// Code is taken from DEMO repository of GEE platform

// Use an elevation dataset and terrain functions to create

// a custom visualization of topography.

// Load a global elevation image.

var elev = ee.Image('USGS/GMTED2010');

// Zoom to an area of interest.

Map.setCenter(-121.069, 50.709, 6);

// Add the elevation to the map.

Map.addLayer(elev, {}, 'elev');

// Use the terrain algorithms to compute a hillshade with 8-bit values.

var shade = ee.Terrain.hillshade(elev);

Map.addLayer(shade, {}, 'hillshade', false);

// Create a "sea" variable to be used for cartographic purposes

var sea = elev.lte(0);

Map.addLayer(sea.mask(sea), {palette:'000022'}, 'sea', false);

// Create a custom elevation palette from hex strings.

var elevationPalette = ['006600', '002200', 'fff700', 'ab7634', 'c4d0ff', 'ffffff'];

// Use these visualization parameters, customized by location.

var visParams = {min: 1, max: 3000, palette: elevationPalette};

// Create a mosaic of the sea and the elevation data

var visualized = ee.ImageCollection([

// Mask the elevation to get only land

elev.mask(sea.not()).visualize(visParams),

// Use the sea mask directly to display sea.

sea.mask(sea).visualize({palette:'000022'})

]).mosaic();

// Note that the visualization image doesn't require visualization parameters.

Map.addLayer(visualized, {}, 'elev palette', false);

// Convert the visualized elevation to HSV, first converting to [0, 1] data.

var hsv = visualized.divide(255).rgbToHsv();

// Select only the hue and saturation bands.

var hs = hsv.select(0, 1);

// Convert the hillshade to [0, 1] data, as expected by the HSV algorithm.

var v = shade.divide(255);

// Create a visualization image by converting back to RGB from HSV.

// Note the cast to byte in order to export the image correctly.

var rgb = hs.addBands(v).hsvToRgb().multiply(255).byte();

Map.addLayer(rgb, {}, 'styled');