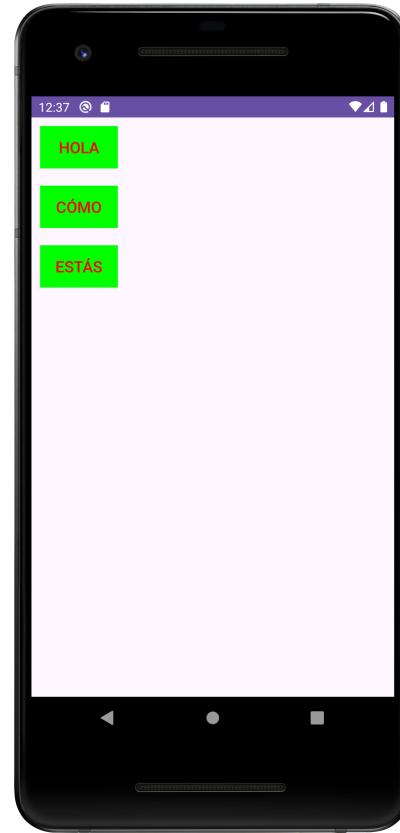


# STYLING

# ANDROID

```
1 <resources>
2     <string name="app_name">My Application</string>
3     <string name="foo">Hola</string>
4     <string name="bar">Cómo</string>
5     <string name="foobar">estás</string>
6 </resources>
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical">
7
8     <Button
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:text="@string/foo"
12        style="@style/myBtn" />
13     <Button
14         android:layout_width="wrap_content"
15         android:layout_height="wrap_content"
16         android:text="@string/bar"
17         style="@style/myBtn" />
18     <Button
19         android:layout_width="wrap_content"
20         android:layout_height="wrap_content"
21         android:text="@string/foobar"
22         style="@style/myBtn" />
23 </LinearLayout>
```



# DESKTOP GUI

.NET



# .NET (C#)

```
1 public partial class MainWindow : Window {
2     public MainWindow() {
3         InitializeComponent();
4
5         Button button = new Button();
6         button.Content = "Hello world";
7         button.HorizontalAlignment = HorizontalAlignment.Center;
8         button.VerticalAlignment = VerticalAlignment.Center;
9
10        Title = "Hello world";
11        Content = button;
12    }
13 }
```



# .NET (C#)

```
1 public partial class MainWindow : Window {
2     public MainWindow() {
3         InitializeComponent();
4
5         Button button = new Button();
6         button.Content = "Hello world";
7         button.HorizontalAlignment = HorizontalAlignment.Center;
8         button.VerticalAlignment = VerticalAlignment.Center;
9
10        Title = "Hello world";
11        Content = button;
12    }
13 }
```



# .NET (C#)

```
1 public partial class MainWindow : Window {
2     public MainWindow() {
3         InitializeComponent();
4
5         Button button = new Button();
6         button.Content = "Hello world";
7         button.HorizontalAlignment = HorizontalAlignment.Center;
8         button.VerticalAlignment = VerticalAlignment.Center;
9
10        Title = "Hello world";
11        Content = button;
12    }
13 }
```



# .NET (C#)

```
1 public partial class MainWindow : Window {
2     public MainWindow() {
3         InitializeComponent();
4
5         Button button = new Button();
6         button.Content = "Hello world";
7         button.HorizontalAlignment = HorizontalAlignment.Center;
8         button.VerticalAlignment = VerticalAlignment.Center;
9
10        Title = "Hello world";
11        Content = button;
12    }
13 }
```



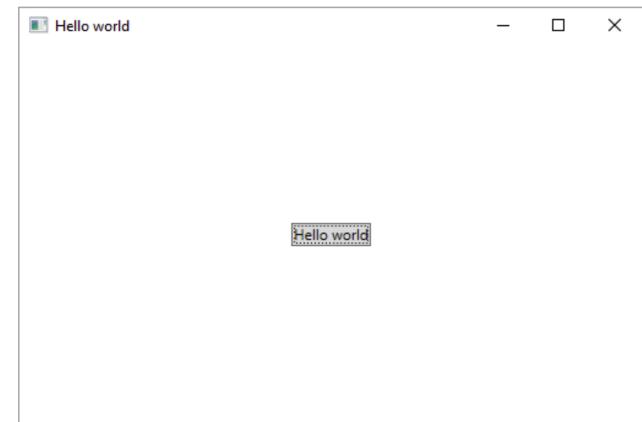
# .NET (C#)

```
1 public partial class MainWindow : Window {
2     public MainWindow() {
3         InitializeComponent();
4
5         Button button = new Button();
6         button.Content = "Hello world";
7         button.HorizontalAlignment = HorizontalAlignment.Center;
8         button.VerticalAlignment = VerticalAlignment.Center;
9
10        Title = "Hello world";
11        Content = button;
12    }
13 }
```



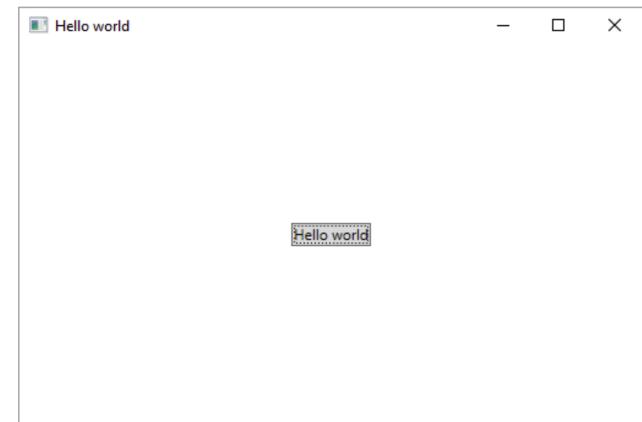
# .NET (C#)

```
1 public partial class MainWindow : Window {
2     public MainWindow() {
3         InitializeComponent();
4
5         Button button = new Button();
6         button.Content = "Hello world";
7         button.HorizontalAlignment = HorizontalAlignment.Center;
8         button.VerticalAlignment = VerticalAlignment.Center;
9
10        Title = "Hello world";
11        Content = button;
12    }
13 }
```



# GUI VIA X(A)ML

```
1 public partial class MainWindow : Window {
2     public MainWindow() {
3         InitializeComponent();
4
5         Button button = new Button();
6         button.Content = "Hello world";
7         button.HorizontalAlignment = HorizontalAlignment.Center;
8         button.VerticalAlignment = VerticalAlignment.Center;
9
10        Title = "Hello world";
11        Content = button;
12    }
13 }
```



# GUI VIA X(A)ML

```
1 public partial class MainWindow : Window
2 {
3     public MainWindow()
4     {
5         InitializeComponent();
6     }
7 }
```

```
1 <Window x:Class="HelloWorld.MainWindow"
2     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4     xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
5     xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
6     xmlns:local="clr-namespace:HelloWorld"
7     mc:Ignorable="d"
8     Title="Hello world">
9     <Button
10        HorizontalAlignment="Center"
11        VerticalAlignment="Center">
12            Hello world</Button>
13 </Window>
```

# GUI VIA X(A)ML

```
1 public partial class MainWindow : Window
2 {
3     public MainWindow()
4     {
5         InitializeComponent();
6     }
7 }
```

```
1 <Window x:Class="HelloWorld.MainWindow"
2     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4     xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
5     xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
6     xmlns:local="clr-namespace:HelloWorld"
7     mc:Ignorable="d"
8     Title="Hello world">
9     <Button
10        HorizontalAlignment="Center"
11        VerticalAlignment="Center">
12            Hello world</Button>
13 </Window>
```

# GUI VIA X(A)ML

```
1 public partial class MainWindow : Window
2 {
3     public MainWindow()
4     {
5         InitializeComponent();
6     }
7 }
```

```
1 <Window x:Class="HelloWorld.MainWindow"
2     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4     xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
5     xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
6     xmlns:local="clr-namespace:HelloWorld"
7     mc:Ignorable="d"
8     Title="Hello world">
9     <Button
10        HorizontalAlignment="Center"
11        VerticalAlignment="Center">
12            Hello world</Button>
13 </Window>
```

# GUI VIA X(A)ML

```
1 public partial class MainWindow : Window
2 {
3     public MainWindow()
4     {
5         InitializeComponent();
6     }
7 }
```

```
1 <Window x:Class="HelloWorld.MainWindow"
2     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4     xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
5     xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
6     xmlns:local="clr-namespace:HelloWorld"
7     mc:Ignorable="d"
8     Title="Hello world">
9     <Button
10        HorizontalAlignment="Center"
11        VerticalAlignment="Center">
12            Hello world</Button>
13 </Window>
```

# GUI VIA X(A)ML

```
1 public partial class MainWindow : Window
2 {
3     public MainWindow()
4     {
5         InitializeComponent();
6     }
7 }
```

```
1 <Window x:Class="HelloWorld.MainWindow"
2     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4     xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
5     xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
6     xmlns:local="clr-namespace:HelloWorld"
7     mc:Ignorable="d"
8     Title="Hello world">
9     <Button
10        HorizontalAlignment="Center"
11        VerticalAlignment="Center">
12            Hello world</Button>
13 </Window>
```

# GUI VIA X(A)ML

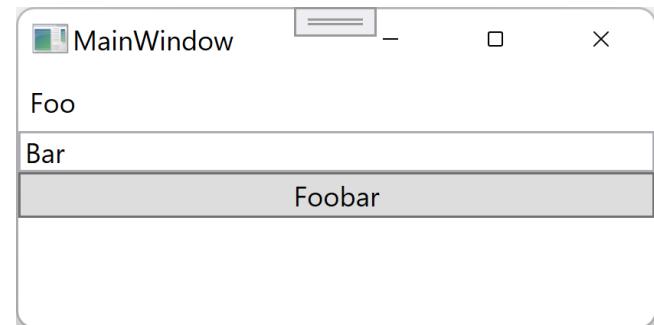
```
1 public partial class MainWindow : Window
2 {
3     public MainWindow()
4     {
5         InitializeComponent();
6     }
7 }
```

```
1 <Window x:Class="HelloWorld.MainWindow"
2     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4     xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
5     xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
6     xmlns:local="clr-namespace:HelloWorld"
7     mc:Ignorable="d"
8     Title="Hello world">
9     <Button
10        HorizontalAlignment="Center"
11        VerticalAlignment="Center">
12            Hello world</Button>
13 </Window>
```

# GUI ELEMENTS

.NET

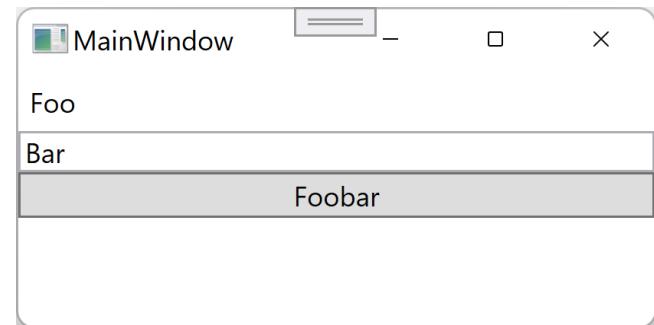
```
1 <Window x:Class="HelloWorld.MainWindow"
2     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presen...
3     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4     xmlns:d="http://schemas.microsoft.com/expression/blend/200...
5     xmlns:mc="http://schemas.openxmlformats.org/markup-compat...
6     xmlns:local="clr-namespace:HelloWorld"
7     mc:Ignorable="d"
8     Title="Hello world">
9     <StackPanel>
10        <Label>Foo</Label>
11        <TextBox>Bar</TextBox>
12        <Button>Foobar</Button>
13    </StackPanel>
14 </Window>
```



# GUI ELEMENTS

.NET

```
1 <Window x:Class="HelloWorld.MainWindow"
2     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presen...
3     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4     xmlns:d="http://schemas.microsoft.com/expression/blend/200...
5     xmlns:mc="http://schemas.openxmlformats.org/markup-compat...
6     xmlns:local="clr-namespace:HelloWorld"
7     mc:Ignorable="d"
8     Title="Hello world">
9     <StackPanel>
10        <Label>Foo</Label>
11        <TextBox>Bar</TextBox>
12        <Button>Foobar</Button>
13    </StackPanel>
14 </Window>
```



# GUI ELEMENTS

.NET

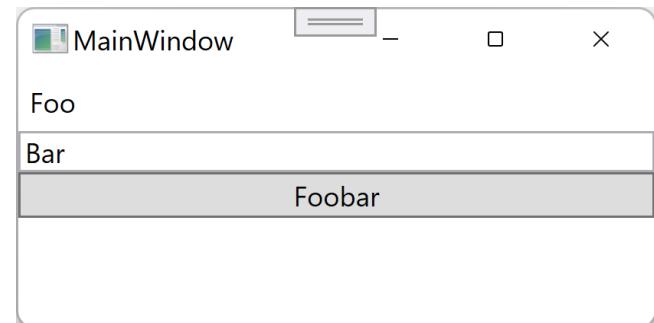
```
1 <Window x:Class="HelloWorld.MainWindow"
2     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presen...
3     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4     xmlns:d="http://schemas.microsoft.com/expression/blend/200...
5     xmlns:mc="http://schemas.openxmlformats.org/markup-compat...
6     xmlns:local="clr-namespace:HelloWorld"
7     mc:Ignorable="d"
8     Title="Hello world">
9     <StackPanel>
10        <Label>Foo</Label>
11        <TextBox>Bar</TextBox>
12        <Button>Foobar</Button>
13    </StackPanel>
14 </Window>
```



# GUI ELEMENTS

.NET

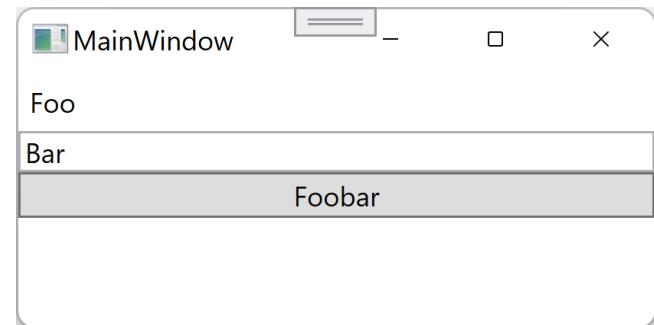
```
1 <Window x:Class="HelloWorld.MainWindow"
2     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presen...
3     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4     xmlns:d="http://schemas.microsoft.com/expression/blend/200...
5     xmlns:mc="http://schemas.openxmlformats.org/markup-compat...
6     xmlns:local="clr-namespace:HelloWorld"
7     mc:Ignorable="d"
8     Title="Hello world">
9     <StackPanel>
10        <Label>Foo</Label>
11        <TextBox>Bar</TextBox>
12        <Button>Foobar</Button>
13    </StackPanel>
14 </Window>
```



# GUI ELEMENTS

.NET

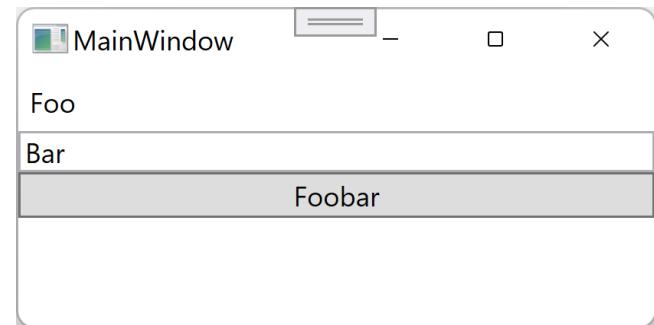
```
1 <Window x:Class="HelloWorld.MainWindow"
2     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presen...
3     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4     xmlns:d="http://schemas.microsoft.com/expression/blend/200...
5     xmlns:mc="http://schemas.openxmlformats.org/markup-compat...
6     xmlns:local="clr-namespace:HelloWorld"
7     mc:Ignorable="d"
8     Title="Hello world">
9     <StackPanel>
10        <Label>Foo</Label>
11        <TextBox>Bar</TextBox>
12        <Button>Foobar</Button>
13    </StackPanel>
14 </Window>
```



# GUI ELEMENTS

.NET

```
1 <Window x:Class="HelloWorld.MainWindow"
2     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presen...
3     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4     xmlns:d="http://schemas.microsoft.com/expression/blend/200...
5     xmlns:mc="http://schemas.openxmlformats.org/markup-compat...
6     xmlns:local="clr-namespace:HelloWorld"
7     mc:Ignorable="d"
8     Title="Hello world">
9     <StackPanel>
10        <Label>Foo</Label>
11        <TextBox>Bar</TextBox>
12        <Button>Foobar</Button>
13    </StackPanel>
14 </Window>
```



# STYLING

## .NET

```
1 <StackPanel>
2   <Button Content="Foo"
3     Background="Azure"
4     Foreground="#FFFFFF"
5     FontSize="18"
6     Margin="10" />
7   <Button Content="Bar" />
8   <Button Content="Foobar" />
9 </StackPanel>
```

# STYLING

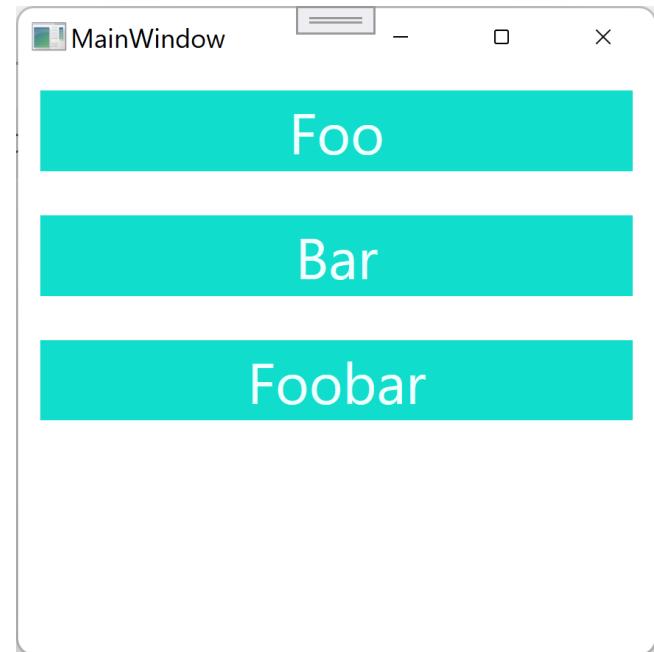
## .NET

```
1 <StackPanel>
2   <Button Content="Foo"
3     Background="Azure"
4     Foreground="#FFFFFF"
5     FontSize="18"
6     Margin="10" />
7   <Button Content="Bar" />
8   <Button Content="Foobar" />
9 </StackPanel>
```

# STYLING

## .NET

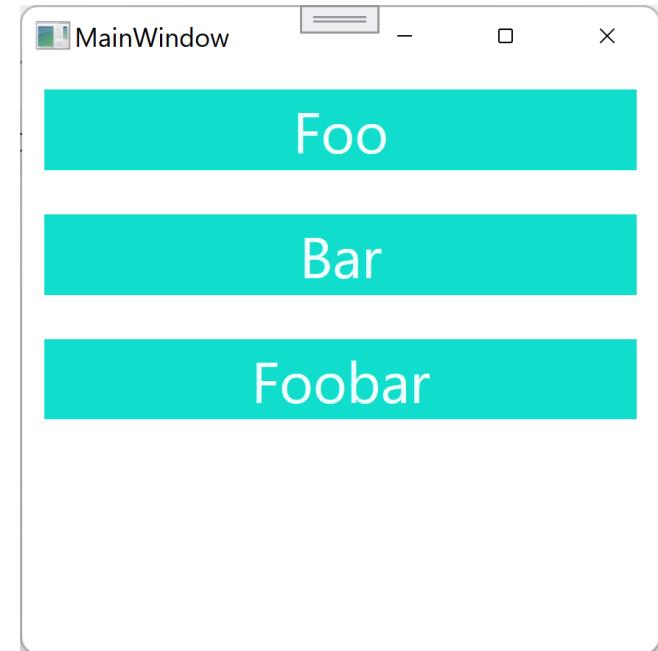
```
1 <Window x:Class="WpfApp1.MainWindow"
2     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4     xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
5     xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
6     mc:Ignorable="d"
7     Title="MainWindow" Height="300" Width="300">
8     <Window.Resources>
9         <Style x:Key="MyButtonStyle" TargetType="Button">
10            <Setter Property="Background" Value="Azure" />
11            <Setter Property="Foreground" Value="#FFFFFF" />
12            <Setter Property="FontSize" Value="18" />
13            <Setter Property="Margin" Value="10" />
14        </Style>
15     </Window.Resources>
16     <StackPanel>
17         <Button Style="{StaticResource MyButtonStyle}">Foo</Button>
18         <Button Style="{StaticResource MyButtonStyle}">Bar</Button>
19         <Button Style="{StaticResource MyButtonStyle}">Foobar</Button>
20     </StackPanel>
21 </Window>
```



# STYLING

## .NET

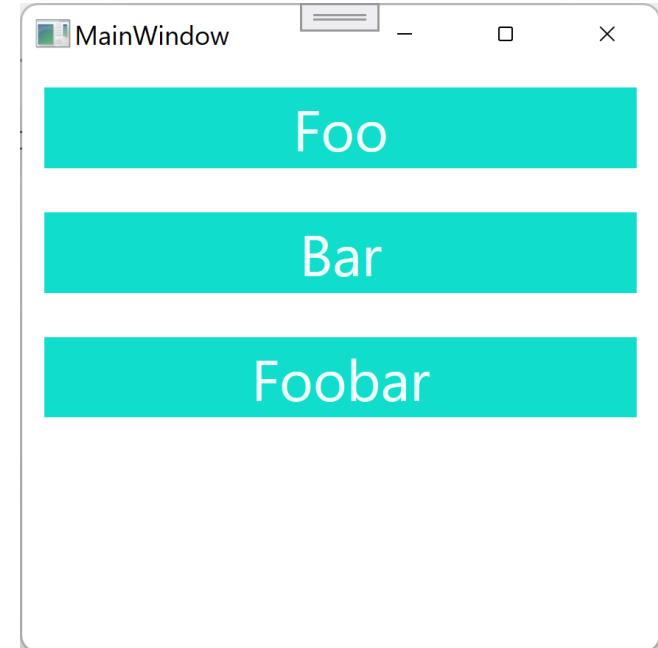
```
1  <Window x:Class="WpfApp1.MainWindow"
2      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
5      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
6      mc:Ignorable="d"
7      Title="MainWindow" Height="300" Width="300">
8  <Window.Resources>
9      <Style x:Key="MyButtonStyle" TargetType="Button">
10         <Setter Property="Background" Value="Azure" />
11         <Setter Property="Foreground" Value="#FFFFFF" />
12         <Setter Property="FontSize" Value="18" />
13         <Setter Property="Margin" Value="10" />
14     </Style>
15 </Window.Resources>
16 <StackPanel>
17     <Button Style="{StaticResource MyButtonStyle}">Foo</Button>
18     <Button Style="{StaticResource MyButtonStyle}">Bar</Button>
19     <Button Style="{StaticResource MyButtonStyle}">Foobar</Button>
20 </StackPanel>
21 </Window>
```



# STYLING

## .NET

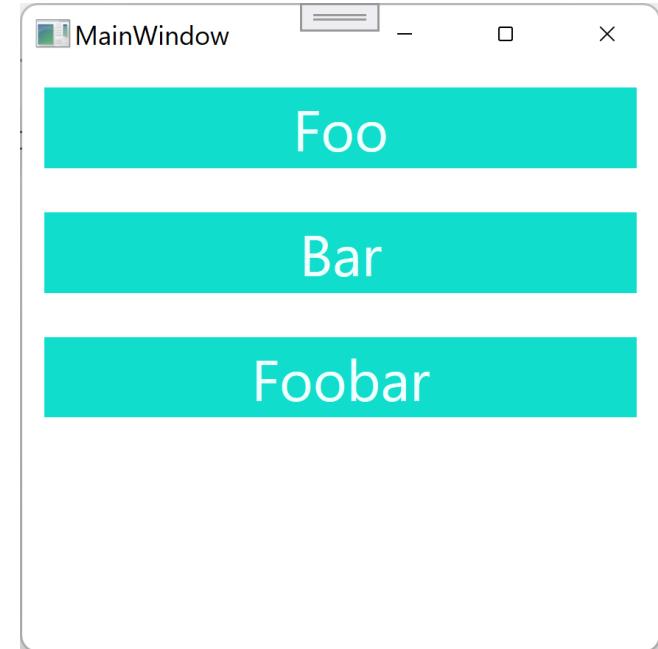
```
1  <Window x:Class="WpfApp1.MainWindow"
2      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
5      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
6      mc:Ignorable="d"
7      Title="MainWindow" Height="300" Width="300">
8  <Window.Resources>
9      <Style x:Key="MyButtonStyle" TargetType="Button">
10         <Setter Property="Background" Value="Azure" />
11         <Setter Property="Foreground" Value="#FFFFFF" />
12         <Setter Property="FontSize" Value="18" />
13         <Setter Property="Margin" Value="10" />
14     </Style>
15 </Window.Resources>
16 <StackPanel>
17     <Button Style="{StaticResource MyButtonStyle}">Foo</Button>
18     <Button Style="{StaticResource MyButtonStyle}">Bar</Button>
19     <Button Style="{StaticResource MyButtonStyle}">Foobar</Button>
20 </StackPanel>
21 </Window>
```



# STYLING

## .NET

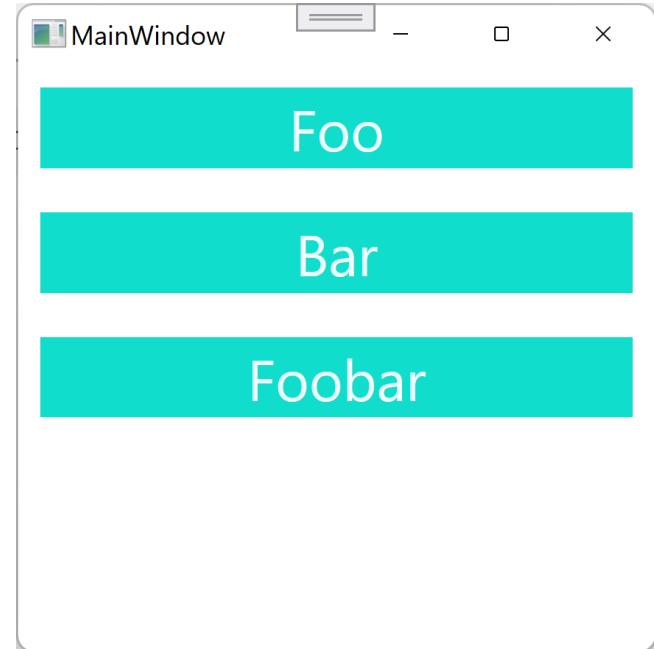
```
1  <Window x:Class="WpfApp1.MainWindow"
2      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
5      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
6      mc:Ignorable="d"
7      Title="MainWindow" Height="300" Width="300">
8  <Window.Resources>
9      <Style x:Key="MyButtonStyle" TargetType="Button">
10         <Setter Property="Background" Value="Azure" />
11         <Setter Property="Foreground" Value="#FFFFFF" />
12         <Setter Property="FontSize" Value="18" />
13         <Setter Property="Margin" Value="10" />
14     </Style>
15 </Window.Resources>
16 <StackPanel>
17     <Button Style="{StaticResource MyButtonStyle}">Foo</Button>
18     <Button Style="{StaticResource MyButtonStyle}">Bar</Button>
19     <Button Style="{StaticResource MyButtonStyle}">Foobar</Button>
20 </StackPanel>
21 </Window>
```



# STYLING

## .NET

```
1  <Window x:Class="WpfApp1.MainWindow"
2      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
5      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
6      mc:Ignorable="d"
7      Title="MainWindow" Height="300" Width="300">
8  <Window.Resources>
9      <Style x:Key="MyButtonStyle" TargetType="Button">
10         <Setter Property="Background" Value="Azure" />
11         <Setter Property="Foreground" Value="#FFFFFF" />
12         <Setter Property="FontSize" Value="18" />
13         <Setter Property="Margin" Value="10" />
14     </Style>
15  </Window.Resources>
16  <StackPanel>
17      <Button Style="{StaticResource MyButtonStyle}">Foo</Button>
18      <Button Style="{StaticResource MyButtonStyle}">Bar</Button>
19      <Button Style="{StaticResource MyButtonStyle}">Foobar</Button>
20  </StackPanel>
21 </Window>
```



# **LISTENER DESIGN PATTERN**

# EVENT-DRIVEN PROGRAMMING

## KEY CONCEPTS

- Event: raised when something has happened
- Event Handler: code that handles the raised event

# EVENT-DRIVEN PROGRAMMING

## WEB DEVELOPMENT (JAVASCRIPT)

```
1 <html>
2   <head></head>
3   <body>
4     <button id="myButton"
5           onclick="alert('Button Clicked!')" >Click me</button>
6   </body>
7 </html>
```

# EVENT-DRIVEN PROGRAMMING

## WEB DEVELOPMENT (JAVASCRIPT)

```
1 <html>
2   <head></head>
3   <body>
4     <button id="myButton"
5           onclick="alert('Button Clicked!')" >Click me</button>
6   </body>
7 </html>
```

# EVENT-DRIVEN PROGRAMMING

## WEB DEVELOPMENT (JAVASCRIPT)

```
1 <html>
2   <head></head>
3   <body>
4     <button id="myButton">Click me</button>
5     <script type="text/javascript">
6       const btn = document.getElementById('myButton');
7       btn.addEventListener('click', function() {
8         alert('Button Clicked!');
9       });
10      </script>
11    </body>
12 </html>
```

# EVENT-DRIVEN PROGRAMMING

## WEB DEVELOPMENT (JAVASCRIPT)

```
1 <html>
2   <head></head>
3   <body>
4     <button id="myButton">Click me</button>
5     <script type="text/javascript">
6       const btn = document.getElementById('myButton');
7       btn.addEventListener('click', function() {
8         alert('Button Clicked!');
9       });
10      </script>
11    </body>
12 </html>
```

# EVENT-DRIVEN PROGRAMMING

## WEB DEVELOPMENT (JAVASCRIPT)

```
1 <html>
2   <head></head>
3   <body>
4     <button id="myButton">Click me</button>
5     <script type="text/javascript">
6       const btn = document.getElementById('myButton');
7       btn.addEventListener('click', function() {
8         alert('Button Clicked!');
9       });
10      </script>
11    </body>
12 </html>
```

# EVENT-DRIVEN PROGRAMMING

## WEB DEVELOPMENT (JAVASCRIPT)

```
1 <html>
2   <head></head>
3   <body>
4     <button id="myButton">Click me</button>
5     <script type="text/javascript">
6       const btn = document.getElementById('myButton');
7       btn.addEventListener('click', function() {
8         alert('Button Clicked!');
9       });
10      </script>
11    </body>
12 </html>
```

# EVENT-DRIVEN PROGRAMMING

## WEB DEVELOPMENT (JAVASCRIPT)

```
1 <html>
2   <head></head>
3   <body>
4     <button id="myButton">Click me</button>
5     <script type="text/javascript">
6       const btn = document.getElementById('myButton');
7       btn.addEventListener('click', function() {
8         alert('Button Clicked!');
9       });
10      </script>
11    </body>
12 </html>
```

# EVENT-DRIVEN PROGRAMMING

## WEB DEVELOPMENT (JAVASCRIPT)

```
1 <html>
2   <head></head>
3   <body>
4     <button id="myButton">Click me</button>
5     <script type="text/javascript">
6       const btn = document.getElementById('myButton');
7       btn.addEventListener('click', function() {
8         alert('Button Clicked!');
9       });
10      </script>
11    </body>
12 </html>
```

# EVENT-DRIVEN PROGRAMMING

## WEB DEVELOPMENT (JAVASCRIPT)

```
1 <html>
2   <head></head>
3   <body>
4     <button id="myButton">Click me</button>
5     <script type="text/javascript">
6       const btn = document.getElementById('myButton');
7       btn.onclick = function() {
8         alert('Button Clicked!');
9       };
10      </script>
11    </body>
12 </html>
```

# EVENT-DRIVEN PROGRAMMING

## ANDROID

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical" >
8     <Button
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:text="Foobar" />
12 </LinearLayout>
```

```
1 Button myButton = findViewById(R.id.myButton);
2 myButton.setOnClickListener(new View.OnClickListener() {
3     @Override
4     public void onClick(View v) {
5         Toast.makeText(MainActivity.this,
6                 "Button Clicked!",
7                 Toast.LENGTH_SHORT).show();
8     }
9 });
```

# EVENT-DRIVEN PROGRAMMING

## ANDROID

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical" >
8     <Button
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:text="Foobar" />
12 </LinearLayout>
```

```
1 Button myButton = findViewById(R.id.myButton);
2 myButton.setOnClickListener(new View.OnClickListener() {
3     @Override
4     public void onClick(View v) {
5         Toast.makeText(MainActivity.this,
6                 "Button Clicked!",
7                 Toast.LENGTH_SHORT).show();
8     }
9 });
```

# EVENT-DRIVEN PROGRAMMING

## ANDROID

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical" >
8     <Button
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:text="Foobar" />
12 </LinearLayout>
```

```
1 Button myButton = findViewById(R.id.myButton);
2 myButton.setOnClickListener(new View.OnClickListener() {
3     @Override
4     public void onClick(View v) {
5         Toast.makeText(MainActivity.this,
6                 "Button Clicked!",
7                 Toast.LENGTH_SHORT).show();
8     }
9 });
```

# EVENT-DRIVEN PROGRAMMING

## ANDROID

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical" >
8     <Button
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:text="Foobar" />
12 </LinearLayout>
```

```
1 Button myButton = findViewById(R.id.myButton);
2 myButton.setOnClickListener(new View.OnClickListener() {
3     @Override
4     public void onClick(View v) {
5         Toast.makeText(MainActivity.this,
6                 "Button Clicked!",
7                 Toast.LENGTH_SHORT).show();
8     }
9 });
```

# EVENT-DRIVEN PROGRAMMING

## ANDROID

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical" >
8     <Button
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:text="Foobar" />
12 </LinearLayout>
```

```
1 Button myButton = findViewById(R.id.myButton);
2 myButton.setOnClickListener(new View.OnClickListener() {
3     @Override
4     public void onClick(View v) {
5         Toast.makeText(MainActivity.this,
6                 "Button Clicked!",
7                 Toast.LENGTH_SHORT).show();
8     }
9 });
```

# EVENT-DRIVEN PROGRAMMING

## ANDROID

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical" >
8     <Button
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:text="Foobar" />
12 </LinearLayout>
```

```
1 Button myButton = findViewById(R.id.myButton);
2 myButton.setOnClickListener(new View.OnClickListener() {
3     @Override
4     public void onClick(View v) {
5         Toast.makeText(MainActivity.this,
6                 "Button Clicked!",
7                 Toast.LENGTH_SHORT).show();
8     }
9 });
```

# EVENT-DRIVEN PROGRAMMING

## ANDROID

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical" >
8     <Button
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:text="Foobar" />
12 </LinearLayout>
```

```
1 Button myButton = findViewById(R.id.myButton);
2 myButton.setOnClickListener(new View.OnClickListener() {
3     @Override
4     public void onClick(View v) {
5         Toast.makeText(MainActivity.this,
6                 "Button Clicked!",
7                 Toast.LENGTH_SHORT).show();
8     }
9 });
```

# EVENT-DRIVEN PROGRAMMING

## ANDROID

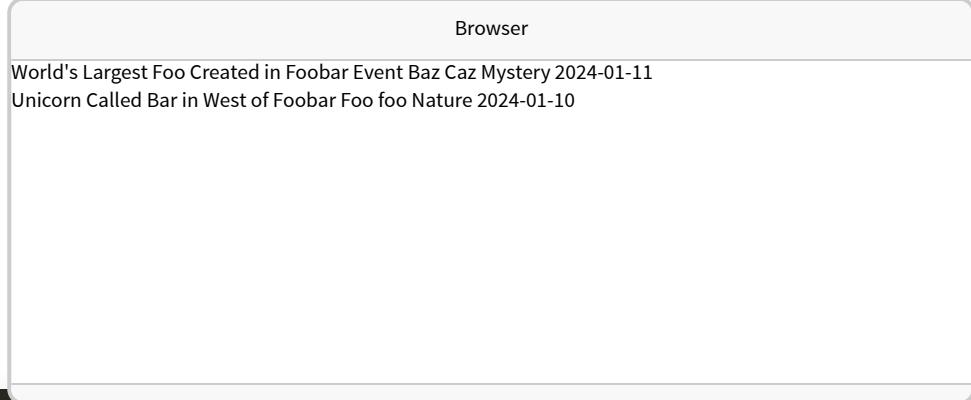
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical" >
8     <Button
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:text="Foobar" />
12 </LinearLayout>
```

```
1 Button myButton = findViewById(R.id.myButton);
2 myButton.setOnClickListener(new View.OnClickListener() {
3     @Override
4     public void onClick(View v) {
5         Toast.makeText(MainActivity.this,
6                 "Button Clicked!",
7                 Toast.LENGTH_SHORT).show();
8     }
9 });
```

# NEWS APP

## WEB

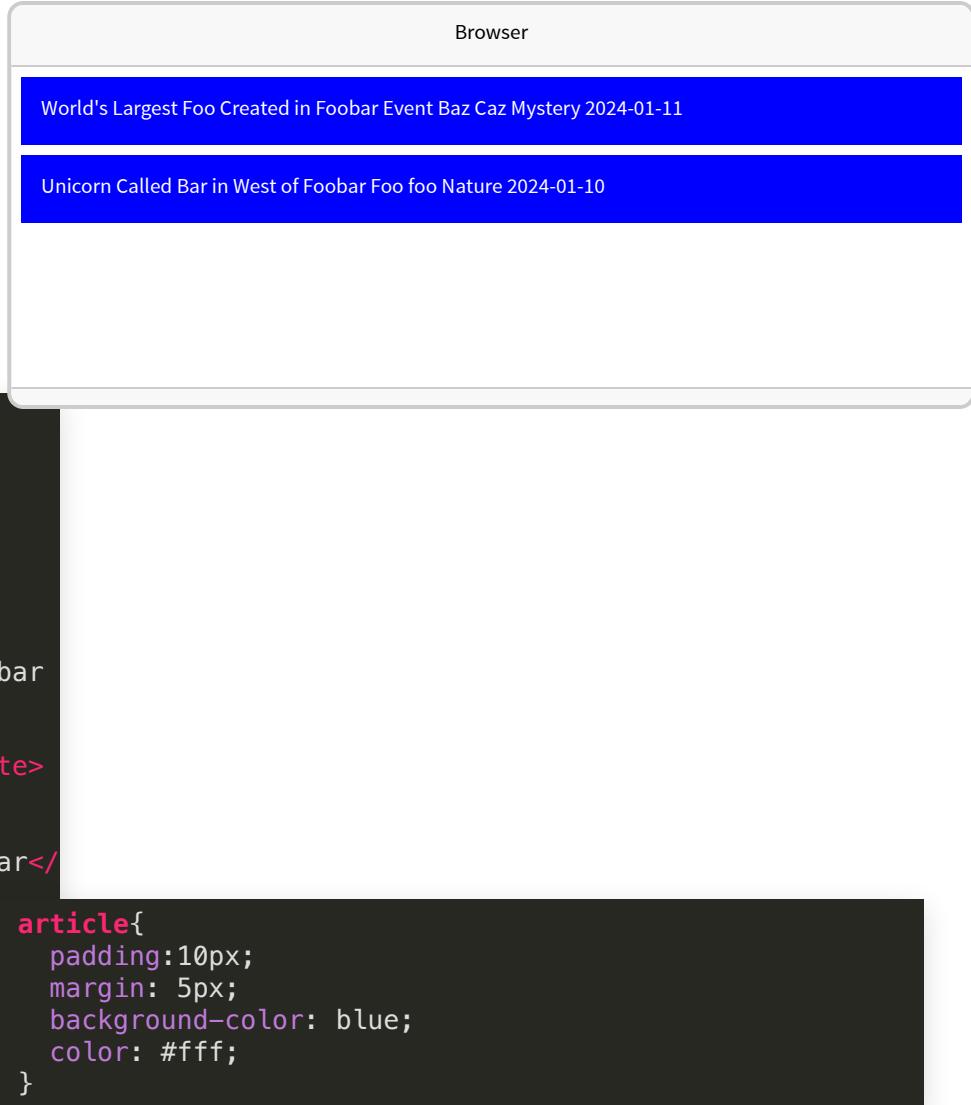
```
1 <!DOCTYPE html>
2 <html>
3   <head></head>
4   <body>
5     <news_articles>
6       <article>
7         <headline>World's Largest Foo Created in Foobar
8         <author>Baz Caz</author>
9         <category>Mystery</category>
10        <publication_date>2024-01-11</publication_date>
11      </article>
12      <article>
13        <headline>Unicorn Called Bar in West of Foobar</
14        <author>Foo foo</author>
15        <category>Nature</category>
16        <publication_date>2024-01-10</publication_date>
17      </article>
18    </news_articles>
19  </body>
20 </html>
```



# NEWS APP

## WEB

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <link href="index.css" rel="stylesheet">
5   </head>
6   <body>
7     <news_articles>
8       <article>
9         <headline>World's Largest Foo Created in Foobar
10        <author>Baz Caz</author>
11        <category>Mystery</category>
12        <publication_date>2024-01-11</publication_date>
13      </article>
14      <article>
15        <headline>Unicorn Called Bar in West of Foobar</
16        <author>Foo foo</author>
17        <category>Nature</category>
18        <publication_date>2024-01-10</publication_date>
19      </article>
20    </news_articles>
21  </body>
22 </html>
```



# NEWS APP

## WEB

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <link href="index.css" rel="stylesheet">
5   </head>
6   <body>
7     <header>
8       <h1>Foo news</h1>
9     </header>
10    <news_articles>
11      <article>
12        <headline>World's Largest Foo Created in F...
13        <author>Baz Caz</author>
14        <category>Mystery</category>
15        <publication_date>2024-01-11</publication_...
16      </article>
17      <article>
18        <headline>Unicorn Called Bar in West of Fo...
19        <author>Foo foo</author>
20        <category>Nature</category>
21        <publication_date>2024-01-10</publication_...
22      </article>
23    </news_articles>
24  </body>
25 </html>
```



```
1 article{
2   padding:10px;
3   margin: 5px;
4   background-color: blue;
5   color: #fff;
6 }
7 #headline{
8   text-align:center;
9   text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);
10  background: linear-gradient(to right, #e74c3c, ...
11 }
```

# NEWS APP

## WEB

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <link href="index.css" rel="stylesheet">
5   </head>
6   <body>
7     <header>
8       <h1>Foo news</h1>
9     </header>
10    <news_articles>
11      <article>
12        <headline>World's Largest Foo Created in F
13        <author>Baz Caz</author>
14        <category>Mystery</category>
15        <publication_date>2024-01-11</publication_
16        
17      </article>
18      <article>
19        <headline>Unicorn Called Bar in West of Fo
20        <author>Grace Lewis</author>
21        <category>Nature</category>
22        <publication_date>2024-01-10</publication_
23        
24      </article>
25    </news_articles>
26  </body>
```



```
1 article{
2   padding:10px;
3   margin: 5px;
4   background-color: blue;
5   color: #fff;
6 }
7 #headline{
8   text-align:center;
9   text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);
10  background: linear-gradient(to right, #e74c3c, ;
11 }
12 article img{
13   float:right
14 }
```

# NEWS APP

## WEB

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <link href="index.css" rel="stylesheet">
5   </head>
6   <body>
7     <header>
8       <h1>Foo news</h1>
9     </header>
10    <news_articles>
11      <a href="article.html?id=1">
12        <article>
13          <headline>World's Largest Foo Created in</headline>
14          <author>Baz Caz</author>
15          <category>Mystery</category>
16          <publication_date>2024-01-11</publication_date>
17          
18        </article>
19      </a>
20      <a href="article.html?id=2">
21        <article>
22          <headline>Unicorn Called Bar in West of</headline>
23          <author>Grace Lewis</author>
24          <category>Nature</category>
25          <publication_date>2024-01-10</publication_date>
26          
```



```
1 article{
2   padding:10px;
3   margin: 5px;
4   background-color: blue;
5   color: #fff;
6 }
7 #headline{
8   text-align:center;
9   text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);
10  background: linear-gradient(to right, #e74c3c, #ff9900);
11 }
12 article img{
13   float:right
14 }
15 a article:hover{
16   background-color: yellow !important;
17   border: 2px solid #ff9900;
18   box-shadow: 0 0 10px rgba(0, 0, 0, 0.5);
19 }
```

# MODERN FRAMEWORKS

- The problem: UIs get more complicated
- Very hard to develop complicated UIs this way
- Solution: Web UI frameworks
  - For example: React, Vue, Angular

# MODERN FRAMEWORKS

## REACT

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <div id="root"></div>
5     <script>
6       const h = document.createElement('h1')
7       h.textContent = 'Hello'
8       h.className = 'foo'
9
10      document.getElementById('root').appendChild(h)
11    </script>
12  </body>
13 </html>
```

# MODERN FRAMEWORKS

## REACT

```
1 <html>
2   <body>
3     <div id="root"></div>
4     <script src="react.development.js"></script>
5     <script src="react-dom.development.js"></script>
6     <script>
7       //const h = document.createElement('h1')
8       const h = React.createElement('h1')
9       h.textContent = 'Hello'
10      h.className = 'foo'
11
12      //document.getElementById('root').appendChild(h)
13      ReactDOM.render(h, document.getElementById('root'))
14    </script>
15  </body>
16 </html>
```

# MODERN FRAMEWORKS

## REACT

```
1 <html>
2   <body>
3     <div id="root"></div>
4     <script src="react.development.js"></script>
5     <script src="react-dom.development.js"></script>
6     <script>
7       const h = React.createElement('h1', {
8         className : 'foo'
9       }, 'Hello')
10      ReactDOM.render(h, document.getElementById('root'))
11    </script>
12  </body>
13</html>
```

# MODERN FRAMEWORKS

## REACT

```
1 <html>
2   <body>
3     <div id="root"></div>
4     <script src="react.development.js"></script>
5     <script src="react-dom.development.js"></script>
6     <script src="babel.js"></script>
7
8     <script type="text/babel">
9       const h = <h1>Hello {1+2}
10      <div>
11        <span className="foo" foo="bar">Hi</span>
12      </div>
13    </h1>
14
15    ReactDOM.render(h, document.getElementById('root'))
16  </script>
17  <body>
18 </html>
```

# TRANSPILATION

- JavaScript evolves, adding new functionality
- This may lead to correct code according to the new ECMAScript standard, but older browsers may not support it yet
- Transpilation helps by converting new syntax into old syntax, supporting old browsers
  - Babel is an example of a transpiler

# REACT

```
Command Prompt  
D:\>npx create-react-app hello_world  
npx: installed 67 in 3.262s  
Creating a new React app in D:\hello_world.  
Installing packages. This might take a couple of minutes.  
Installing react, react-dom, and react-scripts with cra-template...
```

# REACT

```
Success! Created hello_world at D:\hello_world
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd hello_world
  npm start
```

# REACT

The screenshot shows a development environment for a React application. On the left, the Explorer sidebar displays the project structure:

- HELLO\_WORLD
- node\_modules
- public
- src
  - App.js
  - index.js
- package-lock.json
- package.json

Two code editors are open in the center:

- index.html**: Contains the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>React App</title>
  </head>
  <body>
    <div id="root"></div>
  </body>
</html>
```

- App.js**: Contains the following code:

```
function App() {
  return (
    <div>
      | Hello world
    </div>
  );
}

export default App;
```

On the right, a browser window titled "React App" shows the output: "Hello world". The browser's developer tools are open, showing the Elements tab with the DOM structure:

```
<!DOCTYPE html>
<html lang="en">
  <head></head>
  <body> == $0
    <div id="root">
      <div>Hello world</div>
    </div>
    <script src="/static/js/bundle.js"></script>
    <script src="/static/js/vendors~main.chunk.js"></script>
    <script src="/static/js/main.chunk.js"></script>
  </body>
</html>
```

The developer tools also show the Styles panel with CSS rules for the body element.

At the bottom, the status bar shows: master\*, 0 0 △ 0, Ln 12, Col 1, Spaces: 2, UTF-8, LF, HTML, html, body.

# REACT

The screenshot shows a development environment for a React application named "HELLO\_WORLD".

**EXPLORER:** Shows the project structure with files: index.html, App.js, Foo.js, index.js, package-lock.json, and package.json.

**index.html:** Contains the basic HTML structure for a React app.

**Foo.js:** A simple functional component that returns a single 

element containing three nested 

elements with attributes a, b, and c.

```
src > JS Foo.js > Foo
1 function Foo({a,b,c}) {
2   return (
3     <div>
4       | a: {a}, b: {b}, c: {c}
5     </div>
6   );
7 }
8
9 export default Foo;
```

**index.js:** The main entry point of the application, rendering the  component.

```
src > JS index.js M
src > JS index.js
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import App from './App';
4
5 ReactDOM.render(
6   <React.StrictMode>
7     | <App />
8   </React.StrictMode>,
9   document.getElementById('root')
10 );
```

**App.js:** The component that contains the logic for rendering multiple  components.

```
src > JS App.js > ...
1 import Foo from './Foo';
2
3 function App() {
4   var i = 1+2;
5   return (
6     <div>
7       | <Foo a="1" b="a" c={i} />
8       | <Foo a="2" b="b" c={i+1} />
9       | <Foo a="3" b="c" c={i+i} />
10    </div>
11  );
12 }
13
14 export default App;
15 |
```

**Terminal:** Shows the command "npm start" being run, which outputs the application's output to the console.

```
a: 1, b: a, c: 3
a: 2, b: b, c: 4
a: 3, b: c, c: 6
```

**Browser Preview:** Shows the rendered output at "localhost:3001".

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body> == $0
    <div id="root">
      <div>
        <div>
          "a: "
          "1"
          ", b: "
          "a"
          ", c: "
          "3"
        </div>
      </div>...</div>
    </div>
<script src="/static/js/bundle.js"></script>
<script src="/static/js/vendors~main.chunk.js"></script>
<script src="/static/js/main.chunk.js"></script>
</body>
</html>
```

# REACT

The screenshot shows a dark-themed code editor interface with multiple tabs and panels. On the left, the Explorer panel displays the project structure:

- HELLO\_WORLD
- node\_modules
- public
- index.html
- src
  - App.js
  - Foo.js
  - index.js
- package-lock.json
- package.json

The main workspace contains several code files:

- index.html**:

```
public > index.html > ...
1  <html lang="en">
2    <head>
3      <meta charset="utf-8" />
4      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
5      <title>React App</title>
6    </head>
7    <body>
8      <div id="root"></div>
9    </body>
10   </html>
```
- Foo.js**:

```
src > Foo.js > Foo
1  import { useState } from 'react';
2
3  function Foo() {
4    const [bar, setBar] = useState(0);
5
6    console.log(bar);
7
8    return (
9      <div onClick={() => setBar(bar + 1)}>
10        {bar}
11      </div>
12    );
13  }
14
15 export default Foo;
```
- index.js** (marked with a 'M' icon):

```
src > index.js
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3  import App from './App';
4
5  ReactDOM.render(
6    <React.StrictMode>
7      <App />
8    </React.StrictMode>,
9    document.getElementById('root')
10 );
```
- App.js**:

```
src > App.js > App
1  import Foo from "./Foo";
2
3  function App() {
4    return (
5      <div>
6        <Foo />
7        <Foo />
8        <Foo />
9      </div>
10    );
11  }
12
13 export default App;
```

On the right side, there is a navigation bar labeled "React App" with icons for back, forward, and refresh, and a status bar at the bottom showing "HMR Waiting" with counts (3 0, 2 1, 2 2, 1 3, 2 2).