```
AdView v;
assume width(v, w);
assume height(v, h);
assert w > 0.2 * screen_width;
assert h > 0.2 * screen_height;
```

The size ratio between the ad and the screen is required to be greater than a minimum threshold (i.e., 0.2)

```
AdView v;
View w;
let isFullscreen(u) = width(u, screen_width) /\ height(u, screen_height);
assume isFullscreen(v);
assume contains(w, v);
assert !(exists b, Button(b) /\ contains(w, b));
```

Full-screen ads should not overlap with other buttons.

```
AdView v;
assert !(exists b, Button(b) /\ adjacent(b, v));
```

Ads should not be placed adjacent to a button.

```
View w;
assume nonContentView(w);
assert !(exists v, contains(w, v) /\ AdView(v));
```

Publishers are not permitted to place ads on any non-content-based pages such as thank you, error, log in, or exit screens.

```
View w;
AdView v;
assume contains(w, v);
let isFullscreen(u) = width(u, screen_width) /\ height(u, screen_height);
assert isFullscreen(v) -> (exists m, invoke(w, create, m) /\ loadAd(m, v));
```

Activities that display full-screen ads should call the preload function of the ad when they are created.

```
View w;
AdView v;
assume contains(w, v);
let isFullscreen(u) = width(u, screen_width) /\ height(u, screen_height);
assert isFullscreen(v) -> (exists m, invoke(w, destroy, m) /\ loadAd(m, v));
```

Activities that display full-screen ads should call the preload function of the ad when they are exited.

```
AdView v;
AdView u;
let isFullscreen(u) = width(u, screen_width) /\ height(u, screen_height);
assume isFullscreen(v) /\ isFullscreen(u);
assert !showWindow(v, click, u);
```

Application should not show one full screen ad after clicking another one.

```
View bg;
assume (exists v, View(v) /\ background(bg, v));
let popAd(v) = exists v', showWindow(v, click, v') /\ AdView(v');
assert transparent(bg) -> !popAd(bg);
```

The background of the ad must be not clickable (no clickable ``white space"). If you use the image element as the background of the ad, the image must be not clickable.

```
View v;
View u;
Button v;
assume entryView(v);
let isFullscreen(u) = width(u, screen_width) /\ height(u, screen_height);
assume contains(v, b) /\ showWindow(b, click, u);
assert !(AdView(u) /\ isFullscreen(u));
```

Application should not show a full screen ad when a button in the app entry screen is clicked.

```
AdView v;
assume width(v, w);
assume height(v, h);
assert w < 0.5 * screen_width;
assert h < 0.5 * screen_height;
```

The size of the ad position or ad display cannot exceed 50% of the screen.

```
View x;
AdView y;
AdView z;
assume contains(x, y);
assume contains(x, z);
assert y == z;
```

In apps, at most one ad may be displayed on a single screen at a time

---

```
View v;
assume textSize(v, s);
assert !(s < 10);
```

The smallest recommended font size in Material design is 10sp.

```
Button x;
Button y;
View z;
assume leftOf(x, z);
assume leftOf(z, y);
assume width(z, zw);
assert zw > 0;
```

Buttons should have non-zero margin between them.

```
View w;
FAB x;
FAB y;
assert !(contains(w, x) /\ contains(w, y));
```

Don't display multiple FABs on a single screen.

```
Button b;
View w;
assume entryView(w);
assume contains(w, b);
assert !width(b, screen_width);
```

Button width should be smaller than screen width;

```
AdView v;
let consentForm(d) = exists u, Dialog(d) /\ Button(u) /\ contains(d, u) /\
                               text(u, "confirm") /\
                               invoke(u, click, setConsentFlag);
let showForm(w, d) = entryView(w) /\ launchDialog(w, create, d) /\
                     consentForm(d);
assert personalized(v) -> exists w d, showForm(w, d);
```

Applications that display personalized ads should get user consent when they are started

---

```
Method m;
let recordAudio(m) = call(m, android.media.MediaRecorder);
assume recordAudio(m);
assert exists Button b, invoke(b, click, m);
```

An app should not record audio information without the user clicking on a button.