

# Accessibility in Action

Technical Solutions to the Accessibility Challenge



 @KellyShuster

# Schedule

What is Accessibility?

Android vs iOS

Exploring Android Accessibility Features

Coding Accessible Apps





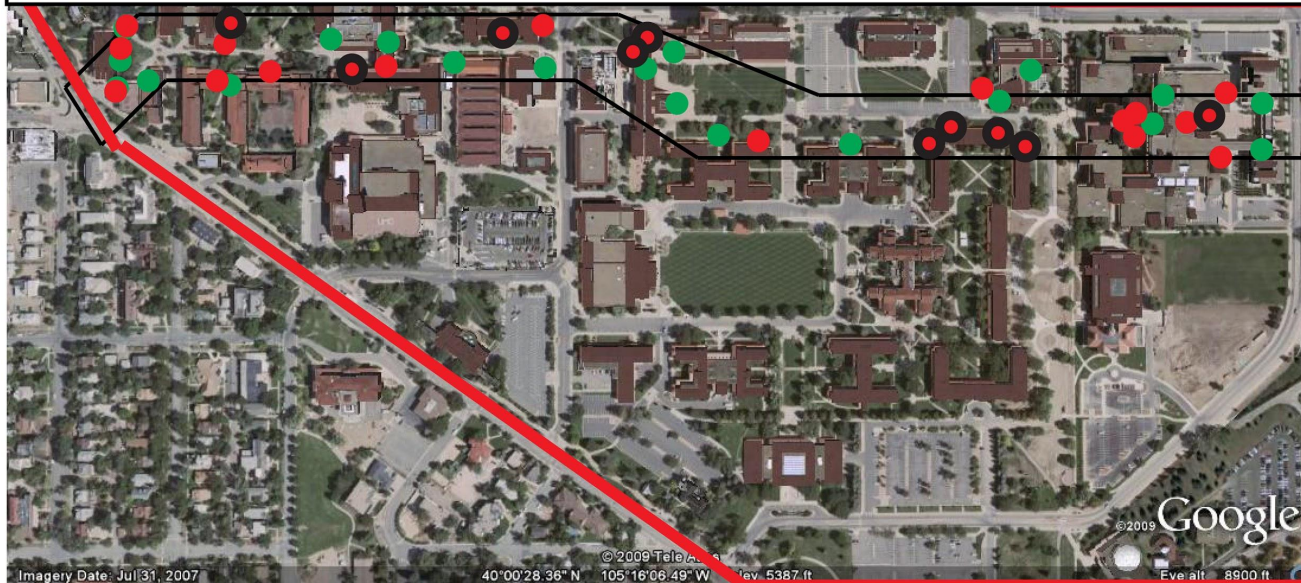
Entryways open to the path: 46



● Are accessible: 20

● Not accessible by wheelchair: 26

● Due to a single set of stairs: 10







**What is Accessibility?**

Mobility Impairment

Audio Impairment

Vision Impairment



# Mobility Impairment

**Dexterity  
Issues**

**Complete  
Paralysis**



# Mobility Impairment

**Dexterity  
Issues**

**Complete  
Paralysis**

**48 dp Rhythm**

**Custom touch  
& hold delay**

**Head switch**

**Sip & puff**

**EMG sensors**


# Audio Impairment



# Audio Impairment

**Hearing  
Loss**

**Complete  
Deafness**



**Audio notifications must  
have visual component**

**System wide closed captioning**

# Visual Impairment

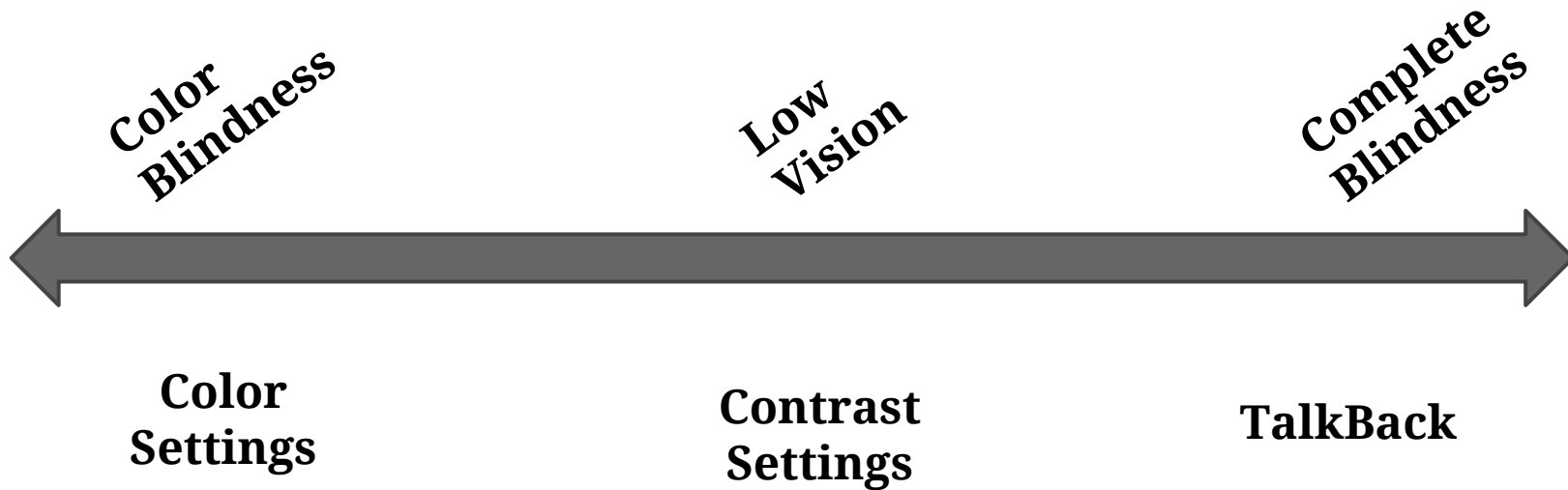
**Color  
Blindness**

**Low  
Vision**

**Complete  
Blindness**



# Visual Impairment



**Who are the Users?**



Physical Challenges

35.2 million

Hearing Challenges

37.6 million

Vision Challenges

20.6 million

**Accessible UI == Good UI**

**Google**

A photograph of three men on a dirt trail. The man on the left is smiling and looking back over his shoulder. The man in the center is seen from behind. The man on the right is wearing a prosthetic leg and is also smiling. They are all on bicycles. A semi-transparent white box with a thin white border is centered over the image, containing the title and subtitle.

# IMPACT CHALLENGE

An open invitation to aim our collective  
might at creating real, impactful change.



# ACCESSIBILITY







# **Android vs iOS**





TalkBack (Donut)



VoiceOver (iOS 3)



Poor Documentation  
Smaller Community



Good Documentation  
Larger Community



No Screen Blanking

No Quick On/Off



Screen blanking

Three Tap On/Off



**OPEN!**



**CLOSED!**

# **Exploring Android Accessibility Settings**



# **How to Use TalkBack**



# **TalkBack Basics**

Touch to Explore

# **TalkBack Basics**

Touch to Explore

Read through elements (swipe left or right)

# **TalkBack Basics**

Touch to Explore

Read through elements (swipe left or right)

Double tap to select

# **TalkBack Basics**

Touch to Explore

Read through elements (swipe left or right)

Double tap to select

Tap once to silence

# **TalkBack Basics**

Touch to Explore

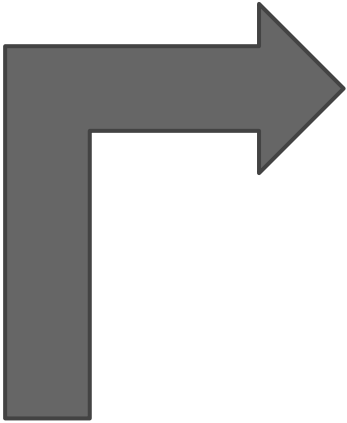
Read through elements (swipe left or right)

Double tap to select

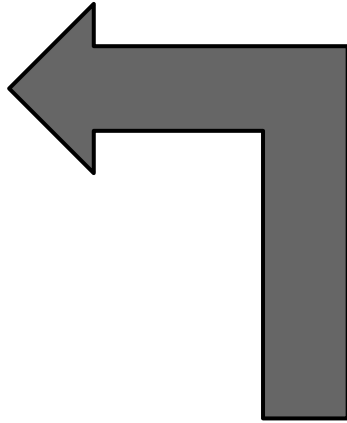
Tap once to silence

Listen for “Earcon” clues

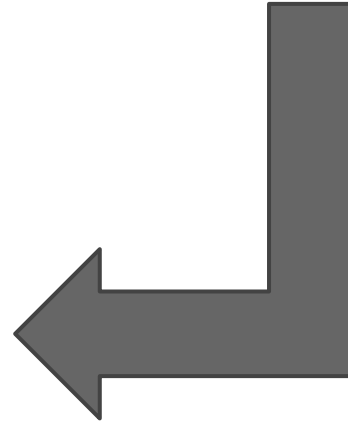
# Special Gestures



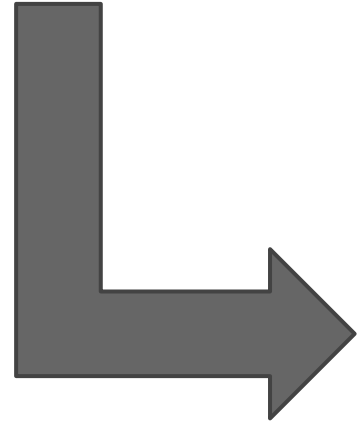
Local  
Context  
Menu



Home

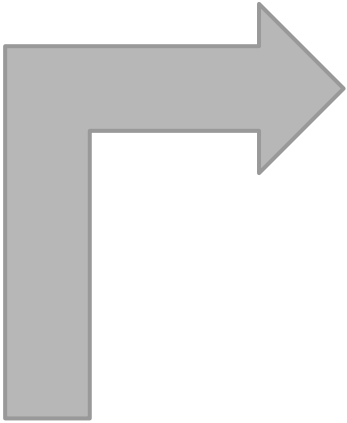


Back

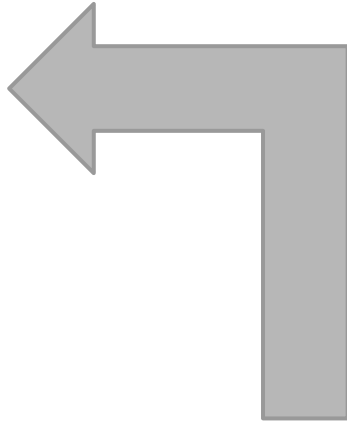


Global  
Context  
Menu

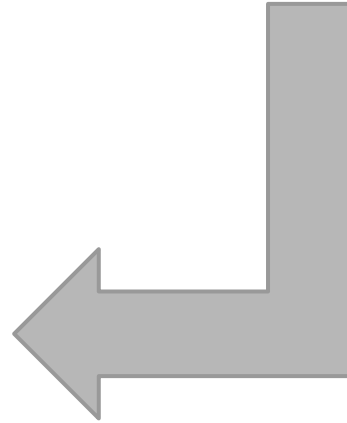
# Special Gestures



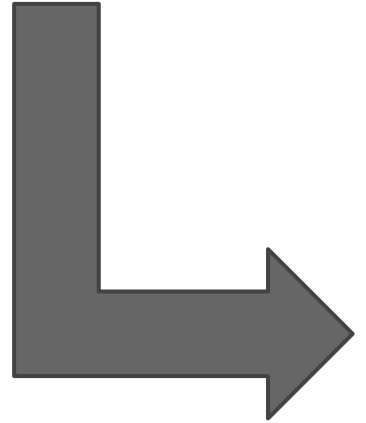
Local  
Context  
Menu



Home



Back



Global  
Context  
Menu

**Development**



# Development Tips & Tricks

TalkBack on-off turns off USB debugging!

# Development Tips & Tricks

TalkBack on-off turns off USB debugging!

TalkBack has limited emulator support

# Development Tips & Tricks

TalkBack on-off turns off USB debugging!

TalkBack has limited emulator support

No screenshots allowed with TalkBack on

# Development Tips & Tricks

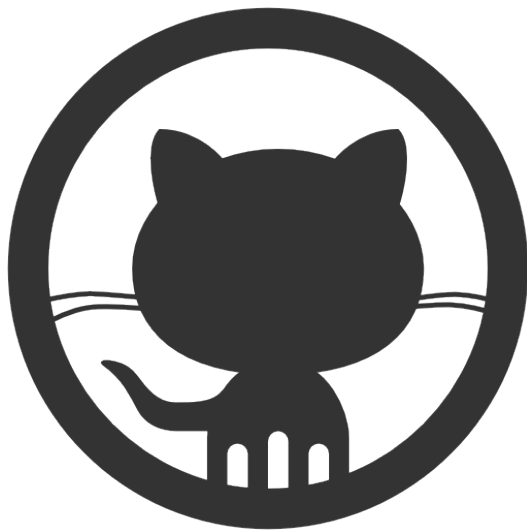
TalkBack on-off turns off USB debugging!

TalkBack has limited emulator support

No screenshots allowed with TalkBack on

Work with QA

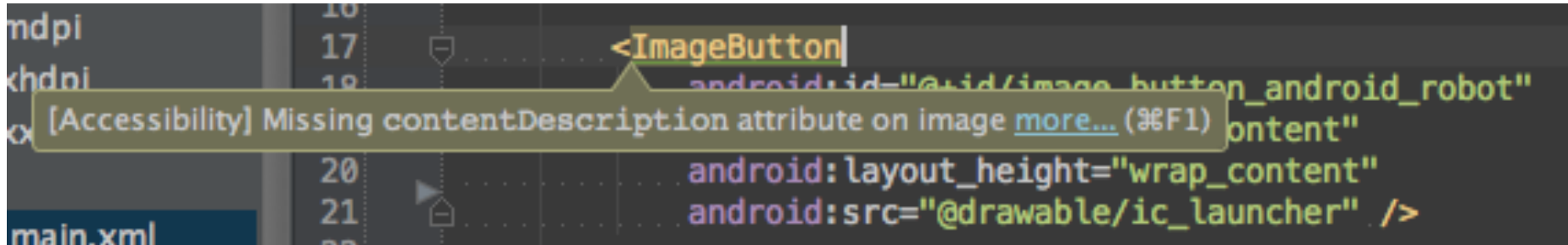
**Demo**



# KioKrofovitch/android-accessibility

 @KellyShuster

# ImageView & ImageButton



**Image with Meaning**



# Image with Meaning

```
<ImageView
```

```
    android:id="@+id/image_bulbasaur_quiz"  
    android:layout_width="75dp"  
    android:layout_height="75dp"  
    android:layout_below="@id/text_question"  
    android:layout_centerHorizontal="true"  
    android:layout_marginTop="10dp"  
    android:contentDescription="@string/bulbasaur"  
    android:src="@drawable/bulbasaur" />
```

# Image with Meaning

```
<ImageView
    android:id="@+id/image_bulbasaur_quiz"
    android:layout_width="75dp"
    android:layout_height="75dp"
    android:layout_below="@id/text_question"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="10dp"
    android:contentDescription="@string/bulbasaur"
    android:src="@drawable/bulbasaur" />
```

**Decorative Image**

# Decorative Image

```
<ImageView  
    android:id="@+id/image_border_1"  
    android:layout_width="120dp"  
    android:layout_height="30dp"  
    android:contentDescription="@null"  
    android:src="@drawable/ballons" />
```

# Decorative Image

```
<ImageView  
    android:id="@+id/image_border_1"  
    android:layout_width="120dp"  
    android:layout_height="30dp"  
    android:contentDescription="@null"  
    android:src="@drawable/ballons" />
```

# Decorative Image

```
<ImageView  
    android:id="@+id/image_border_1"  
    android:layout_width="120dp"  
    android:layout_height="30dp"  
    android:contentDescription="@null"  
    android:src="@drawable/ballons" />
```

# States

# States: Pokeball

```
mAllPokemon = (LinearLayout) findViewById(R.id.layout_all_pokemon);
mPokemonToggleButton = (ImageButton) findViewById(R.id.image_button_pokeball);
mPokemonToggleButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(mAllPokemon.getVisibility() == View.VISIBLE){
            mAllPokemon.setVisibility(View.GONE);
            mPokemonToggleButton.setImageResource(R.drawable.pokeball_closed);
            mPokemonToggleButton.setContentDescription(getString(R.string.show_pokemon));
        }
        else {
            mAllPokemon.setVisibility(View.VISIBLE);
            ...
            mPokemonToggleButton.setImageResource(R.drawable.pokeball_open);
            mPokemonToggleButton.setContentDescription(getString(R.string.hide_pokemon));
        }
    }
});
```



# States: Pokeball

```
mAllPokemon = (LinearLayout) findViewById(R.id.layout_all_pokemon);
mPokemonToggleButton = (ImageButton) findViewById(R.id.image_button_pokeball);
mPokemonToggleButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(mAllPokemon.getVisibility() == View.VISIBLE){
            mAllPokemon.setVisibility(View.GONE);
            mPokemonToggleButton.setImageResource(R.drawable.pokeball_closed);
            mPokemonToggleButton.setContentDescription(getString(R.string.show_pokemon));
        }
        else {
            mAllPokemon.setVisibility(View.VISIBLE);
            ...
            mPokemonToggleButton.setImageResource(R.drawable.pokeball_open);
            mPokemonToggleButton.setContentDescription(getString(R.string.hide_pokemon));
        }
    }
});
```

# States: Pokeball

```
mAllPokemon = (LinearLayout) findViewById(R.id.layout_all_pokemon);
mPokemonToggleButton = (ImageButton) findViewById(R.id.image_button_pokeball);
mPokemonToggleButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(mAllPokemon.getVisibility() == View.VISIBLE){
            mAllPokemon.setVisibility(View.GONE);
            mPokemonToggleButton.setImageResource(R.drawable.pokeball_closed);
            mPokemonToggleButton.setContentDescription(getString(R.string.show_pokemon));
        }
        else {
            mAllPokemon.setVisibility(View.VISIBLE);
            ...
            mPokemonToggleButton.setImageResource(R.drawable.pokeball_open);
            mPokemonToggleButton.setContentDescription(getString(R.string.hide_pokemon));
        }
    }
});
```

# States: Pokeball

```
mAllPokemon = (LinearLayout) findViewById(R.id.layout_all_pokemon);
mPokemonToggleButton = (ImageButton) findViewById(R.id.image_button_pokeball);
mPokemonToggleButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(mAllPokemon.getVisibility() == View.VISIBLE){
            mAllPokemon.setVisibility(View.GONE);
            mPokemonToggleButton.setImageResource(R.drawable.pokeball_closed);
            mPokemonToggleButton.setContentDescription(getString(R.string.show_pokemon));
        }
        else {
            mAllPokemon.setVisibility(View.VISIBLE);
            ...
            mPokemonToggleButton.setImageResource(R.drawable.pokeball_open);
            mPokemonToggleButton.setContentDescription(getString(R.string.hide_pokemon));
        }
    }
});
```

# States: Pokeball

```
mAllPokemon = (LinearLayout) findViewById(R.id.layout_all_pokemon);
mPokemonToggleButton = (ImageButton) findViewById(R.id.image_button_pokeball);
mPokemonToggleButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(mAllPokemon.getVisibility() == View.VISIBLE){
            mAllPokemon.setVisibility(View.GONE);
            mPokemonToggleButton.setImageResource(R.drawable.pokeball_closed);
            mPokemonToggleButton.setContentDescription(getString(R.string.show_pokemon));
        }
        else {
            mAllPokemon.setVisibility(View.VISIBLE);
            ...
            mPokemonToggleButton.setImageResource(R.drawable.pokeball_open);
            mPokemonToggleButton.setContentDescription(getString(R.string.hide_pokemon));
        }
    }
});
```

# States: Pokeball

```
mAllPokemon = (LinearLayout) findViewById(R.id.layout_all_pokemon);
mPokemonToggleButton = (ImageButton) findViewById(R.id.image_button_pokeball);
mPokemonToggleButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(mAllPokemon.getVisibility() == View.VISIBLE){
            mAllPokemon.setVisibility(View.GONE);
            mPokemonToggleButton.setImageResource(R.drawable.pokeball_closed);
            mPokemonToggleButton.setContentDescription(getString(R.string.show_pokemon));
        }
        else {
            mAllPokemon.setVisibility(View.VISIBLE);
            ...
            mPokemonToggleButton.setImageResource(R.drawable.pokeball_open);
            mPokemonToggleButton.setContentDescription(getString(R.string.hide_pokemon));
        }
    }
});
```

# States: Pokeball

```
mAllPokemon = (LinearLayout) findViewById(R.id.layout_all_pokemon);
mPokemonToggleButton = (ImageButton) findViewById(R.id.image_button_pokeball);
mPokemonToggleButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(mAllPokemon.getVisibility() == View.VISIBLE){
            mAllPokemon.setVisibility(View.GONE);
            mPokemonToggleButton.setImageResource(R.drawable.pokeball_closed);
            mPokemonToggleButton.setContentDescription(getString(R.string.show_pokemon));
        }
        else {
            mAllPokemon.setVisibility(View.VISIBLE);
            ...
            mPokemonToggleButton.setImageResource(R.drawable.pokeball_open);
            mPokemonToggleButton.setContentDescription(getString(R.string.hide_pokemon));
        }
    }
});
```

# States: Pokemon Visibility

```
@Override
public void onClick(View v) {
    if(mAllPokemon.getVisibility() == View.VISIBLE){
        ...
    }
    else {
        mAllPokemon.setVisibility(View.VISIBLE);
        if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.JELLY_BEAN) {
            mAllPokemon.announceForAccessibility(getString(R.string.now_showing_pokemon));
        }
        else{
            mAllPokemon.setContentDescription(getString(R.string.now_showing_pokemon));
            mAllPokemon.sendAccessibilityEvent(AccessibilityEvent.TYPE_VIEW_FOCUSED);
        }
        mPokemonToggleButton.setImageResource(R.drawable.pokeball_open);
        mPokemonToggleButton.setContentDescription(getString(R.string.hide_pokemon));
    }
}
```

# States: Pokemon Visibility

```
@Override
public void onClick(View v) {
    if(mAllPokemon.getVisibility() == View.VISIBLE){
        ...
    }
    else {
        mAllPokemon.setVisibility(View.VISIBLE);
        if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.JELLY_BEAN) {
            mAllPokemon.announceForAccessibility(getString(R.string.now_showing_pokemon));
        }
        else{
            mAllPokemon.setContentDescription(getString(R.string.now_showing_pokemon));
            mAllPokemon.sendAccessibilityEvent(AccessibilityEvent.TYPE_VIEW_FOCUSED);
        }
        mPokemonToggleButton.setImageResource(R.drawable.pokeball_open);
        mPokemonToggleButton.setContentDescription(getString(R.string.hide_pokemon));
    }
}
```



# States: Pokemon Visibility

```
@Override
public void onClick(View v) {
    if(mAllPokemon.getVisibility() == View.VISIBLE){
        ...
    }
    else {
        mAllPokemon.setVisibility(View.VISIBLE);
        if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.JELLY_BEAN) {
            mAllPokemon.announceForAccessibility(getString(R.string.now_showing_pokemon));
        }
        else{
            mAllPokemon.setContentDescription(getString(R.string.now_showing_pokemon));
            mAllPokemon.sendAccessibilityEvent(AccessibilityEvent.TYPE_VIEW_FOCUSED);
        }
        mPokemonToggleButton.setImageResource(R.drawable.pokeball_open);
        mPokemonToggleButton.setContentDescription(getString(R.string.hide_pokemon));
    }
}
```

**EditText**

# EditText

```
<EditText
```

```
    android:id="@+id/edit_demo"
```

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

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

```
    android:layout_below="@id/text_view_different"
```

```
    android:hint="@string/type_here"/>
```

# EditText

```
<EditText
    android:id="@+id/edit_demo"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/text_view_different"
    android:hint="@string/type_here"/>
```

**TextView is Different for TalkBack**

# TextView is Different for TalkBack

```
<TextView
```

```
    android:id="@+id/text_view_different"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_below="@id/text_view_clickable"  
    android:layout_centerHorizontal="true"  
    android:layout_margin="10dp"  
    android:contentDescription="@string/monday"  
    android:text="@string/mon"  
    android:textSize="20sp" />
```

# TextView is Different for TalkBack

```
<TextView
    android:id="@+id/text_view_different"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/text_view_clickable"
    android:layout_centerHorizontal="true"
    android:layout_margin="10dp"
    android:contentDescription="@string/monday"
    android:text="@string/mon"
    android:textSize="20sp" />
```

# TextView is Different for TalkBack

```
<TextView
    android:id="@+id/text_view_different"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/text_view_clickable"
    android:layout_centerHorizontal="true"
    android:layout_margin="10dp"
    android:contentDescription="@string/monday"
    android:text="@string/mon"
    android:textSize="20sp" />
```



# **Toasts & Appearing Items**

# Hidden TextView

```
@Override
public void onClick(View v) {
    if(mHiddenTextView.getVisibility() == View.VISIBLE){
        mHiddenTextView.setVisibility(View.GONE);
    }
    else {
        mHiddenTextView.setVisibility(View.VISIBLE);
        mHiddenTextView.sendAccessibilityEvent(AccessibilityEvent.TYPE_VIEW_FOCUSED);
    }
}
```

# Hidden TextView

```
@Override
public void onClick(View v) {
    if(mHiddenTextView.getVisibility() == View.VISIBLE){
        mHiddenTextView.setVisibility(View.GONE);
    }
    else {
        mHiddenTextView.setVisibility(View.VISIBLE);
        mHiddenTextView.sendAccessibilityEvent(AccessibilityEvent.TYPE_VIEW_FOCUSED);
    }
}
```

# Dynamic Views

# Dynamic Views

```
mNumberTextView = (TextView) findViewById(R.id.text_view_numbers);  
  
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT) {  
    mNumberTextView.setAccessibilityLiveRegion(View.ACCESSIBILITY_LIVE_REGION_POLITE);  
}
```

**Read Layout as Element**

# Read Layout as Element

```
// Note you must include v4 to use this.  
// Make sure text is read on the *layout* for VI users, not on  
// individual textViews  
ViewCompat.setImportantForAccessibility(  
    mTextReadMe, ViewCompat.IMPORTANT_FOR_ACCESSIBILITY_NO);  
ViewCompat.setImportantForAccessibility(  
    mTextAsA, ViewCompat.IMPORTANT_FOR_ACCESSIBILITY_NO);  
ViewCompat.setImportantForAccessibility(  
    mTextSingle, ViewCompat.IMPORTANT_FOR_ACCESSIBILITY_NO);  
ViewCompat.setImportantForAccessibility(  
    mTextItem, ViewCompat.IMPORTANT_FOR_ACCESSIBILITY_NO);
```

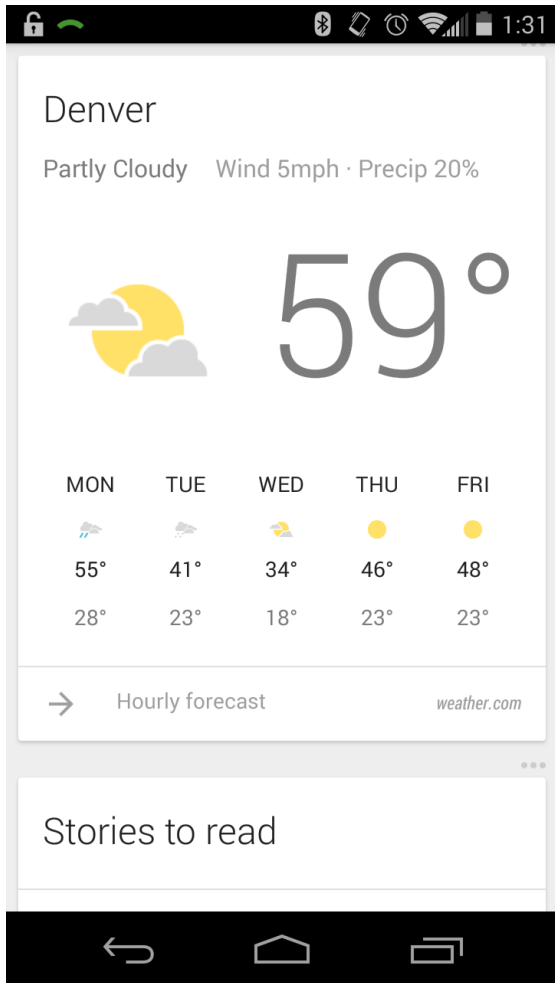
# Read Layout as Element

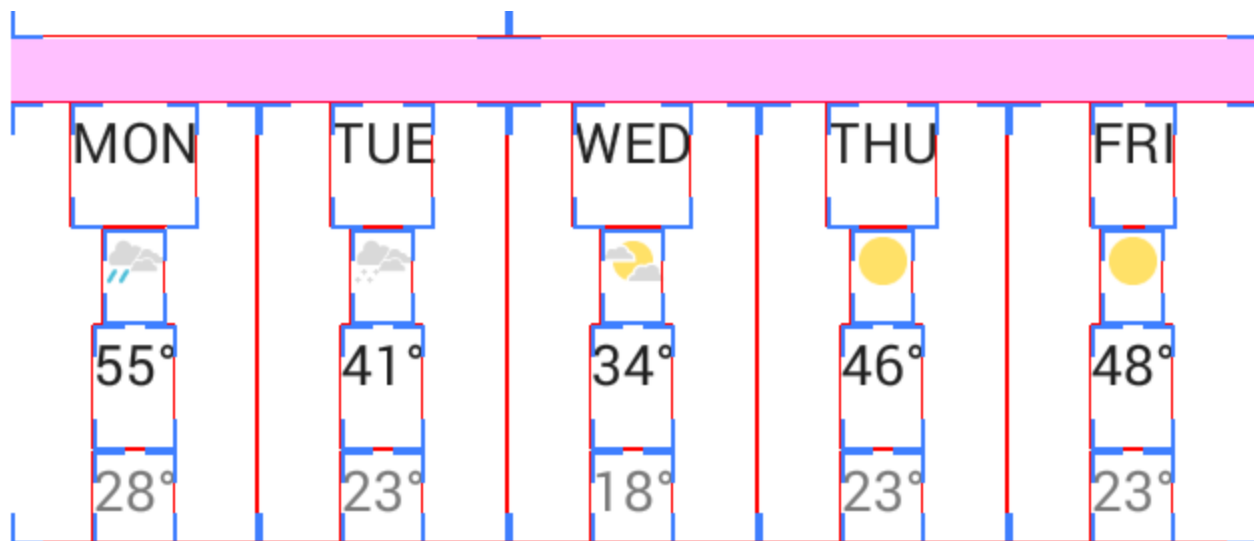
```
// Note you must include v4 to use this.  
// Make sure text is read on the *layout* for VI users, not on  
// individual textViews  
ViewCompat.setImportantForAccessibility(  
    mTextReadMe, ViewCompat.IMPORTANT_FOR_ACCESSIBILITY_NO);  
ViewCompat.setImportantForAccessibility(  
    mTextAsA, ViewCompat.IMPORTANT_FOR_ACCESSIBILITY_NO);  
ViewCompat.setImportantForAccessibility(  
    mTextSingle, ViewCompat.IMPORTANT_FOR_ACCESSIBILITY_NO);  
ViewCompat.setImportantForAccessibility(  
    mTextItem, ViewCompat.IMPORTANT_FOR_ACCESSIBILITY_NO);
```



# Read Layout as Element

```
mLayoutAllText.setContentDescription(  
    mTextReadMe.getText().toString() + " " +  
    mTextAsA.getText().toString()      + " " +  
    mTextSingle.getText().toString() + " " +  
    mTextItem.getText().toString()     );
```





# **Floating Action Button**

# FAB Traversal Order

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_material);

    // Set Traversal Order for Accessibility so FAB isn't hard to access
    FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.floating_action_button);
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP_MR1) {
        fab.setAccessibilityTraversalBefore(R.id.scroll_cards);
    }
    . . .
}
```

# FAB Traversal Order

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_material);

    // Set Traversal Order for Accessibility so FAB isn't hard to access
    FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.floating_action_button);
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP_MR1) {
        fab.setAccessibilityTraversalBefore(R.id.scroll_cards);
    }
    . . .
}
```

# CardView

# CardView

```
CardView charmanderCard = (CardView) findViewById(R.id.card_view_charmander);
charmanderCard.setContentDescription(getString(R.string.charmander_access_details));
charmanderCard.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Toast.makeText(MaterialActivity.this, "Yay Charmander!", Toast.LENGTH_LONG).show();
    }
});
```



# CardView

```
CardView charmanderCard = (CardView) findViewById(R.id.card_view_charmander);  
charmanderCard.setContentDescription(getString(R.string.charmander_access_details));  
charmanderCard.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Toast.makeText(MaterialActivity.this, "Yay Charmander!", Toast.LENGTH_LONG).show();  
    }  
});
```

**Now What?**



**KioKrofovitch/android-accessibility**

eventmobi.com/adcboston

 @KellyShuster

# Resources

Android Accessibility Documentation <https://developer.android.com/guide/topics/ui/accessibility/index.html>

Kelly's Accessibility Demo App with Comments <https://github.com/KioKrofovitch/android-accessibility>

The 48dp Design Rhythm <http://developer.android.com/design/style/metrics-grids.html#48dp-rhythm>

New Accessibility Features in Lollipop <http://www.androidcentral.com/accessibility-features-android-50-lollipop>

VI Opinion: iOS vs Android <https://takesugar.wordpress.com/2014/07/22/accessibility-head-to-head-android-vs-apple/>

Fab Library <https://github.com/futuresimple/android-floating-action-button>

Switch Access Demo <https://www.youtube.com/watch?v=9xqpkNudt58>

# Credits

Blind Institute of Technology <http://blindinstituteoftechnology.org/>

*Case Study: Implementation of ADA on CU Boulder Campus*, Ryan McDonald, June 2009