Picture In Picture (PiP)

By François Beaufort
Dives into Chromium source code

Since April 2017, Chrome for Android O supports Picture In Picture. It allows users to play a `<video>` element in a small overlay window that isn't blocked by other windows, so that they can watch while doing other things.

Here's how it works: open Chrome, go to a website that contains a video and play it fullscreen. From there, press the Home button to go to your Android Home Screen and the playing video will automatically transition to Picture In Picture. That's all! Pretty cool right?

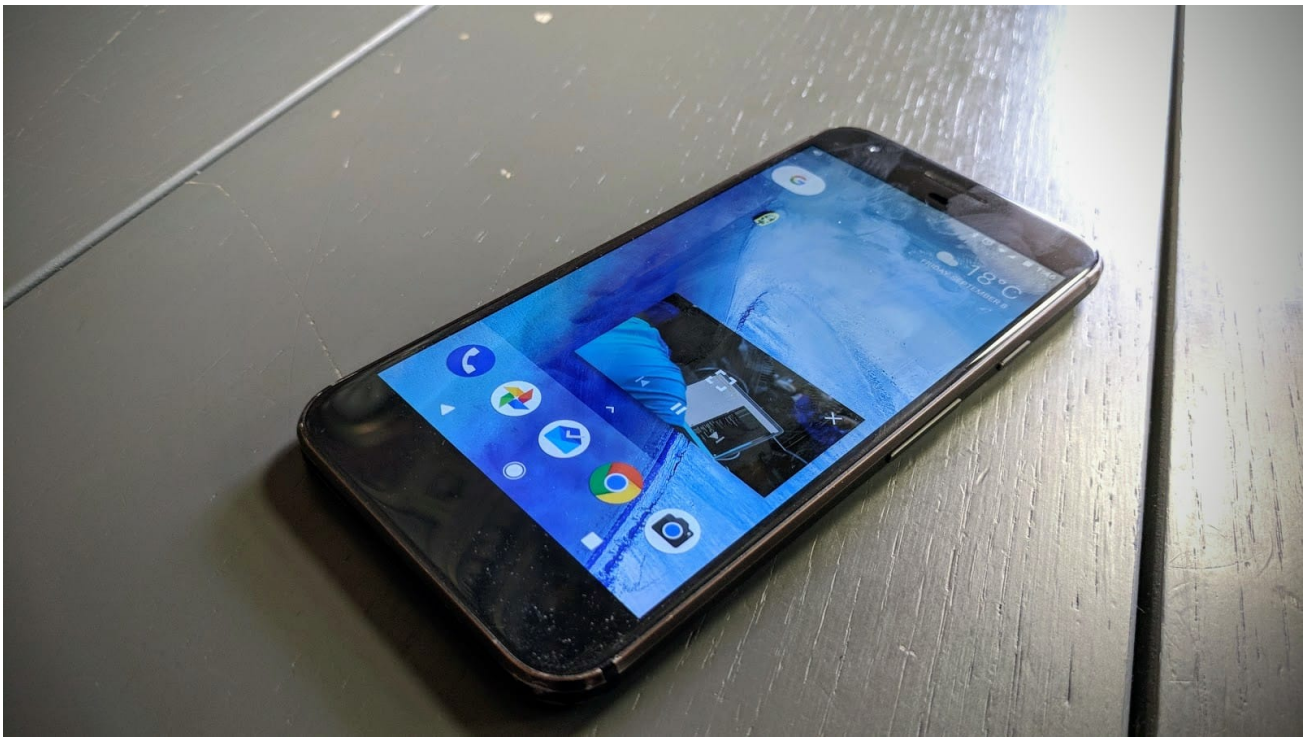

Figure 1. Android Picture in Picture photo

It is, but... what about desktop? What if the website wants to control that experience?

The good news is that a Picture In Picture Web API specification is being drafted as we speak. This spec aims to allow websites to initiate and control this behavior by exposing the following set of properties to the API:

- Notify the website when a video enters and leaves Picture in Picture mode.
- Allow the website to trigger Picture in Picture on a video element via a user gesture.

- Allow the website to exit Picture in Picture.
- Allow the website to check if Picture in Picture can be triggered.

And this is how it could look like:

```
<video id="video" src="https://example.com/file.mp4"></video>

<button id="pipButton"></button>

<script>
  // Hide button if Picture In Picture is not supported.
  pipButton.hidden = !document.pictureInPictureEnabled;

  pipButton.addEventListener('click', function() {
    // If there is no element in Picture In Picture yet, let's request Picture
    // In Picture for the video, otherwise leave it.
    if (!document.pictureInPictureElement) {
      video.requestPictureInPicture()
      .catch(error => {
        // Video failed to enter Picture In Picture mode.
      });
    } else {
      document.exitPictureInPicture()
      .catch(error => {
        // Video failed to leave Picture In Picture mode.
      });
    }
  });
</script>
```

Warning: The code above is not implemented by browsers yet.

Feedback

So what do you think? Please submit your feedback and raise issues in the Picture In Picture WICG repository. We're eager to hear your thoughts!

Preventing Android's default PIP behavior

Today, you can prevent video from using Android's default PiP behavior in Chrome by responding to a resize event, and detecting when the window size has changed significantly

(see code below). This is not recommended as a permanent solution but provides a temporary option until the Web API is implemented.

```
// See whether resize is small enough to be PiP. It's a hack, but it'll
// work for now.
window.addEventListener('resize', function() {
  if (!document.fullscreenElement) {
    return;
  }

  var minimumScreenSize = 0.33;
  var screenArea = screen.width * screen.height;
  var windowArea = window.outerHeight * window.outerWidth;

  // If the size of the window relative to the screen is less than a third,
  // let's assume we're in PiP and exit fullscreen to prevent Auto PiP.
  if ((windowArea / screenArea) < minimumScreenSize) {
    document.exitFullscreen();
  }
});
```

Except as otherwise noted, the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code samples are licensed under the Apache 2.0 License. For details, see our Site Policies. Java is a registered trademark of Oracle and/or its affiliates.

Last updated July 2, 2018.