

accelerometer, autoplay, camera, ch-device-memory, ch-downlink, ch-dpr, ch-ect, ch-lang, ch-prefers-color-scheme, ch-rtt, ch-ua, ch-ua-arch, ch-ua-bitness, ch-ua-full-version, ch-ua-mobile, ch-ua-model, ch-ua-platform, ch-ua-platform-version, ch-ua-reduced, ch-viewport-width, ch-width, clipboard-read, clipboard-write, cross-origin-isolated, display-capture, document-domain, encrypted-media, fullscreen, geolocation, gyroscope, hid, idle-detection, magnetometer, microphone, midi, otp-credentials, payment, picture-in-picture, publickey-credentials-get, screen-wake-lock, serial, sync-xhr, usb, xr-spatial-tracking

https://accounts.shopify.com/oauth/authorize?  
client\_id=7ee65a63608843c577db8b23c4d7316ea0a01bd2f7594f8a9c06ea668c1b775  
c&destination\_uuid=fd6f184b-4c7c-4160-  
ab1b-7d11f36a8ca8&nonce=229afc745a2f7a3b007d2598b20fe7b8&prompt=merge&re  
direct\_uri=https://ama-sub-provider.myshopify.com/admin/auth/identity/  
callback&response\_type=code&scope=email https://api.shopify.com/auth/  
destinations.readonly openid profile https://api.shopify.com/auth/partners.collaborator-  
relationships.readonly https://api.shopify.com/auth/banking.manage https://  
api.shopify.com/auth/shopify-chat.admin.graphql https://api.shopify.com/auth/  
flow.workflows.manage [https://api.shopify.com/auth/organization-  
identity.manage&state=8c8edd20a078f9fd8ab527ffc8fb1693&ui\\_locales=en&ux=shop](https://api.shopify.com/auth/organization-identity.manage&state=8c8edd20a078f9fd8ab527ffc8fb1693&ui_locales=en&ux=shop)

-----  
https://accounts.shopify.com/oauth/authorize?  
client\_id=7ee65a63608843c577db8b23c4d7316ea0a01bd2f7594f8a9c06ea668c1b775  
c&destination\_uuid=fd6f184b-4c7c-4160-  
ab1b-7d11f36a8ca8&nonce=229afc745a2f7a3b007d2598b20fe7b8&prompt=merge&re  
direct\_uri=https://ama-sub-provider.myshopify.com/admin/auth/identity/  
callback&response\_type=code&scope=email https://api.shopify.com/auth/  
destinations.readonly openid profile https://api.shopify.com/auth/partners.collaborator-  
relationships.readonly https://api.shopify.com/auth/banking.manage https://  
api.shopify.com/auth/shopify-chat.admin.graphql https://api.shopify.com/auth/  
flow.workflows.manage [https://api.shopify.com/auth/organization-  
identity.manage&state=8c8edd20a078f9fd8ab527ffc8fb1693&ui\\_locales=en&ux=shop](https://api.shopify.com/auth/organization-identity.manage&state=8c8edd20a078f9fd8ab527ffc8fb1693&ui_locales=en&ux=shop)

client\_id=7ee65a63608843c577db8b23c4d7316ea0a01bd2f7594f8a9c06ea668c1b775  
c&destination\_uuid=fd6f184b-4c7c-4160-  
ab1b-7d11f36a8ca8&nonce=229afc745a2f7a3b007d2598b20fe7b8&prompt=merge&re  
direct\_uri=https%3A%2F%2Fama-sub-provider.myshopify.com%2Fadmin%2Fauth  
%2Fidentity%2Fcallback&response\_type=code&scope=email%20https%3A%2F  
%2Fapi.shopify.com%2Fauth%2Fdestinations.readonly%20openid%20profile%20https  
%3A%2F%2Fapi.shopify.com%2Fauth%2Fpartners.collaborator-relationships.readonly  
%20https%3A%2F%2Fapi.shopify.com%2Fauth%2Fbanking.manage%20https%3A

%2F%2Fapi.shopify.com%2Fauth%2Fshopify-chat.admin.graphql%20https%3A%2F%2Fapi.shopify.com%2Fauth%2Fflow.workflows.manage%20https%3A%2F%2Fapi.shopify.com%2Fauth%2Forganization-identity.manage&state=8c8edd20a078f9fd8ab527ffc8fb1693&ui\_locales=en&ux=shop

-----

<https://ama-sub-provider.myshopify.com/>

"Chromium";v="94", "Google Chrome";v="94", ";Not A Brand";v="99"

Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.61 Mobile Safari/537.36

7ee65a63608843c577db8b23c4d7316ea0a01bd2f7594f8a9c06ea668c1b775c

fdf6184b-4c7c-4160-ab1b-7d11f36a8ca8

229afc745a2f7a3b007d2598b20fe7b8

229afc745a2f7a3b007d2598b20fe7b8

<https://ama-sub-provider.myshopify.com/admin/auth/identity/callback>

code

email <https://api.shopify.com/auth/destinations.readonly> openid profile <https://api.shopify.com/auth/partners.collaborator-relationships.readonly> <https://api.shopify.com/auth/banking.manage> <https://api.shopify.com/auth/shopify-chat.admin.graphql> <https://api.shopify.com/auth/flow.workflows.manage> <https://api.shopify.com/auth/organization-identity.manage>

8c8edd20a078f9fd8ab527ffc8fb1693

en

shop

-----

7ee65a63608843c577db8b23c4d7316ea0a01bd2f7594f8a9c06ea668c1b775c

fdf6184b-4c7c-4160-ab1b-7d11f36a8ca8

229afc745a2f7a3b007d2598b20fe7b8

merge

<https://ama-sub-provider.myshopify.com/admin/auth/identity/callback>

code

email <https://api.shopify.com/auth/destinations.readonly> openid profile <https://api.shopify.com/auth/partners.collaborator-relationships.readonly> <https://api.shopify.com/auth/banking.manage> <https://api.shopify.com/auth/shopify-chat.admin.graphql> <https://api.shopify.com/auth/flow.workflows.manage> <https://api.shopify.com/auth/organization-identity.manage>

8c8edd20a078f9fd8ab527ffc8fb1693

en

shop

-----

<https://ama-sub-provider.myshopify.com/>

"Chromium";v="94", "Google Chrome";v="94", ";Not A Brand";v="99"

?1

"Android"

1

Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.61 Mobile Safari/537.36

7ee65a63608843c577db8b23c4d7316ea0a01bd2f7594f8a9c06ea668c1b775c

fdf6184b-4c7c-4160-ab1b-7d11f36a8ca8

<https://ama-sub-provider.myshopify.com/admin/auth/identity/callback>

email <https://api.shopify.com/auth/destinations.readonly> openid profile <https://api.shopify.com/auth/partners.collaborator-relationships.readonly> <https://api.shopify.com/auth/banking.manage> <https://api.shopify.com/auth/shopify-chat.admin.graphql> <https://api.shopify.com/auth/flow.workflows.manage> <https://api.shopify.com/auth/organization-identity.manage>

"Chromium";v="94", "Google Chrome";v="94", ";Not A Brand";v="99"

Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.61 Mobile Safari/537.36

fdf6184b-4c7c-4160-ab1b-7d11f36a8ca8

-----  
VM156:1250 crbug/1173575, non-JS module files deprecated.  
(anonymous) @ VM156:1250

// Copyright (c) 2012 The Chromium Authors. All rights reserved.  
// Use of this source code is governed by a BSD-style license that can be  
// found in the LICENSE file.

// #import {assertInstanceOf} from './assert.m.js';  
// #import {dispatchSimpleEvent} from './cr.m.js';

/\*\*  
 \* Alias for document.getElementById. Found elements must be HTMLElements.  
 \* @param {string} id The ID of the element to find.  
 \* @return {HTMLElement} The found element or null if not found.  
 \*/  
/\* #export \*/ function \$(id) {  
 // Disable getElementById restriction here, since we are instructing other  
 // places to re-use the \$() that is defined here.  
 // eslint-disable-next-line no-restricted-properties  
 const el = document.getElementById(id);  
 return el ? assertInstanceOf(el, HTMLElement) : null;  
}

// TODO(devlin): This should return SVGElement, but closure compiler is missing  
// those externs.

/\*\*  
 \* Alias for document.getElementById. Found elements must be SVGElements.  
 \* @param {string} id The ID of the element to find.  
 \* @return {Element} The found element or null if not found.  
 \*/  
/\* #export \*/ function getSVGElement(id) {  
 // Disable getElementById restriction here, since it is not suitable for SVG  
 // elements.  
 // eslint-disable-next-line no-restricted-properties  
 const el = document.getElementById(id);  
 return el ? assertInstanceOf(el, Element) : null;  
}

```

}

/**
 * @return {?Element} The currently focused element (including elements that are
 *     behind a shadow root), or null if nothing is focused.
 */
/* #export */ function getDeepActiveElement() {
  let a = document.activeElement;
  while (a && a.shadowRoot && a.shadowRoot.activeElement) {
    a = a.shadowRoot.activeElement;
  }
  return a;
}

//

/**
 * @param {Node} el A node to search for ancestors with lclassName.
 * @param {string} className A class to search for.
 * @return {Element} A node with class of lclassName or null if none is found.
 */
/* #export */ function findAncestorByClass(el, className) {
  return /** @type {Element} */ (findAncestor(el, function(el) {
    return el.classList && el.classList.contains(className);
  }));
}

/**
 * Return the first ancestor for which the {@code predicate} returns true.
 * @param {Node} node The node to check.
 * @param {function(Node):boolean} predicate The function that tests the
 *     nodes.
 * @param {boolean=} includeShadowHosts
 * @return {Node} The found ancestor or null if not found.
 */
/* #export */ function findAncestor(node, predicate, includeShadowHosts) {
  while (node !== null) {
    if (predicate(node)) {
      break;
    }
    node = includeShadowHosts && node instanceof ShadowRoot ? node.host :
      node.parentNode;
  }
  return node;
}

```

```

/**
 * Disables text selection and dragging, with optional callbacks to specify
 * overrides.
 * @param {function(Event):boolean=} opt_allowSelectStart Unless this function
 * is defined and returns true, the onselectionstart event will be
 * suppressed.
 * @param {function(Event):boolean=} opt_allowDragStart Unless this function
 * is defined and returns true, the ondragstart event will be suppressed.
 */
/* #export */ function disableTextSelectAndDrag(
    opt_allowSelectStart, opt_allowDragStart) {
    // Disable text selection.
    document.onselectstart = function(e) {
        if (!(opt_allowSelectStart && opt_allowSelectStart.call(this, e))) {
            e.preventDefault();
        }
    };

    // Disable dragging.
    document.ondragstart = function(e) {
        if (!(opt_allowDragStart && opt_allowDragStart.call(this, e))) {
            e.preventDefault();
        }
    };
}

/**
 * Check the directionality of the page.
 * @return {boolean} True if Chrome is running an RTL UI.
 */
/* #export */ function isRTL() {
    return document.documentElement.dir === 'rtl';
}

/**
 * Get an element that's known to exist by its ID. We use this instead of just
 * calling getElementById and not checking the result because this lets us
 * satisfy the JSCompiler type system.
 * @param {string} id The identifier name.
 * @return {!HTMLElement} the Element.
 */
/* #export */ function getRequiredElement(id) {
    return assertInstanceOf(
        $(id), HTMLElement, 'Missing required element: ' + id);
}

```

```

/**
 * Query an element that's known to exist by a selector. We use this instead of
 * just calling querySelector and not checking the result because this lets us
 * satisfy the JSCompiler type system.
 * @param {string} selectors CSS selectors to query the element.
 * @param {(!Document!!DocumentFragment!!Element)=} opt_context An optional
 *   context object for querySelector.
 * @return {!HTMLElement} the Element.
 */
/* #export */ function queryRequiredElement(selectors, opt_context) {
  const element = (opt_context || document).querySelector(selectors);
  return assertInstanceOf(
    element, HTMLElement, 'Missing required element: ' + selectors);
}

/**
 * Creates a new URL which is the old URL with a GET param of key=value.
 * @param {string} url The base URL. There is not sanity checking on the URL so
 *   it must be passed in a proper format.
 * @param {string} key The key of the param.
 * @param {string} value The value of the param.
 * @return {string} The new URL.
 */
/* #export */ function appendParam(url, key, value) {
  const param = encodeURIComponent(key) + '=' + encodeURIComponent(value);

  if (url.indexOf('?') === -1) {
    return url + '?' + param;
  }
  return url + '&' + param;
}

/**
 * Creates an element of a specified type with a specified class name.
 * @param {string} type The node type.
 * @param {string} className The class name to use.
 * @return {Element} The created element.
 */
/* #export */ function createElementWithClassName(type, className) {
  const elm = document.createElement(type);
  elm.className = className;
  return elm;
}

/**
 * transitionend does not always fire (e.g. when animation is aborted

```

```

* or when no paint happens during the animation). This function sets up
* a timer and emulate the event if it is not fired when the timer expires.
* @param {HTMLElement} el The element to watch for transitionend.
* @param {number=} opt_timeOut The maximum wait time in milliseconds for the
*   transitionend to happen. If not specified, it is fetched from !ell
*   using the transitionDuration style value.
*/
/* #export */ function ensureTransitionEndEvent(el, opt_timeOut) {
  if (opt_timeOut === undefined) {
    const style = getComputedStyle(el);
    opt_timeOut = parseFloat(style.transitionDuration) * 1000;

    // Give an additional 50ms buffer for the animation to complete.
    opt_timeOut += 50;
  }

  let fired = false;
  el.addEventListener('transitionend', function f(e) {
    el.removeEventListener('transitionend', f);
    fired = true;
  });
  window.setTimeout(function() {
    if (!fired) {
      cr.dispatchSimpleEvent(el, 'transitionend', true);
    }
  }, opt_timeOut);
}

/**
 * Alias for document.scrollTop getter.
 * @param {HTMLDocument} doc The document node where information will be
 *   queried from.
 * @return {number} The Y document scroll offset.
 */
/* #export */ function scrollTopForDocument(doc) {
  return doc.documentElement.scrollTop || doc.body.scrollTop;
}

/**
 * Alias for document.scrollTop setter.
 * @param {HTMLDocument} doc The document node where information will be
 *   queried from.
 * @param {number} value The target Y scroll offset.
 */
/* #export */ function setScrollTopForDocument(doc, value) {
  doc.documentElement.scrollTop = doc.body.scrollTop = value;
}

```



```

}

/**
 * Alias for document.scrollTop getter.
 * @param {!HTMLDocument} doc The document node where information will be
 *   queried from.
 * @return {number} The X document scroll offset.
 */
/* #export */ function scrollTopForDocument(doc) {
  return doc.documentElement.scrollTop || doc.body.scrollTop;
}

/**
 * Alias for document.scrollTop setter.
 * @param {!HTMLDocument} doc The document node where information will be
 *   queried from.
 * @param {number} value The target X scroll offset.
 */
/* #export */ function setScrollLeftForDocument(doc, value) {
  doc.documentElement.scrollTop = doc.body.scrollTop = value;
}

/**
 * Replaces '&', '<', '>', '"', and "'" characters with their HTML encoding.
 * @param {string} original The original string.
 * @return {string} The string with all the characters mentioned above replaced.
 */
/* #export */ function HTMLEscape(original) {
  return original.replace(/&/g, '&amp;')
    .replace(/</g, '&lt;')
    .replace(/>/g, '&gt;')
    .replace(/"/g, '&quot;')
    .replace(/'/g, '&#39;');
}

/**
 * Shortens the provided string (if necessary) to a string of length at most
 * |maxLength|.
 * @param {string} original The original string.
 * @param {number} maxLength The maximum length allowed for the string.
 * @return {string} The original string if its length does not exceed
 *   |maxLength|. Otherwise the first |maxLength| - 1 characters with '...'
 *   appended.
 */
/* #export */ function elide(original, maxLength) {
  if (original.length <= maxLength) {

```

```

    return original;
}
return original.substring(0, maxLength - 1) + '\u2026';
}

```

```

/**
 * Quote a string so it can be used in a regular expression.
 * @param {string} str The source string.
 * @return {string} The escaped string.
 */
/* #export */ function quoteString(str) {
    return str.replace(/([\\.\+*\?[\^\]\$\(\)\{\}\=\!<\>\|:])/g, '\\$1');
}

```

```

/**
 * Calls |callback| and stops listening the first time any event in |eventNames|
 * is triggered on |target|.
 * @param {!EventTarget} target
 * @param {!Array<string>|string} eventNames Array or space-delimited string of
 *     event names to listen to (e.g. 'click mousedown').
 * @param {function(!Event)} callback Called at most once. The
 *     optional return value is passed on by the listener.
 */
/* #export */ function listenOnce(target, eventNames, callback) {
    if (!Array.isArray(eventNames)) {
        eventNames = eventNames.split(/ +/);
    }

```

```

    const removeAllAndCallCallback = function(event) {
        eventNames.forEach(function(eventName) {
            target.removeEventListener(eventName, removeAllAndCallCallback, false);
        });
        return callback(event);
    };

```

```

    eventNames.forEach(function(eventName) {
        target.addEventListener(eventName, removeAllAndCallCallback, false);
    });
}

```

```

/**
 * @param {!Event} e
 * @return {boolean} Whether a modifier key was down when processing |e|.
 */
/* #export */ function hasKeyModifiers(e) {
    return !! (e.altKey || e.ctrlKey || e.metaKey || e.shiftKey);
}

```

```

}

/**
 * @param {!Element} el
 * @return {boolean} Whether the element is interactive via text input.
 */
/* #export */ function isTextInputElement(el) {
  return el.tagName === 'INPUT' || el.tagName === 'TEXTAREA';
}

/* #ignore */ console.warn('crbug/1173575, non-JS module files deprecated.');
```

-----

```

VM164:2134 crbug/1173575, non-JS module files deprecated.
(anonymous) @ VM164:2134
```

```

// Copyright (c) 2012 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.
```

```

/**
 * @fileoverview This file defines a singleton which provides access to all data
 * that is available as soon as the page's resources are loaded (before DOM
 * content has finished loading). This data includes both localized strings and
 * any data that is important to have ready from a very early stage (e.g. things
 * that must be displayed right away).
 *
 * Note that loadTimeData is not guaranteed to be consistent between page
 * refreshes (https://crbug.com/740629) and should not contain values that might
 * change if the page is re-opened later.
 */
```

```

/** @type {!LoadTimeData} */
// eslint-disable-next-line no-var
var loadTimeData;
```

```

class LoadTimeData {
  constructor() {
    /** @type {?Object} */
    this.data_ = null;
  }
}
```

```

/**
 * Sets the backing object.
```

```

*
* Note that there is no getter for ldata_l to discourage abuse of the form:
*
*   var value = loadTimeData.data()['key'];
*
* @param {Object} value The de-serialized page data.
*/
set data(value) {
  expect(!this.data_, 'Re-setting data. ');
  this.data_ = value;
}

/**
* @param {string} id An ID of a value that might exist.
* @return {boolean} True if lidl is a key in the dictionary.
*/
valueExists(id) {
  return id in this.data_;
}

/**
* Fetches a value, expecting that it exists.
* @param {string} id The key that identifies the desired value.
* @return {*} The corresponding value.
*/
getValue(id) {
  expect(this.data_, 'No data. Did you remember to include strings.js?');
  const value = this.data_[id];
  expect(typeof value !== 'undefined', 'Could not find value for ' + id);
  return value;
}

/**
* As above, but also makes sure that the value is a string.
* @param {string} id The key that identifies the desired string.
* @return {string} The corresponding string value.
*/
getString(id) {
  const value = this.getValue(id);
  expectIsType(id, value, 'string');
  return /** @type {string} */ (value);
}

/**
* Returns a formatted localized string where $1 to $9 are replaced by the
* second to the tenth argument.

```

```

* @param {string} id The ID of the string we want.
* @param {...(string|number)} var_args The extra values to include in the
*   formatted output.
* @return {string} The formatted string.
*/
getStringF(id, var_args) {
  const value = this.getString(id);
  if (!value) {
    return "";
  }

  const args = Array.prototype.slice.call(arguments);
  args[0] = value;
  return this.substituteString.apply(this, args);
}

/**
* Returns a formatted localized string where $1 to $9 are replaced by the
* second to the tenth argument. Any standalone $ signs must be escaped as
* $$$.
* @param {string} label The label to substitute through.
*   This is not an resource ID.
* @param {...(string|number)} var_args The extra values to include in the
*   formatted output.
* @return {string} The formatted string.
*/
substituteString(label, var_args) {
  const varArgs = arguments;
  return label.replace(/\$(.|\n)/g, function(m) {
    expect(m.match(/^\$[1-9]/), 'Unescaped $ found in localized string. ');
    return m === '$$' ? '$' : varArgs[m[1]];
  });
}

/**
* Returns a formatted string where $1 to $9 are replaced by the second to
* tenth argument, split apart into a list of pieces describing how the
* substitution was performed. Any standalone $ signs must be escaped as $$.
* @param {string} label A localized string to substitute through.
*   This is not an resource ID.
* @param {...(string|number)} var_args The extra values to include in the
*   formatted output.
* @return {Array<{value: string, arg: (null|string)}>} The formatted
*   string pieces.
*/
getSubstitutedStringPieces(label, var_args) {

```

```

const varArgs = arguments;
// Split the string by separately matching all occurrences of $1-9 and of
// non $1-9 pieces.
const pieces = (label.match(/(\$(1-9)|((!^\$)|\$(!^1-9)|$))+/g) ||
    []).map(function(p) {
    // Pieces that are not $1-9 should be returned after replacing $$
    // with $.
    if (!p.match(/^\$(1-9)$/)) {
        expect(
            (p.match(/^\$/g) || []).length % 2 === 0,
            'Unescaped $ found in localized string.');
        return {value: p.replace(/^\$/g, '$'), arg: null};
    }

    // Otherwise, return the substitution value.
    return {value: varArgs[p[1]], arg: p};
});

return pieces;
}

/**
 * As above, but also makes sure that the value is a boolean.
 * @param {string} id The key that identifies the desired boolean.
 * @return {boolean} The corresponding boolean value.
 */
getBoolean(id) {
    const value = this.getValue(id);
    expectIsType(id, value, 'boolean');
    return /** @type {boolean} */ (value);
}

/**
 * As above, but also makes sure that the value is an integer.
 * @param {string} id The key that identifies the desired number.
 * @return {number} The corresponding number value.
 */
getInteger(id) {
    const value = this.getValue(id);
    expectIsType(id, value, 'number');
    expect(value === Math.floor(value), 'Number isn\'t integer: ' + value);
    return /** @type {number} */ (value);
}

/**
 * Override values in loadTimeData with the values found in Ireplacementsl.

```

```

* @param {Object} replacements The dictionary object of keys to replace.
*/
overrideValues(replacements) {
  expect(
    typeof replacements === 'object',
    'Replacements must be a dictionary object.');
```

```

  for (const key in replacements) {
    this.data_[key] = replacements[key];
  }
}

/**
 * Reset loadTimeData's data. Should only be used in tests.
 * @param {?Object} newData The data to restore to, when null restores to
 *   unset state.
 */
resetForTesting(newData = null) {
  this.data_ = newData;
}

/**
 * @return {boolean} Whether loadTimeData.data has been set.
 */
isInitialized() {
  return this.data_ !== null;
}
}

/**
 * Checks condition, throws error message if expectation fails.
 * @param {*} condition The condition to check for truthiness.
 * @param {string} message The message to display if the check fails.
 */
function expect(condition, message) {
  if (!condition) {
    throw new Error(
      'Unexpected condition on ' + document.location.href + ': ' + message);
  }
}

/**
 * Checks that the given value has the given type.
 * @param {string} id The id of the value (only used for error message).
 * @param {*} value The value to check the type on.
 * @param {string} type The type we expect lvalue to be.
 */

```

```
function expectIsType(id, value, type) {
  expect(
    typeof value === type, '[' + value + ']' (' + id + ') is not a ' + type);
}
```

```
expect(!loadTimeData, 'should only include this file once');
loadTimeData = new LoadTimeData;
```

```
// Expose !loadTimeData directly on !window!, since within a JS module the
// scope is local and not all files have been updated to import the exported
// !loadTimeData! explicitly.
window.loadTimeData = loadTimeData;
```

```
console.warn('crbug/1173575, non-JS module files deprecated.');
```

-----

```
// Copyright (c) 2012 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.
```

```
/**
 * @fileoverview This file defines a singleton which provides access to all data
 * that is available as soon as the page's resources are loaded (before DOM
 * content has finished loading). This data includes both localized strings and
 * any data that is important to have ready from a very early stage (e.g. things
 * that must be displayed right away).
 *
 * Note that loadTimeData is not guaranteed to be consistent between page
 * refreshes (https://crbug.com/740629) and should not contain values that might
 * change if the page is re-opened later.
 */
```

```
/** @type {!LoadTimeData} */
// eslint-disable-next-line no-var
var loadTimeData;
```

```
class LoadTimeData {
  constructor() {
    /** @type {?Object} */
    this.data_ = null;
  }
}
```

```
/**
 * Sets the backing object.
```



```

*
* Note that there is no getter for ldata_l to discourage abuse of the form:
*
*   var value = loadTimeData.data()['key'];
*
* @param {Object} value The de-serialized page data.
*/
set data(value) {
  expect(!this.data_, 'Re-setting data. ');
  this.data_ = value;
}

/**
* @param {string} id An ID of a value that might exist.
* @return {boolean} True if lidl is a key in the dictionary.
*/
valueExists(id) {
  return id in this.data_;
}

/**
* Fetches a value, expecting that it exists.
* @param {string} id The key that identifies the desired value.
* @return {*} The corresponding value.
*/
getValue(id) {
  expect(this.data_, 'No data. Did you remember to include strings.js?');
  const value = this.data_[id];
  expect(typeof value !== 'undefined', 'Could not find value for ' + id);
  return value;
}

/**
* As above, but also makes sure that the value is a string.
* @param {string} id The key that identifies the desired string.
* @return {string} The corresponding string value.
*/
getString(id) {
  const value = this.getValue(id);
  expectIsType(id, value, 'string');
  return /** @type {string} */ (value);
}

/**
* Returns a formatted localized string where $1 to $9 are replaced by the
* second to the tenth argument.

```

```

* @param {string} id The ID of the string we want.
* @param {...(string|number)} var_args The extra values to include in the
*   formatted output.
* @return {string} The formatted string.
*/
getStringF(id, var_args) {
  const value = this.getString(id);
  if (!value) {
    return "";
  }

  const args = Array.prototype.slice.call(arguments);
  args[0] = value;
  return this.substituteString.apply(this, args);
}

/**
* Returns a formatted localized string where $1 to $9 are replaced by the
* second to the tenth argument. Any standalone $ signs must be escaped as
* $$$.
* @param {string} label The label to substitute through.
*   This is not an resource ID.
* @param {...(string|number)} var_args The extra values to include in the
*   formatted output.
* @return {string} The formatted string.
*/
substituteString(label, var_args) {
  const varArgs = arguments;
  return label.replace(/\$(.|\n)/g, function(m) {
    expect(m.match(/^\$[1-9]/), 'Unescaped $ found in localized string. ');
    return m === '$$' ? '$' : varArgs[m[1]];
  });
}

/**
* Returns a formatted string where $1 to $9 are replaced by the second to
* tenth argument, split apart into a list of pieces describing how the
* substitution was performed. Any standalone $ signs must be escaped as $$.
* @param {string} label A localized string to substitute through.
*   This is not an resource ID.
* @param {...(string|number)} var_args The extra values to include in the
*   formatted output.
* @return {!Array<!{value: string, arg: (null|string)}>}> The formatted
*   string pieces.
*/
getSubstitutedStringPieces(label, var_args) {

```

```

const varArgs = arguments;
// Split the string by separately matching all occurrences of $1-9 and of
// non $1-9 pieces.
const pieces = (label.match(/(\$(1-9)|((!^\$)|\$(!^1-9)|\$))+/g) ||
    []).map(function(p) {
    // Pieces that are not $1-9 should be returned after replacing $$
    // with $.
    if (!p.match(/^\$(1-9)$/)) {
        expect(
            (p.match(/^\$/g) || []).length % 2 === 0,
            'Unescaped $ found in localized string.');
        return {value: p.replace(/^\$/g, '$'), arg: null};
    }

    // Otherwise, return the substitution value.
    return {value: varArgs[p[1]], arg: p};
});

return pieces;
}

/**
 * As above, but also makes sure that the value is a boolean.
 * @param {string} id The key that identifies the desired boolean.
 * @return {boolean} The corresponding boolean value.
 */
getBoolean(id) {
    const value = this.getValue(id);
    expectIsType(id, value, 'boolean');
    return /** @type {boolean} */ (value);
}

/**
 * As above, but also makes sure that the value is an integer.
 * @param {string} id The key that identifies the desired number.
 * @return {number} The corresponding number value.
 */
getInteger(id) {
    const value = this.getValue(id);
    expectIsType(id, value, 'number');
    expect(value === Math.floor(value), 'Number isn\'t integer: ' + value);
    return /** @type {number} */ (value);
}

/**
 * Override values in loadTimeData with the values found in Ireplacementsl.

```

```

    * @param {Object} replacements The dictionary object of keys to replace.
    */
    overrideValues(replacements) {
        expect(
            typeof replacements === 'object',
            'Replacements must be a dictionary object. ');
        for (const key in replacements) {
            this.data_[key] = replacements[key];
        }
    }
}

/**
 * Reset loadTimeData's data. Should only be used in tests.
 * @param {?Object} newData The data to restore to, when null restores to
 *   unset state.
 */
resetForTesting(newData = null) {
    this.data_ = newData;
}

/**
 * @return {boolean} Whether loadTimeData.data has been set.
 */
isInitialized() {
    return this.data_ !== null;
}
}

/**
 * Checks condition, throws error message if expectation fails.
 * @param {*} condition The condition to check for truthiness.
 * @param {string} message The message to display if the check fails.
 */
function expect(condition, message) {
    if (!condition) {
        throw new Error(
            'Unexpected condition on ' + document.location.href + ': ' + message);
    }
}

/**
 * Checks that the given value has the given type.
 * @param {string} id The id of the value (only used for error message).
 * @param {*} value The value to check the type on.
 * @param {string} type The type we expect lvalue to be.
 */

```

```
function expectIsType(id, value, type) {
  expect(
    typeof value === type, '[' + value + ']' (' + id + ') is not a ' + type);
}
```

```
expect(!loadTimeData, 'should only include this file once');
loadTimeData = new LoadTimeData;
```

```
// Expose lloadTimeData directly on lwindowl, since within a JS module the
// scope is local and not all files have been updated to import the exported
// lloadTimeData explicitly.
window.loadTimeData = loadTimeData;
```

```
console.warn('crbug/1173575, non-JS module files deprecated.');
```

-----

```
crbug/1173575, non-JS module files deprecated.
(anonymous) @ VM31:929
```

```
// Copyright (c) 2013 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.
```

```
/**
```

```
 * @fileoverview Assertion support.
```

```
 */
```

```
/**
```

```
 * Verify lconditionl is truthy and return lconditionl if so.
```

```
 * @template T
```

```
 * @param {T} condition A condition to check for truthiness. Note that this
```

```
 * may be used to test whether a value is defined or not, and we don't want
```

```
 * to force a cast to Boolean.
```

```
 * @param {string=} opt_message A message to show on failure.
```

```
 * @return {T} A non-null lconditionl.
```

```
 * @closurePrimitive {asserts.truthy}
```

```
 * @suppress {reportUnknownTypes} because T is not sufficiently constrained.
```

```
 */
```

```
/* #export */ function assert(condition, opt_message) {
```

```
  if (!condition) {
```

```
    let message = 'Assertion failed';
```

```
    if (opt_message) {
```

```
      message = message + ': ' + opt_message;
```

```
    }
```

```

const error = new Error(message);
const global = function() {
  const thisOrSelf = this || self;
  /** @type {boolean} */
  thisOrSelf.traceAssertionsForTesting;
  return thisOrSelf;
}();
if (global.traceAssertionsForTesting) {
  console.warn(error.stack);
}
throw error;
}
return condition;
}

/**
 * Call this from places in the code that should never be reached.
 *
 * For example, handling all the values of enum with a switch() like this:
 *
 * function getValueFromEnum(enum) {
 *   switch (enum) {
 *     case ENUM_FIRST_OF_TWO:
 *       return first
 *     case ENUM_LAST_OF_TWO:
 *       return last;
 *   }
 *   assertNotReached();
 *   return document;
 * }
 *
 * This code should only be hit in the case of serious programmer error or
 * unexpected input.
 *
 * @param {string=} opt_message A message to show when this is hit.
 * @closurePrimitive {asserts.fail}
 */
/* #export */ function assertNotReached(opt_message) {
  assert(false, opt_message || 'Unreachable code hit');
}

/**
 * @param {*} value The value to check.
 * @param {function(new: T, ...)} type A user-defined constructor.
 * @param {string=} opt_message A message to show when this is hit.
 * @return {T}

```

```

* @template T
*/
/* #export */ function assertInstanceOf(value, type, opt_message) {
  // We don't use assert immediately here so that we avoid constructing an error
  // message if we don't have to.
  if (!(value instanceof type)) {
    assertNotReached(
      opt_message ||
      'Value ' + value + ' is not a[n] ' + (type.name || typeof type));
  }
  return value;
}

/* #ignore */ console.warn('crbug/1173575, non-JS module files deprecated.');
```

-----

```

crbug/1173575, non-JS module files deprecated.
(anonymous) @ VM32:1250
```

```

// Copyright (c) 2012 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.
```

```

// #import {assertInstanceOf} from './assert.m.js';
// #import {dispatchSimpleEvent} from './cr.m.js';
```

```

/**
 * Alias for document.getElementById. Found elements must be HTMLElements.
 * @param {string} id The ID of the element to find.
 * @return {HTMLElement} The found element or null if not found.
 */
/* #export */ function $(id) {
  // Disable getElementById restriction here, since we are instructing other
  // places to re-use the $() that is defined here.
  // eslint-disable-next-line no-restricted-properties
  const el = document.getElementById(id);
  return el ? assertInstanceOf(el, HTMLElement) : null;
}
```

```

// TODO(devlin): This should return SVGElement, but closure compiler is missing
// those externs.
```

```

/**
 * Alias for document.getElementById. Found elements must be SVGElements.
```

```

* @param {string} id The ID of the element to find.
* @return {Element} The found element or null if not found.
*/
/* #export */ function getSVGElement(id) {
  // Disable getElementById restriction here, since it is not suitable for SVG
  // elements.
  // eslint-disable-next-line no-restricted-properties
  const el = document.getElementById(id);
  return el ? assertInstanceOf(el, Element) : null;
}

/**
* @return {?Element} The currently focused element (including elements that are
*   behind a shadow root), or null if nothing is focused.
*/
/* #export */ function getDeepActiveElement() {
  let a = document.activeElement;
  while (a && a.shadowRoot && a.shadowRoot.activeElement) {
    a = a.shadowRoot.activeElement;
  }
  return a;
}

//

/**
* @param {Node} el A node to search for ancestors with lclassName.
* @param {string} className A class to search for.
* @return {Element} A node with class of lclassName or null if none is found.
*/
/* #export */ function findAncestorByClass(el, className) {
  return /** @type {Element} */ (findAncestor(el, function(el) {
    return el.classList && el.classList.contains(className);
  }));
}

/**
* Return the first ancestor for which the {@code predicate} returns true.
* @param {Node} node The node to check.
* @param {function(Node):boolean} predicate The function that tests the
*   nodes.
* @param {boolean=} includeShadowHosts
* @return {Node} The found ancestor or null if not found.
*/
/* #export */ function findAncestor(node, predicate, includeShadowHosts) {
  while (node !== null) {

```



```

    if (predicate(node)) {
        break;
    }
    node = includeShadowHosts && node instanceof ShadowRoot ? node.host :
        node.parentNode;
}
return node;
}

/**
 * Disables text selection and dragging, with optional callbacks to specify
 * overrides.
 * @param {function(Event):boolean=} opt_allowSelectStart Unless this function
 * is defined and returns true, the onselectionstart event will be
 * suppressed.
 * @param {function(Event):boolean=} opt_allowDragStart Unless this function
 * is defined and returns true, the ondragstart event will be suppressed.
 */
/* #export */ function disableTextSelectAndDrag(
    opt_allowSelectStart, opt_allowDragStart) {
    // Disable text selection.
    document.onselectstart = function(e) {
        if (!(opt_allowSelectStart && opt_allowSelectStart.call(this, e))) {
            e.preventDefault();
        }
    };

    // Disable dragging.
    document.ondragstart = function(e) {
        if (!(opt_allowDragStart && opt_allowDragStart.call(this, e))) {
            e.preventDefault();
        }
    };
}

/**
 * Check the directionality of the page.
 * @return {boolean} True if Chrome is running an RTL UI.
 */
/* #export */ function isRTL() {
    return document.documentElement.dir === 'rtl';
}

/**
 * Get an element that's known to exist by its ID. We use this instead of just
 * calling getElementById and not checking the result because this lets us

```

```

* satisfy the JSCompiler type system.
* @param {string} id The identifier name.
* @return {!HTMLElement} the Element.
*/
/* #export */ function getRequiredElement(id) {
  return assertInstanceOf(
    $(id), HTMLElement, 'Missing required element: ' + id);
}

/**
* Query an element that's known to exist by a selector. We use this instead of
* just calling querySelector and not checking the result because this lets us
* satisfy the JSCompiler type system.
* @param {string} selectors CSS selectors to query the element.
* @param {(!Document!!DocumentFragment!!Element)=} opt_context An optional
*   context object for querySelector.
* @return {!HTMLElement} the Element.
*/
/* #export */ function queryRequiredElement(selectors, opt_context) {
  const element = (opt_context || document).querySelector(selectors);
  return assertInstanceOf(
    element, HTMLElement, 'Missing required element: ' + selectors);
}

/**
* Creates a new URL which is the old URL with a GET param of key=value.
* @param {string} url The base URL. There is not sanity checking on the URL so
*   it must be passed in a proper format.
* @param {string} key The key of the param.
* @param {string} value The value of the param.
* @return {string} The new URL.
*/
/* #export */ function appendParam(url, key, value) {
  const param = encodeURIComponent(key) + '=' + encodeURIComponent(value);

  if (url.indexOf('?') === -1) {
    return url + '?' + param;
  }
  return url + '&' + param;
}

/**
* Creates an element of a specified type with a specified class name.
* @param {string} type The node type.
* @param {string} className The class name to use.
* @return {Element} The created element.

```

```

*/
/* #export */ function createElementWithClassName(type, className) {
  const elm = document.createElement(type);
  elm.className = className;
  return elm;
}

/**
 * transitionend does not always fire (e.g. when animation is aborted
 * or when no paint happens during the animation). This function sets up
 * a timer and emulate the event if it is not fired when the timer expires.
 * @param {!HTMLElement} el The element to watch for transitionend.
 * @param {number=} opt_timeOut The maximum wait time in milliseconds for the
 *   transitionend to happen. If not specified, it is fetched from !ell
 *   using the transitionDuration style value.
 */
/* #export */ function ensureTransitionEndEvent(el, opt_timeOut) {
  if (opt_timeOut === undefined) {
    const style = getComputedStyle(el);
    opt_timeOut = parseFloat(style.transitionDuration) * 1000;

    // Give an additional 50ms buffer for the animation to complete.
    opt_timeOut += 50;
  }

  let fired = false;
  el.addEventListener('transitionend', function f(e) {
    el.removeEventListener('transitionend', f);
    fired = true;
  });
  window.setTimeout(function() {
    if (!fired) {
      cr.dispatchSimpleEvent(el, 'transitionend', true);
    }
  }, opt_timeOut);
}

/**
 * Alias for document.scrollTop getter.
 * @param {!HTMLDocument} doc The document node where information will be
 *   queried from.
 * @return {number} The Y document scroll offset.
 */
/* #export */ function scrollTopForDocument(doc) {
  return doc.documentElement.scrollTop || doc.body.scrollTop;
}

```

```

/**
 * Alias for document.scrollTop setter.
 * @param {!HTMLDocument} doc The document node where information will be
 *   queried from.
 * @param {number} value The target Y scroll offset.
 */
/* #export */ function scrollTopForDocument(doc, value) {
  doc.documentElement.scrollTop = doc.body.scrollTop = value;
}

```

```

/**
 * Alias for document.scrollLeft getter.
 * @param {!HTMLDocument} doc The document node where information will be
 *   queried from.
 * @return {number} The X document scroll offset.
 */
/* #export */ function scrollLeftForDocument(doc) {
  return doc.documentElement.scrollLeft || doc.body.scrollLeft;
}

```

```

/**
 * Alias for document.scrollLeft setter.
 * @param {!HTMLDocument} doc The document node where information will be
 *   queried from.
 * @param {number} value The target X scroll offset.
 */
/* #export */ function setScrollLeftForDocument(doc, value) {
  doc.documentElement.scrollLeft = doc.body.scrollLeft = value;
}

```

```

/**
 * Replaces '&', '<', '>', '"', and "'" characters with their HTML encoding.
 * @param {string} original The original string.
 * @return {string} The string with all the characters mentioned above replaced.
 */
/* #export */ function HTMLEscape(original) {
  return original.replace(/&/g, '&amp;')
    .replace(/</g, '&lt;')
    .replace(/>/g, '&gt;')
    .replace(/"/g, '&quot;')
    .replace(/'/g, '&#39;');
}

```

```

/**
 * Shortens the provided string (if necessary) to a string of length at most

```

```

* |maxLength|.
* @param {string} original The original string.
* @param {number} maxLength The maximum length allowed for the string.
* @return {string} The original string if its length does not exceed
*   |maxLength|. Otherwise the first |maxLength| - 1 characters with '...'
*   appended.
*/
/* #export */ function elide(original, maxLength) {
  if (original.length <= maxLength) {
    return original;
  }
  return original.substring(0, maxLength - 1) + '\u2026';
}

/**
* Quote a string so it can be used in a regular expression.
* @param {string} str The source string.
* @return {string} The escaped string.
*/
/* #export */ function quoteString(str) {
  return str.replace(/([\\.\|+*\|?\|^\|]\$\\(|\|)\|}\|=\|!<|>\\:])/g, '\\$1');
}

/**
* Calls |callback| and stops listening the first time any event in |eventNames|
* is triggered on |target|.
* @param {!EventTarget} target
* @param {!Array<string>|string} eventNames Array or space-delimited string of
*   event names to listen to (e.g. 'click mousedown').
* @param {function(!Event)} callback Called at most once. The
*   optional return value is passed on by the listener.
*/
/* #export */ function listenOnce(target, eventNames, callback) {
  if (!Array.isArray(eventNames)) {
    eventNames = eventNames.split(/ +/);
  }

  const removeAllAndCallCallback = function(event) {
    eventNames.forEach(function(eventName) {
      target.removeEventListener(eventName, removeAllAndCallCallback, false);
    });
    return callback(event);
  };

  eventNames.forEach(function(eventName) {
    target.addEventListener(eventName, removeAllAndCallCallback, false);
  });
}

```

```

});
}

/**
 * @param {!Event} e
 * @return {boolean} Whether a modifier key was down when processing lel.
 */
/* #export */ function hasKeyModifiers(e) {
  return !(e.altKey || e.ctrlKey || e.metaKey || e.shiftKey);
}

/**
 * @param {!Element} el
 * @return {boolean} Whether the element is interactive via text input.
 */
/* #export */ function isTextInputElement(el) {
  return el.tagName === 'INPUT' || el.tagName === 'TEXTAREA';
}

/* #ignore */ console.warn('crbug/1173575, non-JS module files deprecated.');
```

-----

crbug/1173575, non-JS module files deprecated.  
 (anonymous) @ VM40:2134

// Copyright (c) 2012 The Chromium Authors. All rights reserved.  
 // Use of this source code is governed by a BSD-style license that can be  
 // found in the LICENSE file.

```

/**
 * @fileoverview This file defines a singleton which provides access to all data
 * that is available as soon as the page's resources are loaded (before DOM
 * content has finished loading). This data includes both localized strings and
 * any data that is important to have ready from a very early stage (e.g. things
 * that must be displayed right away).
 *
 * Note that loadTimeData is not guaranteed to be consistent between page
 * refreshes (https://crbug.com/740629) and should not contain values that might
 * change if the page is re-opened later.
 */
```

```

/** @type {!LoadTimeData} */
// eslint-disable-next-line no-var
var loadTimeData;
```

```

class LoadTimeData {
  constructor() {
    /** @type {?Object} */
    this.data_ = null;
  }

  /**
   * Sets the backing object.
   *
   * Note that there is no getter for |data_| to discourage abuse of the form:
   *
   *   var value = loadTimeData.data()['key'];
   *
   * @param {Object} value The de-serialized page data.
   */
  set data(value) {
    expect(!this.data_, 'Re-setting data.');
```

this.data\_ = value;

```
  }

  /**
   * @param {string} id An ID of a value that might exist.
   * @return {boolean} True if |id| is a key in the dictionary.
   */
  valueExists(id) {
    return id in this.data_;
  }

  /**
   * Fetches a value, expecting that it exists.
   * @param {string} id The key that identifies the desired value.
   * @return {*} The corresponding value.
   */
  getValue(id) {
    expect(this.data_, 'No data. Did you remember to include strings.js?');
    const value = this.data_[id];
    expect(typeof value !== 'undefined', 'Could not find value for ' + id);
    return value;
  }

  /**
   * As above, but also makes sure that the value is a string.
   * @param {string} id The key that identifies the desired string.
   * @return {string} The corresponding string value.
   */
  getString(id) {

```

```

    const value = this.getValue(id);
    expectIsType(id, value, 'string');
    return /** @type {string} */ (value);
}

```

```

/**
 * Returns a formatted localized string where $1 to $9 are replaced by the
 * second to the tenth argument.
 * @param {string} id The ID of the string we want.
 * @param {...(string|number)} var_args The extra values to include in the
 *   formatted output.
 * @return {string} The formatted string.
 */

```

```

getStringF(id, var_args) {
    const value = this.getString(id);
    if (!value) {
        return "";
    }
}

```

```

    const args = Array.prototype.slice.call(arguments);
    args[0] = value;
    return this.substituteString.apply(this, args);
}

```

```

/**
 * Returns a formatted localized string where $1 to $9 are replaced by the
 * second to the tenth argument. Any standalone $ signs must be escaped as
 * $$$.
 * @param {string} label The label to substitute through.
 *   This is not an resource ID.
 * @param {...(string|number)} var_args The extra values to include in the
 *   formatted output.
 * @return {string} The formatted string.
 */

```

```

substituteString(label, var_args) {
    const varArgs = arguments;
    return label.replace(/\$(.|\$|\n)/g, function(m) {
        expect(m.match(/^\$[1-9]/), 'Unescaped $ found in localized string.');
        return m === '$$' ? '$' : varArgs[m[1]];
    });
}

```

```

/**
 * Returns a formatted string where $1 to $9 are replaced by the second to
 * tenth argument, split apart into a list of pieces describing how the
 * substitution was performed. Any standalone $ signs must be escaped as $$.

```



```

* @param {string} label A localized string to substitute through.
* This is not an resource ID.
* @param {...(string|number)} var_args The extra values to include in the
* formatted output.
* @return {!Array<!(value: string, arg: (null|string))>} The formatted
* string pieces.
*/
getSubstitutedStringPieces(label, var_args) {
  const varArgs = arguments;
  // Split the string by separately matching all occurrences of $1-9 and of
  // non $1-9 pieces.
  const pieces = (label.match(/(\$(1-9)|((^[^$]|$([1-9]|$))+)/g) ||
    []).map(function(p) {
    // Pieces that are not $1-9 should be returned after replacing $$
    // with $.
    if (!p.match(/^\$(1-9)$/)) {
      expect(
        (p.match(/^\$/g) || []).length % 2 === 0,
        'Unescaped $ found in localized string.');
      return {value: p.replace(/^\$/g, '$'), arg: null};
    }

    // Otherwise, return the substitution value.
    return {value: varArgs[p[1]], arg: p};
  });

  return pieces;
}

/**
* As above, but also makes sure that the value is a boolean.
* @param {string} id The key that identifies the desired boolean.
* @return {boolean} The corresponding boolean value.
*/
getBoolean(id) {
  const value = this.getValue(id);
  expectIsType(id, value, 'boolean');
  return /** @type {boolean} */ (value);
}

/**
* As above, but also makes sure that the value is an integer.
* @param {string} id The key that identifies the desired number.
* @return {number} The corresponding number value.
*/
getInteger(id) {

```

```

    const value = this.getValue(id);
    expectIsType(id, value, 'number');
    expect(value === Math.floor(value), 'Number isn\'t integer: ' + value);
    return /** @type {number} */ (value);
}

/**
 * Override values in loadTimeData with the values found in Ireplacementsl.
 * @param {Object} replacements The dictionary object of keys to replace.
 */
overrideValues(replacements) {
    expect(
        typeof replacements === 'object',
        'Replacements must be a dictionary object.');"
    for (const key in replacements) {
        this.data_[key] = replacements[key];
    }
}

/**
 * Reset loadTimeData's data. Should only be used in tests.
 * @param {?Object} newData The data to restore to, when null restores to
 *     unset state.
 */
resetForTesting(newData = null) {
    this.data_ = newData;
}

/**
 * @return {boolean} Whether loadTimeData.data has been set.
 */
isInitialized() {
    return this.data_ !== null;
}

/**
 * Checks condition, throws error message if expectation fails.
 * @param {*} condition The condition to check for truthiness.
 * @param {string} message The message to display if the check fails.
 */
function expect(condition, message) {
    if (!condition) {
        throw new Error(
            'Unexpected condition on ' + document.location.href + ': ' + message);
    }
}

```

```
}
```

```
/**
```

```
 * Checks that the given value has the given type.
 * @param {string} id The id of the value (only used for error message).
 * @param {*} value The value to check the type on.
 * @param {string} type The type we expect lvalue to be.
 */
```

```
function expectIsType(id, value, type) {
  expect(
    typeof value === type, '[' + value + ']' (' + id + ') is not a ' + type);
}
```

```
expect(!loadTimeData, 'should only include this file once');
loadTimeData = new LoadTimeData;
```

```
// Expose lloadTimeData directly on lwindowl, since within a JS module the
// scope is local and not all files have been updated to import the exported
// lloadTimeData explicitly.
window.loadTimeData = loadTimeData;
```

```
console.warn('crbug/1173575, non-JS module files deprecated.');
```

-----

```
VM31:929 crbug/1173575, non-JS module files deprecated.
(anonymous) @ VM31:929
```

```
// Copyright (c) 2013 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.
```

```
/**
```

```
 * @fileoverview Assertion support.
 */
```

```
/**
```

```
 * Verify lconditionl is truthy and return lconditionl if so.
 * @template T
 * @param {T} condition A condition to check for truthiness. Note that this
 *   may be used to test whether a value is defined or not, and we don't want
 *   to force a cast to Boolean.
 * @param {string=} opt_message A message to show on failure.
 * @return {T} A non-null lconditionl.
 * @closurePrimitive {asserts.truthy}
 * @suppress {reportUnknownTypes} because T is not sufficiently constrained.
```

```

*/
/* #export */ function assert(condition, opt_message) {
  if (!condition) {
    let message = 'Assertion failed';
    if (opt_message) {
      message = message + ': ' + opt_message;
    }
    const error = new Error(message);
    const global = function() {
      const thisOrSelf = this || self;
      /** @type {boolean} */
      thisOrSelf.traceAssertionsForTesting;
      return thisOrSelf;
    }();
    if (global.traceAssertionsForTesting) {
      console.warn(error.stack);
    }
    throw error;
  }
  return condition;
}

/**
 * Call this from places in the code that should never be reached.
 *
 * For example, handling all the values of enum with a switch() like this:
 *
 * function getValueFromEnum(enum) {
 *   switch (enum) {
 *     case ENUM_FIRST_OF_TWO:
 *       return first
 *     case ENUM_LAST_OF_TWO:
 *       return last;
 *   }
 *   assertNotReached();
 *   return document;
 * }
 *
 * This code should only be hit in the case of serious programmer error or
 * unexpected input.
 *
 * @param {string=} opt_message A message to show when this is hit.
 * @closurePrimitive {asserts.fail}
 */
/* #export */ function assertNotReached(opt_message) {
  assert(false, opt_message || 'Unreachable code hit');
}

```

```

}

/**
 * @param {*} value The value to check.
 * @param {function(new: T, ...)} type A user-defined constructor.
 * @param {string=} opt_message A message to show when this is hit.
 * @return {T}
 * @template T
 */
/* #export */ function assertInstanceOf(value, type, opt_message) {
  // We don't use assert immediately here so that we avoid constructing an error
  // message if we don't have to.
  if (!(value instanceof type)) {
    assertNotReached(
      opt_message ||
      'Value ' + value + ' is not a[n] ' + (type.name || typeof type));
  }
  return value;
}

/* #ignore */ console.warn('crbug/1173575, non-JS module files deprecated.');
```

-----

```

crbug/1173575, non-JS module files deprecated.
(anonymous) @ VM32:1250
```

```

// Copyright (c) 2012 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.
```

```

// #import {assertInstanceOf} from './assert.m.js';
// #import {dispatchSimpleEvent} from './cr.m.js';
```

```

/**
 * Alias for document.getElementById. Found elements must be HTMLElements.
 * @param {string} id The ID of the element to find.
 * @return {HTMLElement} The found element or null if not found.
 */
/* #export */ function $(id) {
  // Disable getElementById restriction here, since we are instructing other
  // places to re-use the $() that is defined here.
  // eslint-disable-next-line no-restricted-properties
  const el = document.getElementById(id);
  return el ? assertInstanceOf(el, HTMLElement) : null;
}
```

```
// TODO(devlin): This should return SVGElement, but closure compiler is missing
// those externs.
```

```
/**
```

```
 * Alias for document.getElementById. Found elements must be SVGElements.
```

```
 * @param {string} id The ID of the element to find.
```

```
 * @return {Element} The found element or null if not found.
```

```
 */
```

```
/* #export */ function getSVGElement(id) {
```

```
  // Disable getElementById restriction here, since it is not suitable for SVG
```

```
  // elements.
```

```
  // eslint-disable-next-line no-restricted-properties
```

```
  const el = document.getElementById(id);
```

```
  return el ? assertInstanceOf(el, Element) : null;
```

```
}
```

```
/**
```

```
 * @return {?Element} The currently focused element (including elements that are
 *   behind a shadow root), or null if nothing is focused.
```

```
 */
```

```
/* #export */ function getDeepActiveElement() {
```

```
  let a = document.activeElement;
```

```
  while (a && a.shadowRoot && a.shadowRoot.activeElement) {
```

```
    a = a.shadowRoot.activeElement;
```

```
  }
```

```
  return a;
```

```
}
```

```
//
```

```
/**
```

```
 * @param {Node} el A node to search for ancestors with lclassName.
```

```
 * @param {string} className A class to search for.
```

```
 * @return {Element} A node with class of lclassName or null if none is found.
```

```
 */
```

```
/* #export */ function findAncestorByClass(el, className) {
```

```
  return /** @type {Element} */ (findAncestor(el, function(el) {
```

```
    return el.classList && el.classList.contains(className);
```

```
  }));
```

```
}
```

```
/**
```

```
 * Return the first ancestor for which the {@code predicate} returns true.
```

```
 * @param {Node} node The node to check.
```

```
 * @param {function(Node):boolean} predicate The function that tests the
 *   nodes.
```

```

* @param {boolean=} includeShadowHosts
* @return {Node} The found ancestor or null if not found.
*/
/* #export */ function findAncestor(node, predicate, includeShadowHosts) {
  while (node !== null) {
    if (predicate(node)) {
      break;
    }
    node = includeShadowHosts && node instanceof ShadowRoot ? node.host :
      node.parentNode;
  }
  return node;
}

/**
* Disables text selection and dragging, with optional callbacks to specify
* overrides.
* @param {function(Event):boolean=} opt_allowSelectStart Unless this function
* is defined and returns true, the onselectionstart event will be
* suppressed.
* @param {function(Event):boolean=} opt_allowDragStart Unless this function
* is defined and returns true, the ondragstart event will be suppressed.
*/
/* #export */ function disableTextSelectAndDrag(
  opt_allowSelectStart, opt_allowDragStart) {
  // Disable text selection.
  document.onselectstart = function(e) {
    if (!(opt_allowSelectStart && opt_allowSelectStart.call(this, e))) {
      e.preventDefault();
    }
  };

  // Disable dragging.
  document.ondragstart = function(e) {
    if !(opt_allowDragStart && opt_allowDragStart.call(this, e)) {
      e.preventDefault();
    }
  };
}

/**
* Check the directionality of the page.
* @return {boolean} True if Chrome is running an RTL UI.
*/
/* #export */ function isRTL() {
  return document.documentElement.dir === 'rtl';
}

```

```

}

/**
 * Get an element that's known to exist by its ID. We use this instead of just
 * calling getElementById and not checking the result because this lets us
 * satisfy the JSCompiler type system.
 * @param {string} id The identifier name.
 * @return {!HTMLElement} the Element.
 */
/* #export */ function getRequiredElement(id) {
  return assertInstanceOf(
    $(id), HTMLElement, 'Missing required element: ' + id);
}

/**
 * Query an element that's known to exist by a selector. We use this instead of
 * just calling querySelector and not checking the result because this lets us
 * satisfy the JSCompiler type system.
 * @param {string} selectors CSS selectors to query the element.
 * @param {(!Document!DocumentFragment!Element)=} opt_context An optional
 *   context object for querySelector.
 * @return {!HTMLElement} the Element.
 */
/* #export */ function queryRequiredElement(selectors, opt_context) {
  const element = (opt_context || document).querySelector(selectors);
  return assertInstanceOf(
    element, HTMLElement, 'Missing required element: ' + selectors);
}

/**
 * Creates a new URL which is the old URL with a GET param of key=value.
 * @param {string} url The base URL. There is not sanity checking on the URL so
 *   it must be passed in a proper format.
 * @param {string} key The key of the param.
 * @param {string} value The value of the param.
 * @return {string} The new URL.
 */
/* #export */ function appendParam(url, key, value) {
  const param = encodeURIComponent(key) + '=' + encodeURIComponent(value);

  if (url.indexOf('?') === -1) {
    return url + '?' + param;
  }
  return url + '&' + param;
}

```



```

/**
 * Creates an element of a specified type with a specified class name.
 * @param {string} type The node type.
 * @param {string} className The class name to use.
 * @return {Element} The created element.
 */
/* #export */ function createElementWithClassName(type, className) {
  const elm = document.createElement(type);
  elm.className = className;
  return elm;
}

/**
 * transitionend does not always fire (e.g. when animation is aborted
 * or when no paint happens during the animation). This function sets up
 * a timer and emulate the event if it is not fired when the timer expires.
 * @param {HTMLElement} el The element to watch for transitionend.
 * @param {number=} opt_timeout The maximum wait time in milliseconds for the
 *   transitionend to happen. If not specified, it is fetched from !ell
 *   using the transitionDuration style value.
 */
/* #export */ function ensureTransitionEndEvent(el, opt_timeout) {
  if (opt_timeout === undefined) {
    const style = getComputedStyle(el);
    opt_timeout = parseFloat(style.transitionDuration) * 1000;

    // Give an additional 50ms buffer for the animation to complete.
    opt_timeout += 50;
  }

  let fired = false;
  el.addEventListener('transitionend', function f(e) {
    el.removeEventListener('transitionend', f);
    fired = true;
  });
  window.setTimeout(function() {
    if (!fired) {
      cr.dispatchSimpleEvent(el, 'transitionend', true);
    }
  }, opt_timeout);
}

/**
 * Alias for document.scrollTop getter.
 * @param {HTMLDocument} doc The document node where information will be
 *   queried from.

```

```

* @return {number} The Y document scroll offset.
*/
/* #export */ function scrollTopForDocument(doc) {
  return doc.documentElement.scrollTop || doc.body.scrollTop;
}

/**
* Alias for document.scrollTop setter.
* @param {!HTMLDocument} doc The document node where information will be
*   queried from.
* @param {number} value The target Y scroll offset.
*/
/* #export */ function setScrollTopForDocument(doc, value) {
  doc.documentElement.scrollTop = doc.body.scrollTop = value;
}

/**
* Alias for document.scrollLeft getter.
* @param {!HTMLDocument} doc The document node where information will be
*   queried from.
* @return {number} The X document scroll offset.
*/
/* #export */ function scrollLeftForDocument(doc) {
  return doc.documentElement.scrollLeft || doc.body.scrollLeft;
}

/**
* Alias for document.scrollLeft setter.
* @param {!HTMLDocument} doc The document node where information will be
*   queried from.
* @param {number} value The target X scroll offset.
*/
/* #export */ function setScrollLeftForDocument(doc, value) {
  doc.documentElement.scrollLeft = doc.body.scrollLeft = value;
}

/**
* Replaces '&', '<', '>', '"', and "'" characters with their HTML encoding.
* @param {string} original The original string.
* @return {string} The string with all the characters mentioned above replaced.
*/
/* #export */ function HTMLEscape(original) {
  return original.replace(/&/g, '&amp;')
    .replace(/</g, '&lt;')
    .replace(/>/g, '&gt;')
    .replace(/"/g, '&quot;');
}

```

```

        .replace(/'/g, '&#39;');
    }

    /**
     * Shortens the provided string (if necessary) to a string of length at most
     * |maxLength|.
     * @param {string} original The original string.
     * @param {number} maxLength The maximum length allowed for the string.
     * @return {string} The original string if its length does not exceed
     *     |maxLength|. Otherwise the first |maxLength| - 1 characters with '...'
     *     appended.
     */
    /* #export */ function elide(original, maxLength) {
        if (original.length <= maxLength) {
            return original;
        }
        return original.substring(0, maxLength - 1) + '\u2026';
    }

    /**
     * Quote a string so it can be used in a regular expression.
     * @param {string} str The source string.
     * @return {string} The escaped string.
     */
    /* #export */ function quoteString(str) {
        return str.replace(/([\\.\|+*\|?\|^\|]\$\\(|\)|\{|\}=\|!<|>\\:])/g, '\\$1');
    }

    /**
     * Calls |callback| and stops listening the first time any event in |eventNames|
     * is triggered on |target|.
     * @param {!EventTarget} target
     * @param {!Array<string>|string} eventNames Array or space-delimited string of
     *     event names to listen to (e.g. 'click mousedown').
     * @param {function(!Event)} callback Called at most once. The
     *     optional return value is passed on by the listener.
     */
    /* #export */ function listenOnce(target, eventNames, callback) {
        if (!Array.isArray(eventNames)) {
            eventNames = eventNames.split(/ +/);
        }

        const removeAllAndCallCallback = function(event) {
            eventNames.forEach(function(eventName) {
                target.removeEventListener(eventName, removeAllAndCallCallback, false);
            });
        };
    
```

```

    return callback(event);
};

eventNames.forEach(function(eventName) {
    target.addEventListener(eventName, removeAllAndCallCallback, false);
});
}

/**
 * @param {!Event} e
 * @return {boolean} Whether a modifier key was down when processing |e|.
 */
/* #export */ function hasKeyModifiers(e) {
    return !(e.altKey || e.ctrlKey || e.metaKey || e.shiftKey);
}

/**
 * @param {!Element} el
 * @return {boolean} Whether the element is interactive via text input.
 */
/* #export */ function isTextInputElement(el) {
    return el.tagName === 'INPUT' || el.tagName === 'TEXTAREA';
}

/* #ignore */ console.warn('crbug/1173575, non-JS module files deprecated.');
```

-----

VM40:2134 crbug/1173575, non-JS module files deprecated.  
 (anonymous) @ VM40:2134

// Copyright (c) 2012 The Chromium Authors. All rights reserved.  
 // Use of this source code is governed by a BSD-style license that can be  
 // found in the LICENSE file.

```

/**
 * @fileoverview This file defines a singleton which provides access to all data
 * that is available as soon as the page's resources are loaded (before DOM
 * content has finished loading). This data includes both localized strings and
 * any data that is important to have ready from a very early stage (e.g. things
 * that must be displayed right away).
 *
 * Note that loadTimeData is not guaranteed to be consistent between page
 * refreshes (https://crbug.com/740629) and should not contain values that might
 * change if the page is re-opened later.
 */
```

```

/** @type {!LoadTimeData} */
// eslint-disable-next-line no-var
var loadTimeData;

class LoadTimeData {
  constructor() {
    /** @type {?Object} */
    this.data_ = null;
  }

  /**
   * Sets the backing object.
   *
   * Note that there is no getter for ldata_ to discourage abuse of the form:
   *
   *   var value = loadTimeData.data()['key'];
   *
   * @param {Object} value The de-serialized page data.
   */
  set data(value) {
    expect(!this.data_, 'Re-setting data.');
```

this.data\_ = value;

```
  }

  /**
   * @param {string} id An ID of a value that might exist.
   * @return {boolean} True if lidl is a key in the dictionary.
   */
  valueExists(id) {
    return id in this.data_;
  }

  /**
   * Fetches a value, expecting that it exists.
   * @param {string} id The key that identifies the desired value.
   * @return {*} The corresponding value.
   */
  getValue(id) {
    expect(this.data_, 'No data. Did you remember to include strings.js?');
    const value = this.data_[id];
    expect(typeof value !== 'undefined', 'Could not find value for ' + id);
    return value;
  }

  /**

```

```

* As above, but also makes sure that the value is a string.
* @param {string} id The key that identifies the desired string.
* @return {string} The corresponding string value.
*/
getString(id) {
  const value = this.getValue(id);
  expectIsType(id, value, 'string');
  return /** @type {string} */ (value);
}

/**
* Returns a formatted localized string where $1 to $9 are replaced by the
* second to the tenth argument.
* @param {string} id The ID of the string we want.
* @param {...(string|number)} var_args The extra values to include in the
*   formatted output.
* @return {string} The formatted string.
*/
getStringF(id, var_args) {
  const value = this.getString(id);
  if (!value) {
    return "";
  }

  const args = Array.prototype.slice.call(arguments);
  args[0] = value;
  return this.substituteString.apply(this, args);
}

/**
* Returns a formatted localized string where $1 to $9 are replaced by the
* second to the tenth argument. Any standalone $ signs must be escaped as
* $$$.
* @param {string} label The label to substitute through.
*   This is not an resource ID.
* @param {...(string|number)} var_args The extra values to include in the
*   formatted output.
* @return {string} The formatted string.
*/
substituteString(label, var_args) {
  const varArgs = arguments;
  return label.replace(/\$(.|\n)/g, function(m) {
    expect(m.match(/^\$[1-9]/), 'Unescaped $ found in localized string. ');
    return m === '$$' ? '$' : varArgs[m[1]];
  });
}

```

```

/**
 * Returns a formatted string where $1 to $9 are replaced by the second to
 * tenth argument, split apart into a list of pieces describing how the
 * substitution was performed. Any standalone $ signs must be escaped as $$
 * @param {string} label A localized string to substitute through.
 *   This is not an resource ID.
 * @param {...(string|number)} var_args The extra values to include in the
 *   formatted output.
 * @return {Array<!{value: string, arg: (null|string)}>} The formatted
 *   string pieces.
 */

```

```

getSubstitutedStringPieces(label, var_args) {
  const varArgs = arguments;
  // Split the string by separately matching all occurrences of $1-9 and of
  // non $1-9 pieces.
  const pieces = (label.match(/(\$(1-9)|((\^$\)|\$(1-9)|$))+/g) ||
    []).map(function(p) {
    // Pieces that are not $1-9 should be returned after replacing $$
    // with $.
    if (!p.match(/^\$(1-9)$/)) {
      expect(
        (p.match(/^\$/g) || []).length % 2 === 0,
        'Unescaped $ found in localized string. ');
      return {value: p.replace(/^\$/g, '$'), arg: null};
    }

    // Otherwise, return the substitution value.
    return {value: varArgs[p[1]], arg: p};
  });

  return pieces;
}

```

```

/**
 * As above, but also makes sure that the value is a boolean.
 * @param {string} id The key that identifies the desired boolean.
 * @return {boolean} The corresponding boolean value.
 */

```

```

getBoolean(id) {
  const value = this.getValue(id);
  expectIsType(id, value, 'boolean');
  return /** @type {boolean} */ (value);
}

```

```

/**

```

```

* As above, but also makes sure that the value is an integer.
* @param {string} id The key that identifies the desired number.
* @return {number} The corresponding number value.
*/
getInteger(id) {
  const value = this.getValue(id);
  expectIsType(id, value, 'number');
  expect(value === Math.floor(value), 'Number isn\'t integer: ' + value);
  return /** @type {number} */ (value);
}

/**
* Override values in loadTimeData with the values found in Ireplacementsl.
* @param {Object} replacements The dictionary object of keys to replace.
*/
overrideValues(replacements) {
  expect(
    typeof replacements === 'object',
    'Replacements must be a dictionary object. ');
  for (const key in replacements) {
    this.data_[key] = replacements[key];
  }
}

/**
* Reset loadTimeData's data. Should only be used in tests.
* @param {?Object} newData The data to restore to, when null restores to
*   unset state.
*/
resetForTesting(newData = null) {
  this.data_ = newData;
}

/**
* @return {boolean} Whether loadTimeData.data has been set.
*/
isInitialized() {
  return this.data_ !== null;
}
}

/**
* Checks condition, throws error message if expectation fails.
* @param {*} condition The condition to check for truthiness.
* @param {string} message The message to display if the check fails.
*/

```



```
function expect(condition, message) {
  if (!condition) {
    throw new Error(
      'Unexpected condition on ' + document.location.href + ': ' + message);
  }
}
```

```
/**
 * Checks that the given value has the given type.
 * @param {string} id The id of the value (only used for error message).
 * @param {*} value The value to check the type on.
 * @param {string} type The type we expect lvalue to be.
 */
function expectIsType(id, value, type) {
  expect(
    typeof value === type, '[' + value + '] (' + id + ') is not a ' + type);
}
```

```
expect(!loadTimeData, 'should only include this file once');
loadTimeData = new LoadTimeData;
```

```
// Expose lloadTimeData directly on lwindowl, since within a JS module the
// scope is local and not all files have been updated to import the exported
// lloadTimeData explicitly.
window.loadTimeData = loadTimeData;
```

```
console.warn('crbug/1173575, non-JS module files deprecated.');
```

-----

```
// Copyright 2015 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.
```

```
'use strict';
```

```
// Other constants defined in security_interstitial_page.h.
const SB_BOX_CHECKED = 'boxchecked';
const SB_DISPLAY_CHECK_BOX = 'displaycheckbox';
```

```
// This sets up the Extended Safe Browsing Reporting opt-in, either for
// reporting malware or invalid certificate chains. Does nothing if the
// interstitial type is not SAFEBROWSING or SSL or CAPTIVE_PORTAL.
function setupExtendedReportingCheckbox() {
  const interstitialType = loadTimeData.getString('type');
  if (interstitialType !== 'SAFEBROWSING' && interstitialType !== 'SSL' &&
```

```

    interstitialType !== 'CAPTIVE_PORTAL') {
    return;
}

if (!loadTimeData.getBoolean(SB_DISPLAY_CHECK_BOX)) {
    return;
}

if ($('#privacy-link')) {
    $('#privacy-link').addEventListener('click', function() {

sendCommand(SecurityInterstitialCommandId.CMD_OPEN_REPORTING_PRIVACY);
        return false;
    });
    $('#privacy-link').addEventListener('mousedown', function() {
        return false;
    });
}
$('#opt-in-checkbox').checked = loadTimeData.getBoolean(SB_BOX_CHECKED);
$('#extended-reporting-opt-in').classList.remove('hidden');

const billing =
    interstitialType === 'SAFEBROWSING' && loadTimeData.getBoolean('billing');

let className = 'ssl-opt-in';
if (interstitialType === 'SAFEBROWSING' && !billing) {
    className = 'safe-browsing-opt-in';
}

$('#extended-reporting-opt-in').classList.add(className);

$('body').classList.add('extended-reporting-has-checkbox');

if ($('#whitepaper-link')) {
    $('#whitepaper-link').addEventListener('click', function(event) {
        sendCommand(SecurityInterstitialCommandId.CMD_OPEN_WHITEPAPER);
    });
}

$('#opt-in-checkbox').addEventListener('click', function() {
    sendCommand($('#opt-in-checkbox').checked ?
        SecurityInterstitialCommandId.CMD_DO_REPORT :
        SecurityInterstitialCommandId.CMD_DONT_REPORT);
});
}

```

```

-----

// Copyright 2020 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.

'use strict';

// Other constants defined in security_interstitial_page.h.
const SB_DISPLAY_ENHANCED_PROTECTION_MESSAGE =
  'displayEnhancedProtectionMessage';

// This sets up the enhanced protection message.
function setupEnhancedProtectionMessage() {
  const interstitialType = loadTimeData.getString('type');
  if (interstitialType !== 'SAFEBROWSING' && interstitialType !== 'SSL' &&
      interstitialType !== 'CAPTIVE_PORTAL') {
    return;
  }

  if (!
loadTimeData.getBoolean(SB_DISPLAY_ENHANCED_PROTECTION_MESSAGE)) {
    return;
  }

  if ($('#enhanced-protection-link')) {
    if (mobileNav) {
      // To make sure the touch area of the link is larger than the
      // minimum touch area for accessibility, make the whole block tappable.
      $('#enhanced-protection-message').addEventListener('click', function() {
        sendCommand(SecurityInterstitialCommandId
          .CMD_OPEN_ENHANCED_PROTECTION_SETTINGS);
        return false;
      });
    } else {
      $('#enhanced-protection-link').addEventListener('click', function() {
        sendCommand(SecurityInterstitialCommandId
          .CMD_OPEN_ENHANCED_PROTECTION_SETTINGS);
        return false;
      });
    }
  }
  $('#enhanced-protection-message').classList.remove('hidden');

  const billing =
    interstitialType === 'SAFEBROWSING' && loadTimeData.getBoolean('billing');

```

```
let className = 'ssl-enhanced-protection-message';
if (interstitialType === 'SAFEBROWSING' && !billing) {
  className = 'safe-browsing-enhanced-protection-message';
}

$('enhanced-protection-message').classList.add(className);
}
```

-----

```
// Copyright 2015 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.
```

```
let mobileNav = false;
```

```
/**
```

```
 * For small screen mobile the navigation buttons are moved
 * below the advanced text.
```

```
*/
```

```
function onResize() {
  const helpOuterBox = document.querySelector('#details');
  const mainContent = document.querySelector('#main-content');
  const mediaQuery = '(min-width: 240px) and (max-width: 420px) and ' +
    '(min-height: 401px), ' +
    '(max-height: 560px) and (min-height: 240px) and ' +
    '(min-width: 421px)';
```

```
const detailsHidden = helpOuterBox.classList.contains(HIDDEN_CLASS);
const runnerContainer = document.querySelector('.runner-container');
```

```
// Check for change in nav status.
```

```
if (mobileNav !== window.matchMedia(mediaQuery).matches) {
  mobileNav = !mobileNav;
```

```
// Handle showing the top content / details sections according to state.
```

```
if (mobileNav) {
  mainContent.classList.toggle(HIDDEN_CLASS, !detailsHidden);
  helpOuterBox.classList.toggle(HIDDEN_CLASS, detailsHidden);
  if (runnerContainer) {
    runnerContainer.classList.toggle(HIDDEN_CLASS, !detailsHidden);
  }
} else if (!detailsHidden) {
  // Non mobile nav with visible details.
  mainContent.classList.remove(HIDDEN_CLASS);
```

```

    helpOuterBox.classList.remove(HIDDEN_CLASS);
    if (runnerContainer) {
        runnerContainer.classList.remove(HIDDEN_CLASS);
    }
}
}
}
}

```

```

function setupMobileNav() {
    window.addEventListener('resize', onResize);
    onResize();
}

```

```

document.addEventListener('DOMContentLoaded', setupMobileNav);

```

-----

```

// Copyright 2017 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.

```

```

// This is the shared code for security interstitials. It is used for both SSL
// interstitials and Safe Browsing interstitials.

```

```

/**
 * @typedef {{
 *   dontProceed: function(),
 *   proceed: function(),
 *   showMoreSection: function(),
 *   openHelpCenter: function(),
 *   openDiagnostic: function(),
 *   reload: function(),
 *   openDateSettings: function(),
 *   openLogin: function(),
 *   doReport: function(),
 *   dontReport: function(),
 *   openReportingPrivacy: function(),
 *   openWhitepaper: function(),
 *   reportPhishingError: function(),
 * }}
 */

```

```

// eslint-disable-next-line no-var
var certificateErrorPageController;

```

```

// Should match security_interstitials::SecurityInterstitialCommand
/** @enum {number} */

```

```

const SecurityInterstitialCommandId = {
  CMD_DONT_PROCEED: 0,
  CMD_PROCEED: 1,
  // Ways for user to get more information
  CMD_SHOW_MORE_SECTION: 2,
  CMD_OPEN_HELP_CENTER: 3,
  CMD_OPEN_DIAGNOSTIC: 4,
  // Primary button actions
  CMD_RELOAD: 5,
  CMD_OPEN_DATE_SETTINGS: 6,
  CMD_OPEN_LOGIN: 7,
  // Safe Browsing Extended Reporting
  CMD_DO_REPORT: 8,
  CMD_DONT_REPORT: 9,
  CMD_OPEN_REPORTING_PRIVACY: 10,
  CMD_OPEN_WHITEPAPER: 11,
  // Report a phishing error.
  CMD_REPORT_PHISHING_ERROR: 12,
  // Open enhanced protection settings.
  CMD_OPEN_ENHANCED_PROTECTION_SETTINGS: 13,
};

```

```

const HIDDEN_CLASS = 'hidden';

```

```

/**
 * A convenience method for sending commands to the parent page.
 * @param {SecurityInterstitialCommandId} cmd The command to send.
 */
function sendCommand(cmd) {
  if (window.certificateErrorPageController) {
    switch (cmd) {
      case SecurityInterstitialCommandId.CMD_DONT_PROCEED:
        certificateErrorPageController.dontProceed();
        break;
      case SecurityInterstitialCommandId.CMD_PROCEED:
        certificateErrorPageController.proceed();
        break;
      case SecurityInterstitialCommandId.CMD_SHOW_MORE_SECTION:
        certificateErrorPageController.showMoreSection();
        break;
      case SecurityInterstitialCommandId.CMD_OPEN_HELP_CENTER:
        certificateErrorPageController.openHelpCenter();
        break;
      case SecurityInterstitialCommandId.CMD_OPEN_DIAGNOSTIC:
        certificateErrorPageController.openDiagnostic();
        break;
    }
  }
}

```

```

    case SecurityInterstitialCommandId.CMD_RELOAD:
        certificateErrorPageController.reload();
        break;
    case SecurityInterstitialCommandId.CMD_OPEN_DATE_SETTINGS:
        certificateErrorPageController.openDateSettings();
        break;
    case SecurityInterstitialCommandId.CMD_OPEN_LOGIN:
        certificateErrorPageController.openLogin();
        break;
    case SecurityInterstitialCommandId.CMD_DO_REPORT:
        certificateErrorPageController.doReport();
        break;
    case SecurityInterstitialCommandId.CMD_DONT_REPORT:
        certificateErrorPageController.dontReport();
        break;
    case SecurityInterstitialCommandId.CMD_OPEN_REPORTING_PRIVACY:
        certificateErrorPageController.openReportingPrivacy();
        break;
    case SecurityInterstitialCommandId.CMD_OPEN_WHITEPAPER:
        certificateErrorPageController.openWhitepaper();
        break;
    case SecurityInterstitialCommandId.CMD_REPORT_PHISHING_ERROR:
        certificateErrorPageController.reportPhishingError();
        break;
    case
SecurityInterstitialCommandId.CMD_OPEN_ENHANCED_PROTECTION_SETTINGS:
        certificateErrorPageController.openEnhancedProtectionSettings();
        break;
    }
    return;
}
//
window.domAutomationController.send(cmd);
//
//
}

/**
 * Call this to stop clicks on <a href="#"> links from scrolling to the top of
 * the page (and possibly showing a # in the link).
 */
function preventDefaultOnPoundLinkClicks() {
    document.addEventListener('click', function(e) {
        const anchor = findAncestor(** @type {Node} */ (e.target), function(el) {
            return el.tagName === 'A';
        });
    });
}

```

```

    // Use getAttribute() to prevent URL normalization.
    if (anchor && anchor.getAttribute('href') === '#') {
        e.preventDefault();
    }
});
}

//

//

-----

// Copyright 2014 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.

// This is the shared code for the new (Chrome 37) security interstitials. It is
// used for both SSL interstitials and Safe Browsing interstitials.

let expandedDetails = false;
let keyPressState = 0;

/**
 * This allows errors to be skipped by typing a secret phrase into the page.
 * @param {string} e The key that was just pressed.
 */
function handleKeypress(e) {
    // HTTPS errors are serious and should not be ignored. For testing purposes,
    // other approaches are both safer and have fewer side-effects.
    // See https://goo.gl/ZcZixP for more details.
    const BYPASS_SEQUENCE = window.atob('dGhpc2lzdW5zYWZl');
    if (BYPASS_SEQUENCE.charCodeAt(keyPressState) === e.keyCode) {
        keyPressState++;
        if (keyPressState === BYPASS_SEQUENCE.length) {
            sendCommand(SecurityInterstitialCommandId.CMD_PROCEED);
            keyPressState = 0;
        }
    } else {
        keyPressState = 0;
    }
}

/**
 * This appends a piece of debugging information to the end of the warning.
 * When complete, the caller must also make the debugging div

```



```

* (error-debugging-info) visible.
* @param {string} title The name of this debugging field.
* @param {string} value The value of the debugging field.
* @param {boolean=} fixedWidth If true, the value field is displayed fixed
*                               width.
*/
function appendDebuggingField(title, value, fixedWidth) {
  // The values input here are not trusted. Never use innerHTML on these
  // values!
  const spanTitle = document.createElement('span');
  spanTitle.classList.add('debugging-title');
  spanTitle.innerText = title + ': ';

  const spanValue = document.createElement('span');
  spanValue.classList.add('debugging-content');
  if (fixedWidth) {
    spanValue.classList.add('debugging-content-fixed-width');
  }
  spanValue.innerText = value;

  const pElem = document.createElement('p');
  pElem.classList.add('debugging-content');
  pElem.appendChild(spanTitle);
  pElem.appendChild(spanValue);
  $('error-debugging-info').appendChild(pElem);
}

function toggleDebuggingInfo() {
  const hiddenDebug = $('error-debugging-info').classList.toggle(HIDDEN_CLASS);
  $('error-code').setAttribute('aria-expanded', !hiddenDebug);
}

function setupEvents() {
  const overridable = loadTimeData.getBoolean('overridable');
  const interstitialType = loadTimeData.getString('type');
  const ssl = interstitialType === 'SSL';
  const captivePortal = interstitialType === 'CAPTIVE_PORTAL';
  const badClock = ssl && loadTimeData.getBoolean('bad_clock');
  const lookalike = interstitialType === 'LOOKALIKE';
  const billing =
    interstitialType === 'SAFEBROWSING' && loadTimeData.getBoolean('billing');
  const originPolicy = interstitialType === 'ORIGIN_POLICY';
  const blockedInterception = interstitialType === 'BLOCKED_INTERCEPTION';
  const legacyTls = interstitialType === 'LEGACY_TLS';
  const insecureForm = interstitialType === 'INSECURE_FORM';
  const httpsOnly = interstitialType === 'HTTPS_ONLY';

```

```
const hidePrimaryButton = loadTimeData.getBoolean('hide_primary_button');
const showRecurrentErrorParagraph = loadTimeData.getBoolean(
  'show_recurrent_error_paragraph');
```

```
if (ssl || originPolicy || blockedInterception || legacyTls) {
  $('body').classList.add(badClock ? 'bad-clock' : 'ssl');
  if (loadTimeData.valueExists('errorCode')) {
    $('error-code').textContent = loadTimeData.getString('errorCode');
    $('error-code').classList.remove(HIDDEN_CLASS);
  }
} else if (captivePortal) {
  $('body').classList.add('captive-portal');
} else if (billing) {
  $('body').classList.add('safe-browsing-billing');
} else if (lookalike) {
  $('body').classList.add('lookalike-url');
} else if (insecureForm) {
  $('body').classList.add('insecure-form');
} else if (httpsOnly) {
  $('body').classList.add('https-only');
} else {
  $('body').classList.add('safe-browsing');
  // Override the default theme color.
  document.querySelector('meta[name=theme-color]').setAttribute('content',
    'rgb(217, 48, 37)');
}
```

```
$('icon').classList.add('icon');
```

```
if (legacyTls) {
  $('icon').classList.add('legacy-tls');
}
```

```
if (hidePrimaryButton) {
  $('primary-button').classList.add(HIDDEN_CLASS);
} else {
  $('primary-button').addEventListener('click', function() {
    switch (interstitialType) {
      case 'CAPTIVE_PORTAL':
        sendCommand(SecurityInterstitialCommandId.CMD_OPEN_LOGIN);
        break;

      case 'SSL':
      case 'LEGACY_TLS':
        if (badClock) {
```

```

sendCommand(SecurityInterstitialCommandId.CMD_OPEN_DATE_SETTINGS);
    } else if (overridable) {
        sendCommand(SecurityInterstitialCommandId.CMD_DONT_PROCEED);
    } else {
        sendCommand(SecurityInterstitialCommandId.CMD_RELOAD);
    }
    break;

    case 'SAFEBROWSING':
    case 'ORIGIN_POLICY':
        sendCommand(SecurityInterstitialCommandId.CMD_DONT_PROCEED);
        break;
    case 'HTTPS_ONLY':
    case 'INSECURE_FORM':
    case 'LOOKALIKE':
        sendCommand(SecurityInterstitialCommandId.CMD_DONT_PROCEED);
        break;

    default:
        throw 'Invalid interstitial type';
    }
});
}

if (lookalike || insecureForm || httpsOnly) {
    const proceedButton = 'proceed-button';
    $(proceedButton).classList.remove(HIDDEN_CLASS);
    $(proceedButton).textContent = loadTimeData.getString('proceedButtonText');
    $(proceedButton).addEventListener('click', function(event) {
        sendCommand(SecurityInterstitialCommandId.CMD_PROCEED);
    });
}

if (lookalike) {
    // Lookalike interstitials with a suggested URL have a link in the title:
    // "Did you mean <link>example.com</link>?". Handle those clicks. Lookalike
    // interstitials without a suggested URL don't have this link.
    const dontProceedLink = 'dont-proceed-link';
    if ($(dontProceedLink)) {
        $(dontProceedLink).addEventListener('click', function(event) {
            sendCommand(SecurityInterstitialCommandId.CMD_DONT_PROCEED);
        });
    }
}

if (overridable) {
    const overrideElement = billing ? 'proceed-button' : 'proceed-link';

```

```

// Captive portal page isn't overridable.
$(overrideElement).addEventListener('click', function(event) {
    sendCommand(SecurityInterstitialCommandId.CMD_PROCEED);
});

if (ssl) {
    $(overrideElement).classList.add('small-link');
} else if (billing) {
    $(overrideElement).classList.remove(HIDDEN_CLASS);
    $(overrideElement).textContent =
        loadTimeData.getString('proceedButtonText');
}
} else if (!ssl) {
    $('final-paragraph').classList.add(HIDDEN_CLASS);
}

if (!ssl || !showRecurrentErrorParagraph) {
    $('recurrent-error-message').classList.add(HIDDEN_CLASS);
} else {
    $('body').classList.add('showing-recurrent-error-message');
}

if ($('#diagnostic-link')) {
    $('#diagnostic-link').addEventListener('click', function(event) {
        sendCommand(SecurityInterstitialCommandId.CMD_OPEN_DIAGNOSTIC);
    });
}

if ($('#learn-more-link')) {
    $('#learn-more-link').addEventListener('click', function(event) {
        sendCommand(SecurityInterstitialCommandId.CMD_OPEN_HELP_CENTER);
    });
}

if (captivePortal || billing || lookalike || insecureForm || httpsOnly) {
    // Captive portal, billing, lookalike pages, insecure form, and
    // HTTPS only mode interstitials don't have details buttons.
    $('details-button').classList.add('hidden');
} else {
    $('details-button')
        .setAttribute(
            'aria-expanded', !$('#details').classList.contains(HIDDEN_CLASS));
    $('details-button').addEventListener('click', function(event) {
        const hiddenDetails = $('#details').classList.toggle(HIDDEN_CLASS);
        $('details-button').setAttribute('aria-expanded', !hiddenDetails);
    });
}

```

```

if (mobileNav) {
    // Details appear over the main content on small screens.
    $('main-content').classList.toggle(HIDDEN_CLASS, !hiddenDetails);
} else {
    $('main-content').classList.remove(HIDDEN_CLASS);
}

$('details-button').innerText = hiddenDetails ?
    loadTimeData.getString('openDetails') :
    loadTimeData.getString('closeDetails');
if (!expandedDetails) {
    // Record a histogram entry only the first time that details is opened.
    sendCommand(SecurityInterstitialCommandId.CMD_SHOW_MORE_SECTION);
    expandedDetails = true;
}
});
}

if ($('report-error-link')) {
    $('report-error-link').addEventListener('click', function(event) {

sendCommand(SecurityInterstitialCommandId.CMD_REPORT_PHISHING_ERROR);
    });
}

if (lookalike) {
    console.log(
        'Chrome has determined that ' +
        loadTimeData.getString('lookalikeRequestHostname') +
        ' could be fake or fraudulent.\n\n' +
        'If you believe this is shown in error please visit ' +
        'https://g.co/chrome/lookalike-warnings');
}

preventDefaultOnPoundLinkClicks();
setupExtendedReportingCheckbox();
setupEnhancedProtectionMessage();
setupSSLDebuggingInfo();
document.addEventListener('keypress', handleKeypress);
}

document.addEventListener('DOMContentLoaded', setupEvents);

```

---

<div id="debugging">  
    <div id="error-code" class="error-code" role="button" aria-expanded="false">net::ERR\_CERT\_DATE\_INVALID</div>  
    <div id="error-debugging-info" class="hidden"><p class="debugging-content"><span class="debugging-title">Subject: </span><span class="debugging-content">accounts.shopify.com</span></p><p class="debugging-content"><span class="debugging-title">Issuer: </span><span class="debugging-content">R3</span></p><p class="debugging-content"><span class="debugging-title">Expires on: </span><span class="debugging-content">Nov 13, 2021</span></p><p class="debugging-content"><span class="debugging-title">Current date: </span><span class="debugging-content">Oct 5, 2021</span></p><p class="debugging-content"><span class="debugging-title">PEM encoded chain: </span><span class="debugging-content debugging-content-fixed-width">-----BEGIN CERTIFICATE-----<br>MIIF2DCCBMCGAwIBAgISA3IP+wuRtzmomjRFbuWpyrsCMA0GCSqGSIb3DQEBCwUA<br>MDIx CzAJBgNVBAYTAIVTMRYwFAYDVQQKEw1MZXRQncyBFbmNyeXB0MQswCQYDVQQDD<br>EwJSMzAeFw0yMTA4MTUxNjE3MDNaFw0yMTEyMTMxNjE3MDFaMB8xHTAbBgNVBAMT<br>FGFjY291bnRzLnNob3BpZnkuY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB<br>CgKCAQEAmNO6kENQqqUM0WgozXecZGcPzrlyEbrU/N7mdStXBNjXN9WAEbLSTGrw<br>gkyShj1dbDXVJudJ7YdQXJrl/KJbm87u8a1HEZI4Dmj1FxBIAd+Gs1RsFdQd0nEw<br>5Zgi0Sojj6d87SEqpS2i0Vz6V+hTrbuy+XXP5+J8piNiRp2lkrZQkWKaifZaLyvH<br>w2S2OfTeJxH3SSoKfsA/ST4ZPRDFlppsXgmpFf7sU5u1E3XckNjxl220Q29/vtIs<br>ErLP9noTy1wdKzXJYbu4IT6nVOSs+TGLQBcBCfNQyaG19jMrL299G4QyNy2h6uWX<br>MwPm/Opq3qk3Bnl5uZgyV+mx7EMe1QIDAQABo4IC+TCCAvUwDgYDVR0PAQH/BAQD<br>AgWgMB0GA1UdJQQWMBQGCCsGAQUFBwMBBggrBgEFBQcDAjAMBgNVHRMBAf8EAjAA<br>MB0GA1UdDgQWBBR/TRz5xF8Ma/wcElldjg8fyl3L3zAfBgNVHSMEGDAWgBQULrMX<br>t1hWy65QCUDmH6+dixTCxjBVBggrBgEFBQcBAQRJMEcwIQYIKwYBBQUHMAGGFWH0<br>dHA6Ly9yMy5vLmxlbmNyLm9yZzAiBggrBgEFBQcwAoYWAHR0cDovL3IzLmkubGVu<br>Y3lub3JnLzCBYQYDVRO0RBIHBMIG+gh5hY2NvdW50cy1kZW1vLnNob3BpZnlibG91<br>ZC5jb22CFGFjY291bnRzLnNob3BpZnkuY29tgh9jdXN0b21lcnMitZGVtby5zaG9w<br>aWZ5Y2xvdWQuY29tgiZpZGVudGl0eS1tYWluLWNsdXN0ZXluc2hvcGlmeWNsb3V<br>kLmNvbYliaWRlbnRpdHktbWVyY2hnbG9uY29tZGVudGl0eS1tYWluLWNsdXN0ZXluc2hvcGlmeWNsb3V<br>kLmNvbTBMBG9NVHSAERTBDMAgGBmeBDAECATA3BggrBgEEAYLfEwEBATAoMCYGC<br>CsGAQUFBwIBFhpodHRwOi8vY3BzLmxldHNIbmNyeXB0Lm9yZzCCAQMGCisGAQQB1nkCBAIEgfQEgfEA7wB1AESUZS6w7s6vxEAH2Kj+KMDa5oK+<br>2MsxtT/TM5a1toGoAAABe0rRIFkAAAQDAEYwRAIgc9G4Kf+YE3BwY4IHwrPfCyx<br>HBfmiWZVYY3syP6PMtQCICbrsrVpPy0eYoa+way7bJOxuRAbwT4Sfd4Nd14ITkrE<br>AHYAfT7y+I//iFVoJMLAyp5SiXkrxQ54CX8uapdomX4i8NcAAAF7StEgegAABAMA<br>RzBFAiBQFP/Jq9s0RX2lddzPRWorKmK2vHb0gBjdeR4DafmJfglhAOGXW8pNxx<br>+g<br>AqkEhlLco7khwJUOAcDgcKCIG9HS9FOJ8MA0GCSqGSIb3DQEBCwUAA4IBAQA

BcKanD<br>DWicj0QRpqqbMyHi1uwU0vJoOuCXAB5TMevD1OGhl2HGfV8blgxWlfOU  
LxGoEd/k<br>SA0t9X94AFhu+FhBQnrssUH/D  
+xQFKSbaZcQIB5j1nEFNdtpokxW5JyhjS5MXiYO<br>dWfgQcEhx3HlnYVT  
+Z1CpNslgCSOaoOawx+b1y1/Kj5ZIEld2rZBIOxh+/ifPP9h<br>Z8Y/kug  
+O2Vu7xfPWn4RtVeG39FpK3FcbOxkTA09K1CHtPQf+KZ/  
Ou5BcLn8BJi5<br>MVXdjQTeKuzilyV7QM+dAxHn3/  
reMXmlGengm7nj19K2P3I7Z70dX1sEwUXQR8Wr<br>/+lfVfwGqqjAuskK<br>-----END  
CERTIFICATE-----<br>-----BEGIN CERTIFICATE-----  
<br>MIIFfjCCAv6gAwIBAgIRAJErCErPDBinU/  
bWLiWnX1owDQYJKoZIhvcNAQELBQAw<br>TzELMAkGA1UEBhMCVVMxKTAnBgN  
VBAoTIEludGVybWV0IFNIY3VyaXR5IFJlc2Vh<br>cmNoIEdyb3VwMRUwEwYDVQQU  
EwYDU1JHIFJvb3QgWDEwHhcNMjAwOTA0MDAwMDAw<br>WhcNMjUwOTE1MTYw  
MDAwWjAyMQswCQYDVQQUGEwJVUzEWMBQGA1UEChMNTGV0J3Mg<br>RW5jcml  
wdDELMAkGA1UEAxMCUjMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK<br>  
AoIBAQC7AhUozPaglNMPEuyNVZLD+ILxmaZ6QoinXSaqtSu5xUyxr45r  
+XXlo9cP<br>R5QUVTVXjJ6oojkZ9YI8QqIObvU7wy7bjcCwXPNZOOftz2nwWgsbvsCU  
JCWH+jdx<br>sxPnHKzhm+/  
b5DtFukWWqcFTzjTIUu61ru2P3mBw4qVUq7ZtDpelQDRrK9O8Zutm<br>NHZ6a4uPVy  
mZ+DAXXbpyb/uBxa3Shlg9F8fnCbvxK/eG3MHacV3URuPMrSXBiLxg<br>Z3Vms/  
EY96Jc5IP/Ooi2R6X/ExjqmAl3P51T+c8B5fWmcBcUr2Ok/5mzk53cU6cG<br>/  
kiFHaFpriV1uxPMUgP17VGhi9sVAgMBAAGjggEIMIIBBDAOBgNVHQ8BAf8EBAMC<br>  
>AYYwHQYDVR0IBBYwFAYIKwYBBQUHAWIGCCsGAQUFBwMBMBIGA1UdEwEB/  
wQIMAYB<br>Af8CAQAwHQYDVR0OBBYEFBQusxe3WFbLrIAJQOYfr52LFMLGMB8  
GA1UdIwQYMBaA<br>FHm0WeZ7tuXkAXOACIjIGlj26ZtuMDIGCCsGAQUFBwEBBCY  
wJDAiBggrBgEFBQcw<br>AoYWaHR0cDovL3gxLmkubGVuY3lub3JnLzAnBgNVHR8EI  
DAeMBygGqAYhhZodHRw<br>Oi8veDEuYy5sZW5jci5vcmcvMCIGA1UdIAQbMBkwCA  
YGZ4EMAQIBMA0GCysGAQQB<br>gt8TAQEBMA0GCSqGSIb3DQEBCwUAA4ICAQ  
CFyk5HPqP3hUSFvNVneLKYY611TR6W<br>PTNlclQtdq+34IL9fzLdwALduO/  
ZelN7klJ+m74uyA+eitRY8kc607TkC53wl<br>ikfmZW4/RvTZ8M6UK  
+5UzhK8jCdLuMGYL6KvzXGRSgi3yLgjewQtCPkIVz6D2QQz<br>CkcheAmCJ8MqyJu  
5zlzyZMjAvnnAT45tRAxekrsu94sQ4egdRCnbWSDtY7kh  
+Blm<br>IJNXoB1IBMEKlq4QDUOXoRgffuDghje1WrG9ML+Hbisq/  
yFOGwXD9RiX8F6sw6W4<br>avAuvDsue5L3sz85K+EC4Y/wFVDNvZo4TYXao6Z0f  
+IQKc0t8DQYzk1OXVu8rp2<br>yJMC6alLbBfODALZvYH7n7do1AZIs4l9d1P4jnkDrQox  
B3UqQ9hVI3LEKQ73xF1O<br>yK5GhDDX8oVfGKF5u  
+declsH4YaTw7mP3GFxJSqv3+0IUfJoi5Lc5da149p90lds<br>hCExroL1+7mrylkXPeF  
M5TgO9r0rvZaBFOvV2z0gp35Z0+L4WPIbuEjN/lxPFin+<br>HIUjr8gRsl3qfJOQFy/  
9rKIJR0Y/8Omwt/  
8oTWgy1mdeHmnmjk7j1nYsvC9JSQ6Zv<br>MldITTKB3zhThV1+XWYp6rjd5JW1zbVW  
EkLNxE7GJThEUG3szgBVGP7pSWTUTsqX<br>nLRbwHOoq7hHwg==<br>-----END  
CERTIFICATE-----<br>-----BEGIN CERTIFICATE-----  
<br>MIIFYDCCBEigAwIBAgIQQAf3ITfU6UK47naqPGQKtzANBgkqhkiG9w0BAQsFAD  
A/  
<br>MSQwlgYDVQQKExtEaWdpdGFsIFNpZ25hdHVyZSBUCnVzdCBDby4xZzAVBgNV  
BAMT<br>DkRTRVCBSb290IENBIFgzMB4XDTIxMDEyMDE5MTQwM1oXDTI0MDkzMD  
E4MTQwM1ow<br>TzELMAkGA1UEBhMCVVMxKTAnBgNVBAoTIEludGVybWV0IFNIY

3VyaXR5IFJlc2Vh<br>cmNoIEdyb3VwMRUwEwYDVQQDEwxJU1JHIFJvb3QgWDEwg  
gliMA0GCSqGSIb3DQEB<br>AUUAA4ICDwAwggIKAoICAQCt6CRz9BQ385ueK1coHl  
e+3LffOJCMbjzmV6B493XC<br>ov71am72AE8o295ohmxEk7axY/0UEmu/  
H9LqMZshftEzPLpI9d1537O4/xLxIZpL<br>wYqGcWIKZmZsj348cL  
+tKSIG8+TA5oCu4kuPt5l+IAOf00eXfJlI1PoOK5PCm  
+D<br>LtFJV4yAdLbaL9A4jXsDcCEbdfIwPPqPrt3aY6vrFk/CjhFLfs8L6P  
+1dy70sntK<br>4EwSJQxwjQMpoOFTJOwT2e4ZvxCzSow/  
iaNhUd6shweU9GNx7C7ib1uYgeGJXDR5<br>bHbvO5BieebbpJovJsXQEOEO3tkQjhb  
7t/eo98flAgeYjzYlIefiN5YNNnWe+w5y<br>sR2bvAP5SQXYgd0FtCrWQemsAXaVCg/  
Y39W9Eh81LygXbNKYwagJZHduRze6zqxZ<br>Xmidf3LWicUGQSk  
+WT7dJvUkyRGnWqNMQB9GoZm1pzpRboY7nn1ypxIFeFntPIF4<br>FQsDj43QLwWy  
PntKHEtzBRL8xurgUBN8Q5N0s8p0544fAQjQMNRbcTa0B7rBMDBc<br>SLeCO5imfW  
CKoqMpgsy6vYMEG6KDA0Gh1gXxG8K28Kh8hjtGqEgqiNx2mna/  
H2ql<br>PRmP6zjzZN7IKw0KKP/32+IVQtQi0Cdd4Xn+GOdwiK1O5tmLOsbdJ1Fu/  
7xk9TND<br>TwIDAQABo4IBRjCCAUIwDwYDVR0TAQH/BAUwAwEB/  
zAOBgNVHQ8BAf8EBAMCAQYw<br>SwYIKwYBBQUHAQEEPzA9MDsGCCsGAQUF  
BzAChi9odHRwOi8vYXBwcy5pZGVudHJ1<br>c3QuY29tL3Jvb3RzL2RzdHJvb3RjYXgz  
LnA3YzAfBgNVHSMEGDAWgBTEp7Gkeyxx<br>  
+tvhS5B1/8QVYIWJEDBUBgNVHSAETTLBMAgGBmeBDAECATA/  
BgSrBgEEAYLfEwEB<br>ATAwMC4GCCsGAQUFBwIBFiJodHRwOi8vY3BzLnJvb3Qte  
DEubGV0c2VuY3J5cHQw<br>b3JnMDwGA1UdHwQ1MDMwMaAvoC2GK2h0dHA6Ly9  
jcmwuaWRlbnRydXN0LmNvbS9E<br>U1RST09UQ0FYM0NSTC5jcmwwHQYDVR0OB  
BYEFHm0WeZ7tuXkAXOACljlGij26Ztu<br>MA0GCSqGSIb3DQEBcWUAA4IBAQAkCw  
Bslm7/DILQrt2M51oGrS+o44+/  
yQoDFVDC<br>5WxCu2+b9LRPwkSICHXM6webFGJueN7sJ7o5XPWioW5WIHAQU7  
G75K/  
QosMrAdSW<br>9MUgNTP52GE24HGNTLi1qoJFlcDyqSMo59ahy2cl2qBDLKobkx/  
J3vWraV0T9VuG<br>WCLKTVXkcGdtwlfFRjIbZ4pYg1htmf5X6DYO8A4jqv2II9DjXA6U  
SbW1FzXSLr9O<br>he8Y4IWS6wY7bCkjCWDcRQJMEhg76fsO3txE  
+FiYruq9RUWhiF1myv4Q6W  
+CyBFC<br>Dfvp7OOGAN6dEOM4+qR9sdjoSYKEBpsr6GtPAQw4dy753ec5<br>-----  
END CERTIFICATE-----<br>-----BEGIN CERTIFICATE-----  
<br>MIIDSjCCAjKgAwIBAgIQRK  
+wgNajJ7qJMDmGLvhAazANBgkqhkiG9w0BAQUFADA/  
<br>MSQwlgYDVQQKEExtEaWdpdGFsIFNpZ25hdHVyZSBUCnVzdCBDby4xZzAVBgNV  
BAMT<br>DkRTRVCBSb290IENBIFgzMB4XDTAwMDkzMDIxMTIxOVVoXDTIxMDkzMDE  
0MDExNVow<br>PzEkMCIGA1UEChMbRGlnaXRhbCBTaWduYXR1cmUgVHJ1c3QgQ  
28uMRcwFQYDVQQD<br>Ew5EU1QgUm9vdCBDQSBYmzCCASlWdQYJKoZIhvcNA  
QEBBQADggEPADCCAQoCggEB<br>AN  
+v6ZdQCINXtMxiZfaQguzH0yxrMMpb7NnDfcdAwRgUi+DoM3ZJKuM/  
IUmTrE4O<br>rz5Iy2Xu/  
NMhD2XSKtkyj4zl93ewEnu1lcCJo6m67XMuegwGMOifooUMM0RoOEQ<br>OLI5CjH9  
UL2AZd  
+3UWODyOKIYepLYYHsUmu5ouJLGiifSKOeDNoJjj4XLh7dIN9b<br>xiaqKqy69cK3FCx  
olkHRYxXtqqzTWMIn/5WgTe1QLyNau7Fqckh49ZLOMxt+/  
yUFw<br>7BZy1SbsOFU5Q9D8/



RhcQPGX69Wam40dutolucbY38EVAjqr2m7xPi71XAicPNaD<br>aeQQmxkqtilX4+U9m  
5/wAlOCAwEAAaNCMEAwDwYDVR0TAQH/BAUwAwEB/  
zAOBgNV<br>HQ8BAf8EBAMCAQYwHQYDVR0OBBYEFMSnsaR7LHH62+FLkHX/  
xBVghYkQMA0GCSqG<br>Slb3DQEBCQUAA4IBAQCjGiybFwBcqR7uKGY3Or  
+Dxz9LwwmglSBd49IZRNI+DT69<br>ikugdB/  
OEIKcdBodfpga3csTS7MgROSR6cz8faXbauX  
+5v3gTt23ADq1cEmv8uXr<br>AvHRAosZy5Q6XkjEGB5YGV8eAlrwDPGxrancWYaLbu  
mR9YbK  
+rlmM6pZW87ipxZz<br>R8srzJmwN0jP41ZL9c8PDHlyh8bwRLtTcm1D9SZlmlJnt1ir/  
md2cXjbDaJWFBM5<br>JDGFoqgCWjBH4d1QB7wCCZAA62RjYJsWvljJEubSfZGL  
+T0yjWW06XyxV3bqxbYo<br>Ob8VZRzI9neWagqNdwvYkQsEjgfbKbYK7p2CNTUQ<b  
r>-----END CERTIFICATE-----<br></span></p><p class="debugging-content"><span  
class="debugging-title">Certificate Transparency: </span><span class="debugging-  
content"><br><br>SCT Cloudflare 'Nimbus2021' Log (Embedded in certificate,  
Verified)<br><br>SCT Google 'Xenon2021' log (Embedded in certificate, Verified)</  
span></p></div>  
</div>

-----  
<div id="error-code" class="error-code" role="button" aria-  
expanded="false">net::ERR\_CERT\_DATE\_INVALID</div>

<div id="error-debugging-info" class="hidden"><p class="debugging-content"><span  
class="debugging-title">Subject: </span><span class="debugging-  
content">accounts.shopify.com</span></p><p class="debugging-content"><span  
class="debugging-title">Issuer: </span><span class="debugging-content">R3</span></  
p><p class="debugging-content"><span class="debugging-title">Expires on: </  
span><span class="debugging-content">Nov 13, 2021</span></p><p  
class="debugging-content"><span class="debugging-title">Current date: </span><span  
class="debugging-content">Oct 5, 2021</span></p><p class="debugging-  
content"><span class="debugging-title">PEM encoded chain: </span><span  
class="debugging-content debugging-content-fixed-width">-----BEGIN  
CERTIFICATE-----<br>MIIF2DCCBMCGAwIBAgISA3IP  
+wuRtzmomjRFbuWpyrsCMA0GCSqGSIb3DQEBCwUA<br>MDIx CzAJBgNVBAYTAIV  
TMRywFAYDVQQKEw1MZXQncyBFbmNyeXB0MQswCQYDVQQD<br>EwJSMzAeFw  
0yMTA4MTUxNjE3MDNaFw0yMTE5MTMxNjE3MDFaMBA8xHTAbBgNVBAMT<br>FGFj  
Y291bnRzLnNob3BpZnkuY29tMIIlBjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB<br>CgK  
CAQEAmNO6kENQqqUM0WgozXecZGcPzrlyEbrU/  
N7mdStXBNjXN9WAEbLSTGrw<br>gkyShj1dbDXVJudJ7YdQXJr/  
KJbm87u8a1HEZI4Dmj1FxBIAAd+Gs1RsFdQd0nEw<br>5Zgi0Sojj6d87SEqpS2i0Vz6V  
+hTrbuy+XXP5+J8piNiRp2lkrZQkWKaifZaLyvH<br>w2S2OFtEJxH3SSoKfsA/  
ST4ZPRDFIppsXgmpFf7sU5u1E3XckNjxl220Q29/

vtls<br>ErLP9noTy1wdKzXJYbu4IT6nVOSs  
+TGLQBcBCfNQyaG19jMrL299G4QyNy2h6uWX<br>MwPm/Opq3qk3Bnl5uZgyV  
+mx7EMe1QIDAQABo4IC+TCCAvUwDgYDVR0PAQH/  
BAQD<br>AgWgMB0GA1UdJQQWMBQGCCsGAQUFBwMBBggrBgEFBQcDAjAMBg  
NVHRMBAf8EAjAA<br>MB0GA1UdDgQWBRR/TRz5xF8Ma/  
wcElldjg8fyl3L3zAfBgNVHSMEGDAWgBQUULrMX<br>t1hWy65QCUDmH6+dixTCxjBVB  
ggrBgEFBQcBAQRJMEcwIQYIKwYBBQUHMAAGGFWh0<br>dHA6Ly9yMy5vLmxlbmN  
yLm9yZzAiBggrBgEFBQcwAoYWAHR0cDovL3IzLmkuYVU<br>Y3lub3JnLzCBYQYDV  
R0RBIHBMIG  
+gh5hY2NvdW50cy1kZW1vLnNob3BpZnlibG91<br>ZC5jb22CFGFjY291bnRzLnNob3B  
pZnkuY29tgh9jdXN0b21lcnMtZGVtby5zaG9w<br>aWZ5Y2xvdWQuY29tgiZpZGVudGlo  
eS1tYWluLWNsdXN0ZXIuc2hvcGlmeWNSb3Vk<br>LmNvbYliaWRlbnRpdHktbWVyY2h  
hbnQuc2hvcGlmeWNSb3VkLmNvbYlZaWRlbnRp<br>dHkuc2hvcGlmeWNSb3VkLmNv  
bTBMBgNVHSAERTBDMAgGBmeBDAECATA3BgsrBgEE<br>AYLfEwEBATAoMCYGC  
CsGAQUFBwIBFhpodHRwOi8vY3BzLmxdHNlbnNyeXB0Lm9y<br>ZzCCAQMGCisGA  
QQB1nkCBAIEgfQEgfEA7wB1AESUZS6w7s6vxEAH2Kj+KMDa5oK+<br>2MsxtT/  
TM5a1toGoAAABe0rRIFkAAQDAEYwRAIgcU9G4Kf  
+YE3BwY4IHwrPfcYx<br>HBfmiWZVYY3syP6PMtQCICbrsrVpY0eYoa  
+way7bJOXuRAbwT4Sfd4Nd14ITkrE<br>AHYAfT7y+I//  
iFVoJMLAyp5SiXkrxQ54CX8uapdomX4i8NcAAAF7StEgegAABAMA<br>RzBFAiBQFP/  
Jq9s0RX2lddzPRWorKmK2vHb0gBjdeR4DafmJfglhAOGXW8pNxh  
+q<br>AgkEhlLco7khWJUOAcDqcKCl9HS9FOJ8MA0GCSqGSib3DQEBCwUAA4IBAQ  
BcKanD<br>DWicj0QRpqqbMyHi1uwU0vJoOuCXAB5TMevD1OGhl2HGfV8blgxWlfOU  
LxGoEd/k<br>SA0t9X94AFhu+FhBQnrssUH/D  
+xQFKSbaZcQIB5j1nEFNdtpokxW5JyhjS5MXiYO<br>dWfgQcEhx3HlnYVT  
+Z1CpNslgCSOaoOawx+b1y1/Kj5ZIEld2rZBIOxh+/ifPP9h<br>Z8Y/kug  
+O2Vu7xfPWn4RtVeG39FpK3FcbOxkTA09K1CHtPQf+KZ/  
Ou5BcLn8BJi5<br>MVXdjQTeKuzilyV7QM+dAxHn3/  
reMXmlGengm7nj9K2P3I7Z70dX1sEwUXQR8Wr<br>+/lfVfwGqqjAuskK<br>-----END  
CERTIFICATE-----<br>-----BEGIN CERTIFICATE-----  
<br>MIIFFjCCAv6gAwIBAgIRAJErCErPDBinU/  
bWLiWnX1owDQYJKoZIhvcNAQELBQAw<br>TzELMAkGA1UEBhMCVVMxKTAnBgN  
VBAoTIEludGVybmV0IFNIY3VyaXR5IFJlc2Vh<br>cmNoIEdyb3VwMRUwEwYDVQQD  
EwxJU1JHIFJvb3QgWDEwHhcNMjAwMDAwMDAw<br>WhcNMjUwOTE1MTYw  
MDAwWjAyMQswCQYDVQGEwJVUzEWMBQGA1UEChMNTGV0J3Mg<br>RW5jcml  
wdDELMAkGA1UEAxMCUjMwggEiMA0GCSqGSib3DQEBAQUAA4IBDwAwggEK<br>AoIBAQC7AhUozPaglNMPEuyNVZLD+ILxmaZ6QoinXSaqtSu5xUyxr45r  
+XXlo9cP<br>R5QUVTVXjJ6oojkZ9YI8QqlObvU7wy7bjcCwXPNZOOftz2nwWgsbvsCU  
JCWH+jdx<br>sxPnHKzhm+/  
b5DtFukWWqcFTzjTIUu61ru2P3mBw4qVUq7ZtDpelQDRrK9O8Zutm<br>NHZ6a4uPVy  
mZ+DAXXbpyb/uBxa3Shlg9F8fnCbvxK/eG3MHacV3URuPMrSXBiLxg<br>Z3Vms/  
EY96Jc5IP/Ooi2R6X/ExjqmAI3P51T+c8B5fWmcBcUr2Ok/5mzk53cU6cG<br>/  
kiFHaFpriV1uxPMUgP17VGhi9sVAgMBAAGjggEIMIIBBDAOBgNVHQ8BAf8EBAMC<br>AYYwHQYDVR0IBBYwFAYIKwYBBQUHAWIGCCsGAQUFBwMBMBIGA1UdEwEB/  
wQIMAYB<br>Af8CAQAwHQYDVR0OBBYEFBQusxe3WFbLrIAJQOYfr52LFMLGMB8  
GA1UdIwQYMBAA<br>FHm0WeZ7tuXkAXOACIjIGlj26ZtuMDIGCCsGAQUFBwEBBCY

wJDAiBggrBgEFBQcw<br>AoYWaHR0cDovL3gxLmkubGVuY3lub3JnLzAnBgNVHR8E  
DAeMBygGqAYhhZodHRw<br>Oi8veDEuYy5sZW5jci5vcmcvMCIGA1UdIAQbMBkwCA  
YGZ4EMAQIBMA0GCysGAQQB<br>gt8TAQEBMA0GCSqGSIlb3DQEBcWUAA4ICAQ  
CFyk5HPqP3hUSFvNVneLKYY611TR6W<br>PTNlclQtagDqw+34IL9fzLdwALduO/  
ZelN7klJ+m74uyA+eitRY8kc607TkC53wl<br>ikfmZW4/RvTZ8M6UK  
+5UzhK8jCdLuMGYL6KvzXGRSgi3yLgjewQtCPkIVz6D2QQz<br>CkcheAmCJ8MqyJu  
5zlzyZMjAvnnAT45tRAxekrsu94sQ4egdRCnbWSDtY7kh  
+Blm<br>IJNXoB1IBMEKlq4QDUOXoRgffuDghje1WrG9ML+Hbisq/  
yFOGwXD9RiX8F6sw6W4<br>avAuvDsue5L3sz85K+EC4Y/wFVDNVzo4TYXao6Z0f  
+lQKc0t8DQYzk1OXVu8rp2<br>yJMC6alLbBfODALZvYH7n7do1AZIs4I9d1P4jnkDrQox  
B3UqQ9hVl3LEKQ73xF1O<br>yK5GhDDX8oVfGKF5u  
+declsH4YaTw7mP3GFxJSqv3+0lUFJoi5Lc5da149p90lds<br>hCExroL1+7mrylkXPeF  
M5TgO9r0rvZaBFOvV2z0gp35Z0+L4WPIbuEjN/lxPFin+<br>HIUjr8gRsl3qfJOQFy/  
9rKIJR0Y/8OmwT/  
8oTWgy1mdeHmnmjk7j1nYsvC9JSQ6Zv<br>MldITTKB3zhThV1+XWYp6rjd5JW1zbVW  
EkLNxE7GJThEUG3szgBVGP7pSWTUTsqX<br>nLRbwHOoq7hHwg==<br>-----END  
CERTIFICATE-----<br>-----BEGIN CERTIFICATE-----  
<br>MIIFYDCCBEigAwIBAgIQQAF3ITfU6UK47naqPGQKtzANBgkqhkiG9w0BAQsFAD  
A/  
<br>MSQwlgYDVQQKEExtEaWdpdGFsIFNpZ25hdHVyZSBUcnVzdCBDby4xFzAVBgNV  
BAMT<br>DkRtVCBSb290IENBIFgzMB4XDTIxMDEyMDE5MTQwM1oXDTI0MDkzMD  
E4MTQwM1ow<br>TzELMAkGA1UEBhMCVVMxKTAnBgNVBAoTIEludGVybmV0IFNIY  
3VyaXR5IFJlc2Vh<br>cmNoIEdyb3VwMRUwEwYDVQQDEWxJU1JHIFJvb3QgWDEwg  
gliMA0GCSqGSIlb3DQEB<br>AQUAA4ICDwAwggIKAoICAQCt6CRz9BQ385ueK1coHI  
e+3LffOJCMbjzmV6B493XC<br>ov71am72AE8o295ohmxEk7axY/0UEmu/  
H9LqMZshftEzPLpl9d1537O4/xLxIZpL<br>wYqGcWIKZmZsj348cL  
+tKSIG8+TA5oCu4kuPt5l+IAOf00eXfJlII1PoOK5PCm  
+D<br>LtFJV4yAdLbaL9A4jXsDcCEbdfIwPPqPrt3aY6vrFk/CjhFLfs8L6P  
+1dy70sntK<br>4EwSJQxwjQMpoOFTJOwT2e4ZvxCzSow/  
iaNhUd6shweU9GNx7C7ib1uYgeGJXDR5<br>bHbvO5BieebbpJovJsXQEOEO3tkQjhb  
7t/eo98flAgeYjzYlIefiN5YNNnWe+w5y<br>sR2bvAP5SQXYgd0FtCrWQemsAXaVCg/  
Y39W9Eh81LygXbNKYwagJZHduRze6zqxZ<br>Xmidf3LWicUGQSk  
+WT7dJvUkyRGnWqNMQB9GoZm1pzpRboY7nn1ypxIFeFntPIF4<br>FQsDj43QLwWy  
PntKHEtzBRL8xurgUBN8Q5N0s8p0544fAQjQMNRbcTa0B7rBMDBC<br>SLeCO5imfW  
CKoqMpgsy6vYMEG6KDA0Gh1gXxG8K28Kh8hjtGqEgqiNx2mna/  
H2ql<br>PRmP6zjzZN7IKw0KKP/32+IVQtQi0Cdd4Xn+GOdwiK1O5tmLOsbdJ1Fu/  
7xk9TND<br>TwIDAQABo4IBRjCCAUIwDwYDVR0TAQH/BAUwAwEB/  
zaOBgNVHQ8BAf8EBAMCAQYw<br>SwYIKwYBBQUHAQEEPzA9MDsGCCsGAQUF  
BzAChi9odHRwOi8vYXBwcy5pZGVudHJ1<br>c3QuY29tL3Jvb3RzL2RzdHJvb3RjYXgz  
LnA3YzAfBgNVHSMEGDAWgBTEp7Gkeyxx<br>  
+tvhS5B1/8QVYIWJEDBUBgNVHSAETTLBMAgGBmeBDAECATA/  
BgSrBgEEAYLfEwEB<br>ATAwMC4GCCsGAQUFBwIBFiJodHRwOi8vY3BzLnJvb3Qte  
DEubGV0c2VuY3J5cHQ<br>b3JnMDwGA1UdHwQ1MDMwMaAvoC2GK2h0dHA6Ly9  
jcmwuaWRlbnRydXN0LmNvbS9E<br>U1RST09UQ0FYM0NSTC5jcmwwHQYDVR0OB  
BYEFHm0WeZ7tuXkAXOACljlGij26Ztu<br>MA0GCSqGSIlb3DQEBcWUAA4IBAQAkCw  
Bslm7/DILQrt2M51oGrS+o44+/

yQoDFVDC<br>5WxCu2+b9LRPwkSICHXM6webFGJueN7sJ7o5XPWioW5WIHAQU7  
G75K/  
QosMrAdSW<br>9MUgNTP52GE24HGNTLi1qoJFlcDyqSMo59ahy2cl2qBDLKobkx/  
J3vWraV0T9VuG<br>WCLKTVXkcGdtwlfFRjIBz4pYg1htmf5X6DYO8A4jqv2lI9DjXA6U  
SbW1FzXSLr9O<br>he8Y4IWS6wY7bCkjcWDcRQJMEhg76fsO3txE  
+FiYruq9RUWhiF1myv4Q6W  
+CyBFC<br>Dfvp7OOGAN6dEOM4+qR9sdjoSYKEBpsr6GtPAQw4dy753ec5<br>-----  
END CERTIFICATE-----<br>-----BEGIN CERTIFICATE-----  
<br>MIIDSjCCAjKgAwIBAgIQRK  
+wgNajJ7qJMDmGLvhAazANBqkqhkiG9w0BAQUFADA/  
<br>MSQwlgYDVQQKEExtEaWdpdGFsIFNpZ25hdHVyZSBUCnVzdCBDby4xFzAVBgNV  
BAMT<br>DkRtVCBSb290IENBIFgzMB4XDTAwMDkzMDIxMTIxOVoXDTIxMDkzMDE  
0MDExNVow<br>PzEkMCIGA1UEChMbRGlnaXRhbCBTaWduYXR1cmUgVHJ1c3QgQ  
28uMRcwFQYDVQQD<br>Ew5EU1QgUm9vdCBDQSBYMzCCASlwdQYJKoZIhvcNA  
QEBBQADggEPADCCAQoCggEB<br>AN  
+v6ZdQCINXtMxiZfaQguzH0yxrMMpb7NnDfcdAwRgUi+DoM3ZJKuM/  
IUmTrE4O<br>rz5ly2Xu/  
NMhD2XSKtkyj4zl93ewEnu1lcCJo6m67XMuegwGMOifooUMM0RoOEQ<br>OLI5CjH9  
UL2AZd  
+3UWODyOKIYepLYYHsUmu5ouJLGiifSKOeDNoJjj4XLh7dIN9b<br>xiqKqy69cK3FCx  
olkHryxXtqqzTWMIIn/5WgTe1QLyNau7Fqckh49ZLOMxt+/  
yUFw<br>7BZy1SbsOFU5Q9D8/  
RhcQPGX69Wam40dutolucbY38EVAjqr2m7xPi71XAicPNaD<br>aeQQmxkqtilX4+U9m  
5/wAl0CAwEAaAaNCMEAwDwYDVR0TAQH/BAUwAwEB/  
zAOBgNV<br>HQ8BAf8EBAMCAQYwHQYDVROBBYEFMSnsaR7LHH62+FLkHX/  
xBVghYkQMA0GCSqG<br>Slb3DQEBBQUAA4IBAQCjGiybFwBcqR7uKGY3Or  
+Dxz9LwwmglSBd49IZRNI+DT69<br>ikugdB/  
OEIKcdBodfpga3csTS7MgROSR6cz8faXbauX  
+5v3gTt23ADq1cEmv8uXr<br>AvHRAosZy5Q6XkjEGB5YGV8eAlrwDPGxrancWYaLbu  
mR9YbK  
+rlmM6pZW87ipxZz<br>R8srzJmwN0jP41ZL9c8PDHlyh8bwRLtTcm1D9SZlmlJnt1ir/  
md2cXjbDaJWFBM5<br>JDGfOggCWjBH4d1QB7wCCZAA62RjYJsWvIjJEubSfZGL  
+T0yjWW06XyxV3bqxbYo<br>Ob8VZRzI9neWagqNdwwYkQsEjgfbKbYK7p2CNTUQ<b  
r>-----END CERTIFICATE-----<br></span></p><p class="debugging-content"><span  
class="debugging-title">Certificate Transparency: </span><span class="debugging-  
content"><br><br>SCT Cloudflare 'Nimbus2021' Log (Embedded in certificate,  
Verified)<br><br>SCT Google 'Xenon2021' log (Embedded in certificate, Verified)</  
span></p></div>

-----

<div id="recurrent-error-message" class="hidden">  
Warnings may be common while websites update their security. This should  
improve soon.

```

</div>
<div id="extended-reporting-opt-in" class="hidden">
  <label>
    <div class="checkboxes">
      <input type="checkbox" id="opt-in-checkbox">
      <span class="checkbox"></span>
    </div>
    <span id="opt-in-label">Help improve security on the web for everyone by
sending <a href="#" id="whitepaper-link">URLs of some pages you visit, limited system
information, and some page content</a> to Google. <a id="privacy-link"
href="#">Privacy policy</a></span>
  </label>
</div>

```

```

<span id="opt-in-label">Help improve security on the web for everyone by sending <a
href="#" id="whitepaper-link">URLs of some pages you visit, limited system information,
and some page content</a> to Google. <a id="privacy-link" href="#">Privacy policy</
a></span>

```

```

<div id="enhanced-protection-message" class="hidden">
  <label>
    <div class="icon"></div>
    <div>To get Chrome's highest level of security, <a href="#" id="enhanced-
protection-link">turn on enhanced protection</a></div>
  </label>
</div>

```

-----

```

/* Copyright 2014 The Chromium Authors. All rights reserved.
 * Use of this source code is governed by a BSD-style license that can be
 * found in the LICENSE file. */

/* This file is dynamically processed by a C++ data source handler to fill in
 * some per-platform/locale styles that dramatically alter the page. This is
 * done to reduce flicker, as JS may not run before the page is rendered.
 *
 * There are two ways to include this stylesheet:
 * 1. via its chrome://resources/ URL in HTML, i.e.:
 *
 * <link rel="stylesheet" href="chrome://resources/css/text_defaults.css">
 *
 * 2. via the webui::AppendWebUICSSTextDefaults() method to directly append it
 * to an HTML string.

```

```
* Otherwise its placeholders won't be expanded. */
```

```
body {  
  font-family: system-ui, sans-serif;  
  font-size: 75%;  
}
```

```
button {  
  font-family: system-ui, sans-serif;  
}
```

-----

```
// Copyright (c) 2012 The Chromium Authors. All rights reserved.  
// Use of this source code is governed by a BSD-style license that can be  
// found in the LICENSE file.
```

```
/**  
 * @fileoverview This file defines a singleton which provides access to all data  
 * that is available as soon as the page's resources are loaded (before DOM  
 * content has finished loading). This data includes both localized strings and  
 * any data that is important to have ready from a very early stage (e.g. things  
 * that must be displayed right away).  
 *  
 * Note that loadTimeData is not guaranteed to be consistent between page  
 * refreshes (https://crbug.com/740629) and should not contain values that might  
 * change if the page is re-opened later.  
 */
```

```
/** @type {!LoadTimeData} */  
// eslint-disable-next-line no-var  
var loadTimeData;
```

```
class LoadTimeData {  
  constructor() {  
    /** @type {?Object} */  
    this.data_ = null;  
  }  
}
```

```
/**  
 * Sets the backing object.  
 *  
 * Note that there is no getter for |data_| to discourage abuse of the form:  
 *
```

```

*   var value = loadTimeData.data()['key'];
*
* @param {Object} value The de-serialized page data.
*/
set data(value) {
  expect(!this.data_, 'Re-setting data. ');
  this.data_ = value;
}

/**
* @param {string} id An ID of a value that might exist.
* @return {boolean} True if id is a key in the dictionary.
*/
valueExists(id) {
  return id in this.data_;
}

/**
* Fetches a value, expecting that it exists.
* @param {string} id The key that identifies the desired value.
* @return {*} The corresponding value.
*/
getValue(id) {
  expect(this.data_, 'No data. Did you remember to include strings.js?');
  const value = this.data_[id];
  expect(typeof value !== 'undefined', 'Could not find value for ' + id);
  return value;
}

/**
* As above, but also makes sure that the value is a string.
* @param {string} id The key that identifies the desired string.
* @return {string} The corresponding string value.
*/
getString(id) {
  const value = this.getValue(id);
  expectIsType(id, value, 'string');
  return /** @type {string} */ (value);
}

/**
* Returns a formatted localized string where $1 to $9 are replaced by the
* second to the tenth argument.
* @param {string} id The ID of the string we want.
* @param {...(string|number)} var_args The extra values to include in the
* formatted output.

```

```

* @return {string} The formatted string.
*/
getStringF(id, var_args) {
  const value = this.getString(id);
  if (!value) {
    return "";
  }

  const args = Array.prototype.slice.call(arguments);
  args[0] = value;
  return this.substituteString.apply(this, args);
}

/**
 * Returns a formatted localized string where $1 to $9 are replaced by the
 * second to the tenth argument. Any standalone $ signs must be escaped as
 * $$$.
 * @param {string} label The label to substitute through.
 *   This is not an resource ID.
 * @param {...(string|number)} var_args The extra values to include in the
 *   formatted output.
 * @return {string} The formatted string.
 */
substituteString(label, var_args) {
  const varArgs = arguments;
  return label.replace(/\$(.|\$|\n)/g, function(m) {
    expect(m.match(/^\$[$1-9]/), 'Unescaped $ found in localized string.');
```

return m === '\$\$' ? '\$' : varArgs[m[1]];

```

  });
}

/**
 * Returns a formatted string where $1 to $9 are replaced by the second to
 * tenth argument, split apart into a list of pieces describing how the
 * substitution was performed. Any standalone $ signs must be escaped as $$.
 * @param {string} label A localized string to substitute through.
 *   This is not an resource ID.
 * @param {...(string|number)} var_args The extra values to include in the
 *   formatted output.
 * @return {!Array<{value: string, arg: (null|string)}>} The formatted
 *   string pieces.
 */
getSubstitutedStringPieces(label, var_args) {
  const varArgs = arguments;
  // Split the string by separately matching all occurrences of $1-9 and of
  // non $1-9 pieces.
```



```

const pieces = (label.match(/(\$(1-9))|((^\$)\$(^1-9|$\$))+)/g) ||
    [].map(function(p) {
        // Pieces that are not $1-9 should be returned after replacing $$
        // with $.
        if (!p.match(/^\$(1-9)\$/)) {
            expect(
                (p.match(/^\$/g) || []).length % 2 === 0,
                'Unescaped $ found in localized string.');
            return {value: p.replace(/^\$\$/g, '$'), arg: null};
        }

        // Otherwise, return the substitution value.
        return {value: varArgs[p[1]], arg: p};
    });

return pieces;
}

/**
 * As above, but also makes sure that the value is a boolean.
 * @param {string} id The key that identifies the desired boolean.
 * @return {boolean} The corresponding boolean value.
 */
getBoolean(id) {
    const value = this.getValue(id);
    expectIsType(id, value, 'boolean');
    return /** @type {boolean} */ (value);
}

/**
 * As above, but also makes sure that the value is an integer.
 * @param {string} id The key that identifies the desired number.
 * @return {number} The corresponding number value.
 */
getInteger(id) {
    const value = this.getValue(id);
    expectIsType(id, value, 'number');
    expect(value === Math.floor(value), 'Number isn\'t integer: ' + value);
    return /** @type {number} */ (value);
}

/**
 * Override values in loadTimeData with the values found in Ireplacementsl.
 * @param {Object} replacements The dictionary object of keys to replace.
 */
overrideValues(replacements) {

```

```

    expect(
        typeof replacements === 'object',
        'Replacements must be a dictionary object.');
```

for (const key in replacements) {

```

    this.data_[key] = replacements[key];
}
}
```

/\*\*

- \* Reset loadTimeData's data. Should only be used in tests.
- \* @param {?Object} newData The data to restore to, when null restores to
- \* unset state.

\*/

```

resetForTesting(newData = null) {
    this.data_ = newData;
}

/**
 * @return {boolean} Whether loadTimeData.data has been set.
 */
isInitialized() {
    return this.data_ !== null;
}
}
```

/\*\*

- \* Checks condition, throws error message if expectation fails.
- \* @param {\*} condition The condition to check for truthiness.
- \* @param {string} message The message to display if the check fails.

\*/

```

function expect(condition, message) {
    if (!condition) {
        throw new Error(
            'Unexpected condition on ' + document.location.href + ': ' + message);
    }
}

/**
 * Checks that the given value has the given type.
 * @param {string} id The id of the value (only used for error message).
 * @param {*} value The value to check the type on.
 * @param {string} type The type we expect lvalue to be.
 */
function expectIsType(id, value, type) {
    expect(
        typeof value === type, '[' + value + ']' (' + id + ') is not a ' + type);
}
```

```
}
```

```
expect(!loadTimeData, 'should only include this file once');  
loadTimeData = new LoadTimeData;
```

```
// Expose lloadTimeData directly on lwindowl, since within a JS module the  
// scope is local and not all files have been updated to import the exported  
// lloadTimeData explicitly.  
window.loadTimeData = loadTimeData;
```

```
console.warn('crbug/1173575, non-JS module files deprecated.');
```

-----

```
loadTimeData.data = {"bad_clock":false,"closeDetails":"Hide advanced","ct":"\n\nSCT  
Cloudflare 'Nimbus2021' Log (Embedded in certificate, Verified)\n\nSCT Google  
'Xenon2021' log (Embedded in certificate, Verified)","currentDate":"Oct 5,  
2021","displayEnhancedProtectionMessage":false,"displaycheckbox":false,"enhancedPr  
otectionMessage":"To get Chrome's highest level of security, \u003Ca href=\"#\n id=  
\"enhanced-protection-link\">turn on enhanced protection\u003C/  
a>","errorCode":"net::ERR_CERT_DATE_INVALID","errorType":1,"expirationDate":"Nov  
13, 2021","explanationParagraph":"accounts.shopify.com normally uses encryption to  
protect your information. When Chrome tried to connect to accounts.shopify.com this  
time, the website sent back unusual and incorrect credentials. This may happen when  
an attacker is trying to pretend to be accounts.shopify.com, or a Wi-Fi sign-in screen  
has interrupted the connection. Your information is still secure because Chrome stopped  
the connection before any data was exchanged.","finalParagraph":"You cannot visit  
accounts.shopify.com right now because the website uses HSTS. Network errors and  
attacks are usually temporary, so this page will probably work  
later.","fontfamily":"system-ui, sans-serif","fontsize":"75%","heading":"Your connection is  
not  
private","hide_primary_button":false,"issuer":"R3","language":"en","openDetails":"Advan  
ced","optInLink":"Help improve security on the web for everyone by sending \u003Ca  
href=\"#\n id=\"whitepaper-link\">URLs of some pages you visit, limited system  
information, and some page content\u003C/a> to Google. \u003Ca id=\"privacy-link\"  
href=\"#\n">Privacy policy\u003C/a>","overridable":false,"pem":"-----BEGIN  
CERTIFICATE-----\nMIIF2DCCBMCGAwIBAgISA3IP  
+wuRtzmomjRFbuWpyrsCMA0GCSqGSIb3DQEBCwUA  
\nMDIx CzAJBgNVBAYTAIVTMRYwFAYDVQQKEw1MZXQncyBFbmNyeXB0MQswCQY  
DVQQD  
\nEwJSMzAeFw0yMTA4MTUxNjE3MDNaFw0yMTExMTMxNjE3MDFaMB8xHTAbBgNV  
BAMT  
\nFGFjY291bnRzLnNob3BpZnkuY29tMIIlBlJANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB  
\nCgKCAQEAmNO6kENQqqUM0WgozXecZGcPzrlyEbrU/
```

N7mdStXBNjXN9WAEbLSTGrw\ngkyShj1dbDXVJudJ7YdQXJr/  
KJbm87u8a1HEZI4Dmj1FxBIAd+Gs1RsFdQd0nEw\n5Zgi0Sojj6d87SEqpS2i0Vz6V  
+hTrbuy+XXP5+J8piNiRp2lkrZQkWKaifZaLyvH\nnw2S2OFtEJxH3SSoKfsA/  
ST4ZPRDFIppsXgmpFf7sU5u1E3XckNjxl220Q29/vtIs  
\nErLP9noTy1wdKzXJYbu4IT6nVOSS+TGLQBcBCfNQyaG19jMrL299G4QyNy2h6uWX  
\nMwPm/Opq3qk3Bnl5uZgyV+mx7EMe1QIDAQABo4IC+TCCAvUwDgYDVR0PAQH/  
BAQD  
\nAgWgMB0GA1UdJQQWMBQGCCsGAQUFBwMBBggrBgEFBQcDAjAMBgNVHRMB  
Af8EAjAA\n\nMB0GA1UdDgQWBBR/TRz5xF8Ma/  
wcElldjg8fyl3L3zAfBgNVHSMEGDAWgBQUlRmX  
\nt1hWy65QCUDmH6+dixTCxjBVBggrBgEFBQcBAQRJMEcwIQYIKwYBBQUHMAGGF  
Wh0\ndHA6Ly9yMy5vLmxlbmNyLm9yZzAiBggrBgEFBQcwAoYWaHR0cDovL3lzLmkub  
GVu\n\nY3lub3JnLzCBYQYDVR0RBIHBMIG  
+gh5hY2NvdW50cy1kZW1vLnNob3BpZnlibG91\n\nZC5jb22CFGFjY291bnRzLnNob3BpZ  
nkuY29tgh9jdXN0b21lcnMtZGVtby5zaG9w  
\naWZ5Y2xvdWQuY29tgiZpZGVudGl0eS1tYWluLWNsdXN0ZXluc2hvcGlmeWNSb3Vk  
\nLmNvbYliaWRIbnRpdHktbWVyY2hhbnQuc2hvcGlmeWNSb3VkLmNvbYIZaWRIbnRp  
\ndHkuc2hvcGlmeWNSb3VkLmNvbTBMBgNVHSAERTBDMAgGBmeBDAECATA3Bgsr  
BgEE  
\nAYLfEwEBATAoMCYGCCsGAQUFBwIBFhpodHRwOi8vY3BzLmxldHNlbnNyeXB0L  
m9y\n\nZzCCAQMGCisGAQQB1nkCBAIEgfQEgfEA7wB1AESUZS6w7s6vxEAH2Kj  
+KMDa5oK+\n\n2MsxtT/TM5a1toGoAAABe0rRIFkAAQDAEYwRAIgcU9G4Kf  
+YE3BwY4IHwrPfCyx\n\nHBfmiWZVYY3syP6PMtQCICbrsrVPy0eYoa  
+way7bJOXuRAbwT4Sfd4Nd14ITkrE\n\nnAHYAfT7y+I//  
iFVoJMLAyp5SiXkrxQ54CX8uapdomX4i8NcAAAF7StEgegAABAMA\n\nRzBFAiBQFP/  
Jq9s0RX2lddzPRWorKmK2vHb0gBjdeR4DafmJfglhAOGXW8pNxh+q  
\nAgkEhILco7khwJUOAcDqcKCI9HS9FOJ8MA0GCSqGSIb3DQEBCwUAA4IBAQBcKa  
nD\n\nNDWicj0QRpqqbMyHi1uwU0vJoOuCXAB5TMevD1OGhl2HGfV8blgxWlfoULxGoEd/  
k\n\nSA0t9X94AFhu+FhBQnrssUH/D+xQFKSbaZcQIB5j1nEFNdtpokxW5JyhjS5MXiYO  
\ndWfgQcEhx3HlnYVT+Z1CpNslgCSOaoOawx+b1y1/Kj5ZIEld2rZBIOxh+/ifPP9h\n\nZ8Y/  
kug+O2Vu7xfPWn4RtVeG39FpK3FcbOxkTA09K1CHtPQf+KZ/  
Ou5BcLn8Bji5\n\nnMVXdjQTeKuzilyV7QM+dAxHn3/  
reMXmlGengm7nj9K2P3I7Z70dX1sEwUXQR8Wr\n\nn/+lfVfwGqqjAuskK\n\n-----END  
CERTIFICATE-----\n\n-----BEGIN CERTIFICATE-----  
\nMIIFFjCCAv6gAwIBAgIRAJErCERPDBinU/bWLiWnX1owDQYJKoZIhvcNAQELBQAw  
\nTzELMAkGA1UEBhMCVVMxKTAnBgNVBAoTIEludGVybmV0IFNIY3VyaXR5IFJlc2Vh  
\ncmNoIEdyb3VwMRUwEwYDVRQQDEwXJU1JHIFJvb3QgWDEwHhcNMjAwOTA0MDAwMDAw  
wMDAw  
\nWhcNMjUwOTE1MTYwMDAwWjAQMjswCQYDVRQQGEwJVUzEWMBQGA1UEChM  
NTGV0J3Mg  
\nRW5jcnlwdDELMAkGA1UEAxMCUjMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAw  
ggEK\n\nAoIBAQC7AhUozPaglNMPEuyNVZLD+ILxmaZ6QoinXSaqTSu5xUyxr45r  
+XXlo9cP  
\nR5QUVTvXjJ6oojkZ9YI8QqlObvU7wy7bjcCwXPNZOOftz2nwWgsbvsCUJCWH+jdx  
\nsxPnHKzhm+/b5DtFUKWWqcFTzjTIUu61ru2P3mBw4qVUq7ZtDpelQDRrK9O8Zutm  
\nNHZ6a4uPVymZ+DAXXbpyb/uBxa3Shlg9F8fnCbvxK/eG3MHacV3URuPMrSXBilXg

\nZ3Vms/EY96Jc5IP/Ooi2R6X/ExjqmA13P51T+c8B5fWmcBcUr2Ok/5mzk53cU6cG\n/  
kiFHaFpriV1uxPMUgP17VGhi9sVAgMBAAGjggEIMIIBBDAOBgNVHQ8BAf8EBAMC  
\nAYYwHQYDVR0IBBYwFAYIKwYBBQUHAWIGCCsGAQUFBwMBMBIGA1UdEwEB/  
wQIMAYB  
\nAf8CAQAwHQYDVR0OBBYEFBQusxe3WFbLrIAJQOYfr52LFMLGMB8GA1UdIwQY  
MBaA  
\nFHm0WeZ7tuXkAXOACljlGlj26ZtuMDIGCCsGAQUFBwEBBCYwJDAiBggrBgEFBQc  
w  
\nAoYWaHR0cDovL3gxLmkubGVuY3lub3JnLzAnBgNVHR8EIDAeMBygGqAYhhZodH  
Rw  
\nOi8veDEuYy5sZW5jci5vcmcvMCIGA1UdIAQbMBkwCAYGZ4EMAQIBMA0GCysGAQ  
QB  
\ngt8TAQEBMA0GCSqGS1b3DQEBcWUAA4ICAQCfYk5HPqP3hUSFvNVneLKYy611T  
R6W\n\nPTNlclQtgaDqw+34IL9fzLdwALduO/ZelN7klJ+m74uyA+eitRY8kc607TkC53wl  
\nikfmZW4/RvTZ8M6UK+5UzhK8jCdLuMGYL6KvzXGRSgi3yLgjewQtCPkIVz6D2QQz  
\nCkcheAmCJ8MqyJu5zlzyZMjAvnnAT45tRAXekrsu94sQ4egdRCnbWSDtY7kh+Blm  
\nJNXoB1IBMEKlq4QDUOXoRgffuDghje1WrG9ML+Hbisq/  
yFOGwXD9RiX8F6sw6W4\n\nnavAuvDszue5L3sz85K+EC4Y/wFVDNvZo4TYXao6Z0f  
+lQKc0t8DQYzk1OXVu8rp2\n\nnyJMC6alLbBfODALZvYH7n7do1AZIs4l9d1P4jnkDrQoxB3  
UqQ9hVI3LEKQ73xF1O\n\nnyK5GhDDX8oVfGKF5u  
+declsH4YaTw7mP3GFxJSqv3+0lUFJoi5Lc5da149p90lds  
\nhCExroL1+7mrylkXPeFM5TgO9r0rvZaBFovV2z0gp35Z0+L4WPIbuEjN/lxPFin+  
\nHIUjr8gRsl3qfJOQFy/9rKIJR0Y/8OmwT/8oTWgy1mdeHm mj7j1nYsvC9JSQ6Zv  
\nMIdITTKB3zhThV1+XWYp6rjd5JW1zbVWEkLNxE7GJThEUG3szgBVGP7pSWTUTsq  
X\n\nnLRbwHOoq7hHwg==\n\n-----END CERTIFICATE-----\n\n-----BEGIN CERTIFICATE-----  
\nMIIFYDCCBEigAwIBAgIQQAF3ITfU6UK47naqPGQKtZANBgkqhkiG9w0BAQsFADA/  
\nMSQwlgYDVQQKEExtEaWdpdGFsIFNpZ25hdHVyZSBUCnVzdCBDby4xZzAVBgNVBA  
MT  
\nDkRtVCBSb290IENBIFgzMB4XDTIxMDEyMDE5MTQwM1oXDTI0MDkzMDE4MTQw  
M1ow  
\nTzELMAKGA1UEBhMCVVMxKTAnBgNVBAoTIEludGVybmV0IFNIY3VyaXR5IFJlc2Vh  
\ncmNoIEdyb3VwMRUwEwYDVQQDEwxJUlJHIFJvb3QgWDEwggliMA0GCSqGS1b3D  
QEB\n\nAQUAA4ICDwAwggIKAoICAQCt6CRz9BQ385ueK1coHle  
+3LffOJCMbjzmV6B493XC\n\nnov71am72AE8o295ohmxEk7axY/0UEmu/  
H9LqMZshftEzPLpI9d1537O4/xLxIZpL\n\nnwYqGcWIKZmZsj348cL  
+tKSIG8+TA5oCu4kuPt5l+IAOf00eXfJlIl1PoOK5PCm+D  
\nLtFJV4yAdLbaL9A4jXsDcCEbdfIwPPqPrt3aY6vrFk/CjhFLfs8L6P+1dy70sntK  
\n4EwSJQxwjQMpoOFTJOwT2e4ZvxCzSow/  
iaNhUd6shweU9GNx7C7ib1uYgeGJXDR5\n\nnbHbvO5BieebbpJovJsXQEOEO3tkQjhb7t/  
eo98flAgeYjzYllefI5YNNnWe+w5y\n\nnsR2bvAP5SQXYgd0FtCrWQemsAXaVCg/  
Y39W9Eh81LygXbNKYwagJZHduRze6zqxZ\n\nXmidf3LWicUGQSk  
+WT7dJvUkyRGnWqNMQB9GoZm1pzpRboY7nn1ypxIFeFntPIF4\n\nnFQsDj43QLwWyPn  
tKHETzBRL8xurgUBN8Q5N0s8p0544fAQjQMNRbcTa0B7rBMDbC  
\nSLeCO5imfWCKoqMpgsy6vYMEG6KDA0Gh1gXxG8K28Kh8hjtGqEgqiNx2mna/H2ql  
\nPRmP6zjzZN7IKw0KKP/32+IVQtQi0Cdd4Xn+GOdwiK1O5tmLOsbdJ1Fu/7xk9TND  
\nTwIDAQABo4IBRjCCAUIwDwYDVR0TAAQH/BAUwAwEB/

zAOBgNVHQ8BAf8EBAMCAQYw  
\\nSwYIKwYBBQUHAQEEPPzA9MDsGCCsGAQUFBzACHi9odHRwOi8vYXBwcy5pZGVu  
dHJ1\\nc3QuY29tL3Jvb3RzL2RzdHJvb3RjYXgzLnA3YzAfBgNVHSMEGDAWgBTEp7G  
keyxx\\n+tvhS5B1/8QVYIWJEDBUBgNVHSAETTLMAgGBmeBDAECATA/  
BgsrBgEEAYLfEwEB  
\\nATAwMC4GCCsGAQUFBwIBFiJodHRwOi8vY3BzLnJvb3QteDEubGV0c2VuY3J5cHQ  
u  
\\nb3JnMDwGA1UdHwQ1MDMwMaAvoC2GK2h0dHA6Ly9jcmwuaWRlbnRydXN0LmNv  
bS9E  
\\nU1RST09UQ0FYM0NSTC5jcmwwHQYDVR0OBBYEFHm0WeZ7tuXkAXOACljlGlj26  
Ztu\\nMA0GCSqGSIb3DQEBCwUAA4IBAQAkCwBslm7/DILQrt2M51oGrS+o44+/  
yQoDFVDC  
\\n5WxCu2+b9LRPwkSICHXM6webFGJueN7sJ7o5XPWioW5WIHAQU7G75K/  
QosMrAdSW\\n9MUgNTP52GE24HGNtLi1qoJfIcDyqSMo59ahy2cl2qBDLKobkx/  
J3vWraV0T9VuG  
\\nWCLKTVXkcGdtwlfFRjlBz4pYg1htmf5X6DYO8A4jqv2II9DjXA6USbW1FzXSLr9O  
\\nhe8Y4IWS6wY7bCkjCWDcRQJMEhg76fsO3txE+FiYruq9RUWhiF1myv4Q6W  
+CyBFC\\nDfvp7OOGAN6dEOM4+qR9sdjoSYKEBpsr6GtPAQw4dy753ec5\\n-----END  
CERTIFICATE-----\\n-----BEGIN CERTIFICATE-----\\nMIIDSjCCAjKgAwIBAgIQRK  
+wgNajJ7qJMDmGLvhAazANBqkqhkiG9w0BAQUFADA/  
\\nMSQwlgYDVQQKEExtEaWdpdGFsIFNpZ25hdHVyZSBUCnVzdCBDby4xFzAVBgNVBA  
MT  
\\nDkRTVCBSb290IENBIFgzMB4XDTAwMDkzMDIxMTIxOVVoXDTIxMDkzMDE0MDExN  
Vow  
\\nPzEkMCIGA1UEChMbRGlnaXRhbCBTaWduYXR1cmUgVHJ1c3QgQ28uMRcwFQYD  
VQQD  
\\nEw5EU1QgUm9vdCBDQSBYmzCCASlwDQYJKoZIhvcNAQEBBQADggEPADCCAQ  
oCggEB\\nAN+v6ZdQCINXtMxiZfaQguzH0yxrMMpb7NnDfcdAwRgUi+DoM3ZJKuM/  
lUmTrE4O\\nrz5ly2Xu/  
NMhD2XSKtkyj4zl93ewEnu1lcCJo6m67XMuegwGMOifooUMM0RoOEQ  
\\nOLI5CjH9UL2AZd+3UWODyOKIYepLYYHsUmu5ouJLGiifSKOeDNoJjj4XLh7dIN9b  
\\nxiqKqy69cK3FCxolkHRyxXtqqzTWMI\\n/5WgTe1QLyNau7Fqckh49ZLOMxt+/yUFw  
\\n7BZy1SbsOFU5Q9D8/RhcQPGX69Wam40dutolucbY38EVAjqr2m7xPi71XAicPNaD  
\\naeQQmxkqtilX4+U9m5/wAl0CAwEAANCMEEAwDwYDVR0TAQH/BAUwAwEB/  
zAOBgNV\\nHQ8BAf8EBAMCAQYwHQYDVR0OBBYEFMSnsaR7LHH62+FLkHX/  
xBVghYkQMA0GCSqG\\nSIb3DQEBCwUAA4IBAQCjGiybFwBcqR7uKGY3Or  
+Dxz9LwwmgISBd49IZRNI+DT69\\nikugdB/  
OEIKcdBodfpga3csTS7MgROSR6cz8faXbauX+5v3gTt23ADq1cEmv8uXr  
\\nAvHRAosZy5Q6XkjEGB5YGV8eAlrwDPGxrancWYaLbumR9YbK+rlmM6pZW87ipxZz  
\\nR8srzJmwN0jP41ZL9c8PDHlyh8bwRLtTcm1D9SZlmlJnt1ir/  
md2cXjbDaJWFBM5\\nJDGFoqgCWjBH4d1QB7wCCZAA62RjYJsWvljJEubSfZGL  
+T0yjWW06YyxV3bqxbYo\\nOb8VZRzI9neWagqNdwvYkQsEjgfbKbYK7p2CNTUQ\\n-----  
END CERTIFICATE-----\\n","primaryButtonText":"Reload","primaryParagraph":"Attackers  
might be trying to steal your information from \\u003Cstrong>accounts.shopify.com  
\\u003C/strong> (for example, passwords, messages, or credit cards). \\u003Ca href=\\\"#  
\\\" id=\\\"learn-more-link\\\">Learn more\\u003C/a>","recurrentErrorParagraph":"Warnings



Af8EAjAA\nMB0GA1UdDgQWBBr/TRz5xF8Ma/  
wcElldjg8fyl3L3zAfBgNVHSMEGDAWgBQUlRmX  
\nt1hWy65QCUDmH6+dixTCxjBVBggrBgEFBQcBAQRJMEcwIQYIKwYBBQUHMAGGF  
Wh0\ndHA6Ly9yMy5vLmxlbmNyLm9yZzAiBggrBgEFBQcwAoYWaHR0cDovL3lzLmkub  
GVu\nY3lub3JnLzCBYQYDVR0RBIHBMIG  
+gh5hY2NvdW50cy1kZW1vLnNob3BpZnlibG91\nZC5jb22CFGFjY291bnRzLnNob3BpZ  
nkuY29tgh9jdXN0b21lcnMtZGVtby5zaG9w  
\naWZ5Y2xvdWQuY29tgiZpZGVudGI0eS1tYWluLWNsdXN0ZXluc2hvcGlmeWNsb3Vk  
\nLmNvbYliaWRlbnRpdHktbWVyY2hhbnQuc2hvcGlmeWNsb3VkLmNvbYIZaWRlbnRp  
\ndHkuc2hvcGlmeWNsb3VkLmNvbTBMBgNVHSAERTBDMAgGBmeBDAECATA3BgSR  
BgEE  
\nAYLfEwEBATAoMCYGCCsGAQUFBwIBFhpodHRwOi8vY3BzLmxldHNlbnNyeXB0L  
m9y\nZzCCAQMGCisGAQQB1nkCBAIEgfQEgEA7wB1AESUZS6w7s6vxEAH2Kj  
+KMDa5oK+\n2MsxtT/TM5a1toGoAAABe0rRIFkAAAQDAEYwRAIgcU9G4Kf  
+YE3BwY4IHwrPfCyx\nHBfmiWZVYY3syP6PMtQCICbrsrVPy0eYoa  
+way7bJOXuRAbwT4Sfd4Nd14ITkrE\nAHYAft7y+l//  
iFVoJMLAyp5SiXkrxQ54CX8uapdomX4i8NcAAAF7StEgegAABAMA\nRzBFaiBQFP/  
Jq9s0RX2lddzPRWorKmK2vHb0gBjdeR4DafmJfglhAOGXW8pNxx+q  
\nAgkEhlLco7khwJUOAcDqcKCI9HS9FOJ8MA0GCSqGSIb3DQEBCwUAA4IBAQBcKa  
nD\nDwicj0QRpqqbMyHi1uwU0vJoOuCXAB5TMevD1OGhl2HGfV8blgxWlfOULxGoEd/  
k\nSA0t9X94AFhu+FhBQnrssUH/D+xQFKSbaZcQIB5j1nEFNdtpokxW5JyhjS5MXiYO  
\ndWfgQcEhx3HlnYVT+Z1CpNslgCSOaoOawx+b1y1/Kj5ZIEld2rZBIOxh+/ifPP9h\nZ8Y/  
kug+O2Vu7xfPWn4RtVeG39FpK3FcbOxkTA09K1CHtPQf+KZ/  
Ou5BcLn8Bji5\nMVXdjQTeKuzilyV7QM+dAxHn3/  
reMXmlGengm7nj9K2P3I7Z70dX1sEwUXQR8Wr\n+/lfvfwGqqjAuskK\n-----END  
CERTIFICATE-----\n-----BEGIN CERTIFICATE-----  
\nMIIFFjCCAv6gAwIBAgIRAJErCErPDBinU/bWLiWnX1owDQYJKoZIhvcNAQELBQAw  
\nTzELMAkGA1UEBhMCVVMxKTAnBgNVBAoTIEludGVybWV0IFNIY3VyaXR5IFJlc2Vh  
\ncmNoIEdyb3VwMRUwEwYDVQQDEwxFb3VwMRUwEwYDVQQDEwxFb3VwMRUwEwYDVQ  
wMDAw  
\nWhcNMjUwOTE1MTYwMDAwWjA5MQswCQYDVQQGEwJVUzEWMBQGA1UEChM  
NTGV0J3Mg  
\nRW5jcnlwdDELMAkGA1UEAxMCUjMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAw  
ggEK\nAoIBAQC7AhUozPaglNMPEuyNVZLD+ILxmaZ6QoinXSaqtSu5xUyxr45r  
+XXlo9cP  
\nR5QUVTvXjJ6oojkZ9Yl8QqlObvU7wy7bjcCwXPNZOOftz2nwWgsbvsCUJCWH+jdx  
\nsxPnHKzhm+/b5DtFukWWqcFTzjTIUu61ru2P3mBw4qVUq7ZtDpelQDRrK9O8Zutm  
\nNHZ6a4uPVymZ+DAXXbpyb/uBxa3Shlg9F8fnCbvxK/eG3MHacV3URuPMrSXBilXg  
\nZ3Vms/EY96Jc5IP/Ooi2R6X/ExjqmAl3P51T+c8B5fWmcBcUr2Ok/5mzk53cU6cGn/  
kiFHaFpriV1uxPMUgP17VGhi9sVAgMBAAGjggEIMIIBBDAOBgNVHQ8BAf8EBAMC  
\nAYYwHQYDVR0IBBYwFAYIKwYBBQUHAWIGCCsGAQUFBwMBMBIGA1UdEwEB/  
wQIMAYB  
\nAf8CAQAwHQYDVR0OBBYEFBQusxe3WFbLrIAJQOYfr52LFMLGMB8GA1UdIwQY  
MBAA  
\nFHm0WeZ7tuXkAXOAClJlGij26ZtuMDIGCCsGAQUFBwEBBCYwJDAiBggrBgEFBQc  
w



\nAoYWaHR0cDovL3gxLmkubGVuY3lub3JnLzAnBgNVHR8EIDAeMBygGqAYhhZodH  
Rw  
\nOi8veDEuYy5sZW5jci5vcmcvMCIGA1UdIAQbMBkwCAYGZ4EMAQIBMA0GCysGAQ  
QB  
\ngt8TAQEBMA0GCSqGSib3DQEBCwUAA4ICAQCFyk5HPqP3hUSFvNVneLKYY611T  
R6W\nPTNlclQtgaDqw+34IL9fzLdwALduO/ZelN7klJ+m74uyA+eitRY8kc607TkC53wl  
\nikfmZW4/RvTZ8M6UK+5UzhK8jCdLuMGYL6KvzXGRSgi3yLgjewQtCPkIVz6D2QQz  
\nCkcheAmCJ8MqyJu5zlzyZMjAvnnAT45tRAXekrsu94sQ4egdRCnbWSDtY7kh+Blm  
\nIJNXoB1IBMEKlq4QDUOXoRgffuDghje1WrG9ML+Hbisq/  
yFOGwXD9RiX8F6sw6W4\nnavAuvDszue5L3sz85K+EC4Y/wFVDNvZo4TYXao6Z0f  
+IQKc0t8DQYzk1OXVu8rp2\nnyJMC6alLbBfODALZvYH7n7do1AZls4l9d1P4jnkDrQoxB3  
UqQ9hVI3LEKQ73xF1O\nnyK5GhDDX8oVfGKF5u  
+declsH4YaTw7mP3GFxJSqv3+0IUFJoi5Lc5da149p90lds  
\nhCExroL1+7mrylkXPeFM5TgO9r0rvZaBFOvV2z0gp35Z0+L4WPIbuEjN/lxPFin+  
\nHIUjr8gRsI3qfJOQFy/9rKIJR0Y/8OmwT/8oTWgy1mdeHm mj7j1nYsvC9JSQ6Zv  
\nMldITTKB3zhThV1+XWYp6rjd5JW1zbVWEkLNxE7GJThEUG3szgBVGP7pSWTUTsq  
X\nnLRbwHOoq7hHwg==\n-----END CERTIFICATE-----\n-----BEGIN CERTIFICATE-----  
\nMIIFYDCCBEigAwIBAgIQQAF3ITfU6UK47naqPGQKtZANBgkqhkiG9w0BAQsFADA/  
\nMSQwlgYDVQQKExtEaWdpdGFsIFNpZ25hdHVyZSBUCnVzdCBDby4xFzAVBgNVBA  
MT  
\nDkRtVCBSb290IENBIFgzMB4XDTIxMDEyMDE5MTQwM1oXDTI0MDkzMDE4MTQw  
M1ow  
\nTzELMAkGA1UEBhMCVVMxKTAnBgNVBAoTIEludGVybmV0IFNIY3VyaXR5IFJlc2Vh  
\ncmNoIEdyb3VwMRUwEwYDVQQDEwXJU1JHIFJvb3QgWDEwgglMA0GCSqGSib3D  
QEB\nAQUAA4ICDwAwggIKAoICAQCt6CRz9BQ385ueK1coHle  
+3LffOJCMbjzmV6B493XC\nnov71am72AE8o295ohmxEk7axY/0UEmu/  
H9LqMZshftEzPLPl9d1537O4/xLxIZpL\nnwYqGcWIKZmZsj348cL  
+tKSIG8+TA5oCu4kuPt5l+IAOf00eXfJlII1PoOK5PCm+D  
\nLtFJV4yAdLbaL9A4jXsDcCEbdfwPPqPrt3aY6vrFk/CjhFLfs8L6P+1dy70sntK  
\n4EwSJQxwjQMpoOFTJOwT2e4ZvxCzSow/  
iaNhUd6shweU9GNx7C7ib1uYgeGJXDR5\nnbHbvO5BieebbpJovJsXQEOEO3tkQjhb7t/  
eo98flAgeYjzYllefIn5YNNnWe+w5y\nnsR2bvAP5SQXYgd0FtCrWQemsAXaVCg/  
Y39W9Eh81LygXbNKYwagJZHduRze6zqxZ\nnXmidf3LWicUGQSk  
+WT7dJvUkyRGnWqNMQB9GoZm1pzpRboY7nn1ypxIFeFntPIF4\nnFQsDj43QLwWyPn  
tKHEtzBRL8xurgUBN8Q5N0s8p0544fAQjQMNRbcTa0B7rBMDbC  
\nSLeCO5imfWCKoqMpgsy6vYMEG6KDA0Gh1gXxG8K28Kh8hjtGqEgqiNx2mna/H2ql  
\nPRmP6zjzZN7IKw0KKP/32+IVQtQi0Cdd4Xn+GOdwiK1O5tmLOsbdJ1Fu/7xk9TND  
\nTwIDAQABo4IBRjCCAUIlwDwYDVR0TAQH/BAUwAwEB/  
zAOBgNVHQ8BAf8EBAMCAQYw  
\nSwYIKwYBBQUHAQEEPzA9MDsGCCsGAQUFBzAChi9odHRwOi8vYXBwcy5pZGVu  
dHJ1\nc3QuY29tL3Jvb3RzL2RzdHJvb3RjYXgzLnA3YzAfBgNVHSMEGDAWgBTEp7G  
keyxx\n+tvhS5B1/8QVYIWJEDBUBgNVHSAETTBLMAgGBmeBDAECATA/  
BgSrBgEEAYLfEwEB  
\nATAwMC4GCCsGAQUFBwIBFiJodHRwOi8vY3BzLnJvb3QteDEubGV0c2VuY3J5cHQ  
u  
\nb3JnMDwGA1UdHwQ1MDMwMaAvoC2GK2h0dHA6Ly9jcmwuaWRlbnRydXN0LmNv

bS9E  
\\nU1RST09UQ0FYM0NSTC5jcmwwHQYDVR0OBBYEFHm0WeZ7tuXkAXOAClJlGj26  
Ztu\\nMA0GCSqGSIb3DQEBCwUAA4IBAQAkCwBslm7/DILQrt2M51oGrS+o44+/  
yQoDFVDC  
\\n5WxCu2+b9LRPwkSICHXM6webFGJueN7sJ7o5XPWioW5WIHAQU7G75K/  
QosMrAdSW\\n9MUgNTP52GE24HGNTLi1qoJFicDyqSMo59ahy2cI2qBDLKobkx/  
J3vWraV0T9VuG  
\\nWCLKTVXkcGdtwlfFRjlBz4pYg1html5X6DYO8A4jqv2II9DjXA6USbW1FzXSLr9O  
\\nhe8Y4IWS6wY7bCkjCWDcRQJMEhg76fsO3txE+FiYruq9RUWhiF1myv4Q6W  
+CyBFC\\nDfvp7OOGAN6dEOM4+qR9sdjoSYKEBpsr6GtPAQw4dy753ec5\\n-----END  
CERTIFICATE-----\\n-----BEGIN CERTIFICATE-----\\nMIIDSjCCAjKgAwIBAgIQRK  
+wgNajJ7qJMDmGLvhAazANBgkqhkiG9w0BAQUFADA/  
\\nMSQwlgYDVQQKExtEaWdpdGFsIFNpZ25hdHVyZSBUCnVzdCBDby4xFzAVBgNVBA  
MT  
\\nDkRtVCBSb290IENBIFgzMB4XDTAwMDkzMDIxMTIxOVVoXDTIxMDkzMDE0MDExN  
Vow  
\\nPzEkMCIGA1UEChMbRGlnaXRhbCBTaWduYXR1cmUgVHJ1c3QgQ28uMRcwFQYD  
VQQD  
\\nEw5EU1QgUm9vdCBDQSBYMzCCASlwDQYJKoZIhvcNAQEBBQADggEPADCCAQ  
oCggEB\\nAN+v6ZdQCINXtMxiZfaQguzH0yxrMMpb7NnDfcdAwRgUi+DoM3ZJKuM/  
IUmtRE4O\\nrz5ly2Xu/  
NMhD2XSKtkyj4zl93ewEnu1lcCJo6m67XMuegwGMOifooUMM0RoOEq  
\\nOLI5CjH9UL2AZd+3UWODyOKIYepLYYHsUmu5ouJLGiifSKOeDNoJjj4XLh7dIN9b  
\\nxiqKqy69cK3FCxolkHRyxXtqqzTWMIn/5WgTe1QLyNau7Fqckh49ZLOMxt+/yUFw  
\\n7BZy1SbsOFU5Q9D8/RhcQPGX69Wam40dutolucbY38EVAjqr2m7xPi71XAicPNaD  
\\naeQQmxkqtilX4+U9m5/wAl0CAwEAAANCMEEAwDwYDVR0TAQH/BAUwAwEB/  
zAOBgNV\\nHQ8BAf8EBAMCAQYwHQYDVR0OBBYEFMSnsaR7LHH62+FLkHX/  
xBVghYkQMA0GCSqG\\nSlb3DQEBCwUAA4IBAQCjGiybFwBcqR7uKGY3Or  
+Dxz9LwwmgISBd49IZRNI+DT69\\nikugdB/  
OEIKcdBodfpga3csTS7MgROSR6cz8faXbauX+5v3gTt23ADq1cEmv8uXr  
\\nAvHRAosZy5Q6XkjEGB5YGV8eAlrwDPGxrancWYaLbumR9YbK+rlmM6pZW87ipxZz  
\\nR8srzJmwN0jP41ZL9c8PDHlyh8bwRLtTcm1D9SZlmlJnt1ir/  
md2cXjbDaJWFBM5\\nJDGFoqgCWjBH4d1QB7wCCZAA62RjYJsWvljJEubSfZGL  
+T0yjWW06XyxV3bqxbYo\\nOb8VZRzI9neWagqNdwwYkQsEjgfbKbYK7p2CNTUQ\\n-----  
END CERTIFICATE-----\\n","primaryButtonText":"Reload","primaryParagraph":"Attackers  
might be trying to steal your information from \\u003Cstrong>accounts.shopify.com  
\\u003C/strong> (for example, passwords, messages, or credit cards). \\u003Ca href=\\\"#  
\\\" id=\\\"learn-more-link\\\">Learn more\\u003C/a>","recurrentErrorParagraph":"Warnings  
may be common while websites update their security. This should improve  
soon.","show\_recurrent\_error\_paragraph":false,"subject":"accounts.shopify.com","tabTitl  
e":"Privacy error","textdirection":"ltr","type":"SSL"};</script>

-----

```
<html dir="ltr" lang="en"><head>
  <meta charset="utf-8">
  <meta name="color-scheme" content="light dark">
  <meta name="theme-color" content="#fff">
  <meta name="viewport" content="initial-scale=1, minimum-scale=1, width=device-
width">
  <meta http-equiv="Content-Security-Policy" content="require-trusted-types-for 'script';
trusted-types;">
  <title>Privacy error</title>
  <style>/* Copyright 2017 The Chromium Authors. All rights reserved.
* Use of this source code is governed by a BSD-style license that can be
* found in the LICENSE file. */
```

```
a {
  color: var(--link-color);
}
```

```
body {
  --background-color: #fff;
  --error-code-color: var(--google-gray-700);
  --google-blue-100: rgb(210, 227, 252);
  --google-blue-300: rgb(138, 180, 248);
  --google-blue-600: rgb(26, 115, 232);
  --google-blue-700: rgb(25, 103, 210);
  --google-gray-100: rgb(241, 243, 244);
  --google-gray-300: rgb(218, 220, 224);
  --google-gray-500: rgb(154, 160, 166);
  --google-gray-50: rgb(248, 249, 250);
  --google-gray-600: rgb(128, 134, 139);
  --google-gray-700: rgb(95, 99, 104);
  --google-gray-800: rgb(60, 64, 67);
  --google-gray-900: rgb(32, 33, 36);
  --heading-color: var(--google-gray-900);
  --link-color: rgb(88, 88, 88);
  --popup-container-background-color: rgba(0,0,0,.65);
  --primary-button-fill-color-active: var(--google-blue-700);
  --primary-button-fill-color: var(--google-blue-600);
  --primary-button-text-color: #fff;
  --quiet-background-color: rgb(247, 247, 247);
  --secondary-button-border-color: var(--google-gray-500);
  --secondary-button-fill-color: #fff;
  --secondary-button-hover-border-color: var(--google-gray-600);
  --secondary-button-hover-fill-color: var(--google-gray-50);
  --secondary-button-text-color: var(--google-gray-700);
  --small-link-color: var(--google-gray-700);
  --text-color: var(--google-gray-700);
```

```
background: var(--background-color);
color: var(--text-color);
word-wrap: break-word;
}
```

```
.nav-wrapper .secondary-button {
background: var(--secondary-button-fill-color);
border: 1px solid var(--secondary-button-border-color);
color: var(--secondary-button-text-color);
float: none;
margin: 0;
padding: 8px 16px;
}
```

```
.hidden {
display: none;
}
```

```
html {
-webkit-text-size-adjust: 100%;
font-size: 125%;
}
```

```
.icon {
background-repeat: no-repeat;
background-size: 100%;
}
```

```
@media (prefers-color-scheme: dark) {
body {
--background-color: var(--google-gray-900);
--error-code-color: var(--google-gray-500);
--heading-color: var(--google-gray-500);
--link-color: var(--google-blue-300);
--primary-button-fill-color-active: rgb(129, 162, 208);
--primary-button-fill-color: var(--google-blue-300);
--primary-button-text-color: var(--google-gray-900);
--quiet-background-color: var(--background-color);
--secondary-button-border-color: var(--google-gray-700);
--secondary-button-fill-color: var(--google-gray-900);
--secondary-button-hover-fill-color: rgb(48, 51, 57);
--secondary-button-text-color: var(--google-blue-300);
--small-link-color: var(--google-blue-300);
--text-color: var(--google-gray-500);
}
}
```

```
</style>
```

```
<style>/* Copyright 2014 The Chromium Authors. All rights reserved.
```

```
Use of this source code is governed by a BSD-style license that can be  
found in the LICENSE file. */
```

```
button {  
  border: 0;  
  border-radius: 4px;  
  box-sizing: border-box;  
  color: var(--primary-button-text-color);  
  cursor: pointer;  
  float: right;  
  font-size: .875em;  
  margin: 0;  
  padding: 8px 16px;  
  transition: box-shadow 150ms cubic-bezier(0.4, 0, 0.2, 1);  
  user-select: none;  
}
```

```
[dir='rtl'] button {  
  float: left;  
}
```

```
.bad-clock button,  
.captive-portal button,  
.https-only button,  
.insecure-form button,  
.lookalike-url button,  
.main-frame-blocked button,  
.neterror button,  
.pdf button,  
.ssl button,  
.safe-browsing-billing button {  
  background: var(--primary-button-fill-color);  
}
```

```
button:active {  
  background: var(--primary-button-fill-color-active);  
  outline: 0;  
}
```

```
#debugging {  
  display: inline;  
  overflow: auto;  
}
```

```
.debugging-content {  
  line-height: 1em;  
  margin-bottom: 0;  
  margin-top: 1em;  
}
```

```
.debugging-content-fixed-width {  
  display: block;  
  font-family: monospace;  
  font-size: 1.2em;  
  margin-top: 0.5em;  
}
```

```
.debugging-title {  
  font-weight: bold;  
}
```

```
#details {  
  margin: 0 0 50px;  
}
```

```
#details p:not(:first-of-type) {  
  margin-top: 20px;  
}
```

```
.secondary-button:active {  
  border-color: white;  
  box-shadow: 0 1px 2px 0 rgba(60, 64, 67, .3),  
    0 2px 6px 2px rgba(60, 64, 67, .15);  
}
```

```
.secondary-button:hover {  
  background: var(--secondary-button-hover-fill-color);  
  border-color: var(--secondary-button-hover-border-color);  
  text-decoration: none;  
}
```

```
.error-code {  
  color: var(--error-code-color);  
  font-size: .8em;  
  margin-top: 12px;  
  text-transform: uppercase;  
}
```

```
#error-debugging-info {  
  font-size: 0.8em;
```

```
}
```

```
h1 {  
  color: var(--heading-color);  
  font-size: 1.6em;  
  font-weight: normal;  
  line-height: 1.25em;  
  margin-bottom: 16px;  
}
```

```
h2 {  
  font-size: 1.2em;  
  font-weight: normal;  
}
```

```
.icon {  
  height: 72px;  
  margin: 0 0 40px;  
  width: 72px;  
}
```

```
input[type=checkbox] {  
  opacity: 0;  
}
```

```
input[type=checkbox]:focus ~ .checkbox::after {  
  outline: -webkit-focus-ring-color auto 5px;  
}
```

```
.interstitial-wrapper {  
  box-sizing: border-box;  
  font-size: 1em;  
  line-height: 1.6em;  
  margin: 14vh auto 0;  
  max-width: 600px;  
  width: 100%;  
}
```

```
#main-message > p {  
  display: inline;  
}
```

```
#extended-reporting-opt-in {  
  font-size: .875em;  
  margin-top: 32px;  
}
```

```
#extended-reporting-opt-in label {  
  display: grid;  
  grid-template-columns: 1.8em 1fr;  
  position: relative;  
}
```

```
#enhanced-protection-message {  
  border-radius: 4px;  
  font-size: 1em;  
  margin-top: 32px;  
  padding: 10px 5px;  
}
```

```
#enhanced-protection-message label {  
  display: grid;  
  grid-template-columns: 2.5em 1fr;  
  position: relative;  
}
```

```
#enhanced-protection-message div {  
  margin: 0.5em;  
}
```

```
#enhanced-protection-message .icon {  
  height: 1.5em;  
  vertical-align: middle;  
  width: 1.5em;  
}
```

```
.nav-wrapper {  
  margin-top: 51px;  
}
```

```
.nav-wrapper::after {  
  clear: both;  
  content: " ";  
  display: table;  
  width: 100%;  
}
```

```
.small-link {  
  color: var(--small-link-color);  
  font-size: .875em;  
}
```



```
.checkboxes {  
  flex: 0 0 24px;  
}
```

```
.checkbox {  
  --padding: .9em;  
  background: transparent;  
  display: block;  
  height: 1em;  
  left: -1em;  
  padding-inline-start: var(--padding);  
  position: absolute;  
  right: 0;  
  top: -.5em;  
  width: 1em;  
}
```

```
.checkbox::after {  
  border: 1px solid white;  
  border-radius: 2px;  
  content: "";  
  height: 1em;  
  left: var(--padding);  
  position: absolute;  
  top: var(--padding);  
  width: 1em;  
}
```

```
.checkbox::before {  
  background: transparent;  
  border: 2px solid white;  
  border-inline-end-width: 0;  
  border-top-width: 0;  
  content: "";  
  height: .2em;  
  left: calc(.3em + var(--padding));  
  opacity: 0;  
  position: absolute;  
  top: calc(.3em + var(--padding));  
  transform: rotate(-45deg);  
  width: .5em;  
}
```

```
input[type=checkbox]:checked ~ .checkbox::before {  
  opacity: 1;  
}
```

```
#recurrent-error-message {  
  background: #ededed;  
  border-radius: 4px;  
  margin-bottom: 16px;  
  margin-top: 12px;  
  padding: 12px 16px;  
}
```

```
.showing-recurrent-error-message #extended-reporting-opt-in {  
  margin-top: 16px;  
}
```

```
.showing-recurrent-error-message #enhanced-protection-message {  
  margin-top: 16px;  
}
```

```
@media (max-width: 700px) {  
  .interstitial-wrapper {  
    padding: 0 10%;  
  }  
}
```

```
#error-debugging-info {  
  overflow: auto;  
}  
}
```

```
@media (max-width: 420px) {  
  button,  
  [dir='rtl'] button,  
  .small-link {  
    float: none;  
    font-size: .825em;  
    font-weight: 500;  
    margin: 0;  
    width: 100%;  
  }  
}
```

```
button {  
  padding: 16px 24px;  
}
```

```
#details {  
  margin: 20px 0 20px 0;  
}
```

```

#details p:not(:first-of-type) {
  margin-top: 10px;
}

.secondary-button:not(.hidden) {
  display: block;
  margin-top: 20px;
  text-align: center;
  width: 100%;
}

.interstitial-wrapper {
  padding: 0 5%;
}

#extended-reporting-opt-in {
  margin-top: 24px;
}

#enhanced-protection-message {
  margin-top: 24px;
}

.nav-wrapper {
  margin-top: 30px;
}
}

/**
 * Mobile specific styling.
 * Navigation buttons are anchored to the bottom of the screen.
 * Details message replaces the top content in its own scrollable area.
 */

@media (max-width: 420px) {
  .nav-wrapper .secondary-button {
    border: 0;
    margin: 16px 0 0;
    margin-inline-end: 0;
    padding-bottom: 16px;
    padding-top: 16px;
  }
}

/* Fixed nav. */
@media (min-width: 240px) and (max-width: 420px) and

```

```

        (min-height: 401px),
        (min-width: 421px) and (min-height: 240px) and
        (max-height: 560px) {
body .nav-wrapper {
    background: var(--background-color);
    bottom: 0;
    box-shadow: 0 -12px 24px var(--background-color);
    left: 0;
    margin: 0 auto;
    max-width: 736px;
    padding-inline-end: 24px;
    padding-inline-start: 24px;
    position: fixed;
    right: 0;
    width: 100%;
    z-index: 2;
}

.interstitial-wrapper {
    max-width: 736px;
}

#details,
#main-content {
    padding-bottom: 40px;
}

#details {
    padding-top: 5.5vh;
}

button.small-link {
    color: var(--google-blue-600);
}
}

@media (max-width: 420px) and (orientation: portrait),
        (max-height: 560px) {
body {
    margin: 0 auto;
}

button,
[dir='rtl'] button,
button.small-link,
.nav-wrapper .secondary-button {

```

```
font-family: Roboto-Regular,Helvetica;  
font-size: .933em;  
margin: 6px 0;  
transform: translatez(0);  
}
```

```
.nav-wrapper {  
  box-sizing: border-box;  
  padding-bottom: 8px;  
  width: 100%;  
}
```

```
#details {  
  box-sizing: border-box;  
  height: auto;  
  margin: 0;  
  opacity: 1;  
  transition: opacity 250ms cubic-bezier(0.4, 0, 0.2, 1);  
}
```

```
#details.hidden,  
#main-content.hidden {  
  height: 0;  
  opacity: 0;  
  overflow: hidden;  
  padding-bottom: 0;  
  transition: none;  
}
```

```
h1 {  
  font-size: 1.5em;  
  margin-bottom: 8px;  
}
```

```
.icon {  
  margin-bottom: 5.69vh;  
}
```

```
.interstitial-wrapper {  
  box-sizing: border-box;  
  margin: 7vh auto 12px;  
  padding: 0 24px;  
  position: relative;  
}
```

```
.interstitial-wrapper p {
```

```
font-size: .95em;
line-height: 1.61em;
margin-top: 8px;
}
```

```
#main-content {
  margin: 0;
  transition: opacity 100ms cubic-bezier(0.4, 0, 0.2, 1);
}
```

```
.small-link {
  border: 0;
}
```

```
.suggested-left > #control-buttons,
.suggested-right > #control-buttons {
  float: none;
  margin: 0;
}
}
```

```
@media (min-width: 421px) and (min-height: 500px) and (max-height: 560px) {
  .interstitial-wrapper {
    margin-top: 10vh;
  }
}
```

```
@media (min-height: 400px) and (orientation:portrait) {
  .interstitial-wrapper {
    margin-bottom: 145px;
  }
}
```

```
@media (min-height: 299px) {
  .nav-wrapper {
    padding-bottom: 16px;
  }
}
```

```
@media (max-height: 560px) and (min-height: 240px) and (orientation:landscape) {
  .extended-reporting-has-checkbox #details {
    padding-bottom: 80px;
  }
}
```

```
@media (min-height: 500px) and (max-height: 650px) and (max-width: 414px) and
```

```
    (orientation: portrait) {  
    .interstitial-wrapper {  
        margin-top: 7vh;  
    }  
}
```

```
@media (min-height: 650px) and (max-width: 414px) and (orientation: portrait) {  
    .interstitial-wrapper {  
        margin-top: 10vh;  
    }  
}
```

/\* Small mobile screens. No fixed nav. \*/

```
@media (max-height: 400px) and (orientation: portrait),  
    (max-height: 239px) and (orientation: landscape),  
    (max-width: 419px) and (max-height: 399px) {  
    .interstitial-wrapper {  
        display: flex;  
        flex-direction: column;  
        margin-bottom: 0;  
    }  
}
```

```
#details {  
    flex: 1 1 auto;  
    order: 0;  
}
```

```
#main-content {  
    flex: 1 1 auto;  
    order: 0;  
}
```

```
.nav-wrapper {  
    flex: 0 1 auto;  
    margin-top: 8px;  
    order: 1;  
    padding-inline-end: 0;  
    padding-inline-start: 0;  
    position: relative;  
    width: 100%;  
}
```

```
button,  
.nav-wrapper .secondary-button {  
    padding: 16px 24px;  
}
```

```
button.small-link {
  color: var(--google-blue-600);
}
}
```

```
@media (max-width: 239px) and (orientation: portrait) {
  .nav-wrapper {
    padding-inline-end: 0;
    padding-inline-start: 0;
  }
}
```

</style>

<style>/\* Copyright 2017 The Chromium Authors. All rights reserved.

\* Use of this source code is governed by a BSD-style license that can be

\* found in the LICENSE file. \*/

```
.bad-clock .icon {
  background-image: -webkit-image-set(
    url(
QVR4Xu3cS1OTVxwG8Ha6dsZNt/0S7ozX
+wUSGKN7yclvkJ2OiNcdbvwMfABXLS1VvLXFSMWUgFAh1oJICCEGq8UFp8+fPu87J
4S3vrmcvlfOceaZMKOSnN/8z/
+c95Yv3B8XFxcXFxcXFzNRZ89+rZLJJNKLDCAZdfp0Hini5zWJ/lzk5e+QAf7bpPzf/
yvKLkD0I1IArCOqwazL70D65Xdub5RUagcA0kDJCYCJ8Hen5b22D8y5czvVmTN9gCk
RwHzwXvKe8t72wij1JWB6AFMIBDAPVZDPIJ/
Fth7zDWAeESD64LPIZ7ICBwAxZBFRIImURiUWLk0ymuCwrG8NtQyqKKfUVYG4RwP7
gs8pnbhsOAO4gapvITluQzFaO+Uoy3nNMAayurlalcOqUKaSUudUKTa9dQN/
t3m0EiWOLtX6fw6W8nUCDyFJHh5ktAMbUuh0yN4HtBPpWgGlxNYQUTSBhTC3Zccv
W3QNoawUR6Af7p49ZpAwtuYPPHIsFukFEehHAA0jy61GwthkjM1UTx8Blq2gleDcQ
+7v3atKnZ2trqK+xs/n8JRFVBUEJL+CBGgYQD/
t369WWomEMTZ0PgkAaQ8g6ik2RJyH+/
apxD6BSm3tpLS9QPXTGDUFTQkTZrT6wGApIJGkKcHD6p38XjLzkwfW6ZADb0oHv
AEaBHAPpZKohAv7YSCWOup3r6CRBpBX3v9R
+pHuSxAB04oDKI4EjGDh1qDRLGXA9Q1iagYQFigxagJwR6BpwxptIsEsYc
+roVANajBhpkcXag+3qDBs5TRMd5jowfPtws0rqMPUz1JAKbQ8i0F1vBWP/
GQHOqEwvgWGywPmNWW0GCWMPA9RrA9Cg339YPQR6AhwBGtuEM47kjhRE0jD
SBh7GKABC4CqGzSBZHpINCDA6PGBXhw9qt4nEo0ADYQBytgCJLvnB97yLs2ZQM
+Q59WV4+NM8nWqESSMPcwoOm8BEA8v2KC5QcxlgxYgTq0sgSY8HFAP4Ewjvx87
Vi9SPgxQ0QYgHqD6q9cIK2jU7z2sHIY4NUAzyF/hkYphptiaDUBjFy6oYa//
aLtnArHn1FaOjvMSkdfZsEgY+7YBqpTLKnvxonrljPiTS/
ijBPHAxKYFxrONDLD5l8f33j9AKQQQHZOsaVr1wSIBmkcSN7hxagGIPtM5Xg4swR6xX
wGqWhnk2aWr1/fEmni0iUfiNPLX84nQ+L8wbw+ceK/
kPKGIInnzSFNA8nbPOQBNalOKOEw1DiMwfuaQj11dDS7z3ChGmVIA0nRvr8oCaKPvb
MJ5ycxqOLObKudP4syfPCmpRcLYDRxqmMIKANLM5cvSf/
```



zqmUI4rQik9ZwAnDnkDbKA/  
K0jYeyhD1ZtRnoFJKxSWuVwWjHEERjijyCsrR8N5iyu3i4iPhLEbON1hNuUApNdXrgAoo  
HJqgHQchDiSApE  
+dXXxdleBE2am8+7GjS2R5oBU1ZCDp5WfBR2HweVtueaWNXDKNXqkeSARRlutiMP  
404pZJJCHU0RwKanf8EI786kEIL0BEisnsOfOJIClZjLSLmjY5fFI33CZ/  
XmzSCkwMqZD5hWgkOgnIELh/  
Yhvb16Vc0RqXa1Ympx1EoikTZw6TnavA9AWujpERgdh0B+5bAp  
+72npBKJHQZuXrAPqXD7dlDIEldVw  
+BGCKmePgO3v9iHBjzaymF0nCUNB9f3C6q7e6eBG6jsyofz57V9DrOpIRc0GMkKUun  
s7DFwC56dwYGnvpTX9hzGwynF47wFz8BNnLYGx1ScUn64WIXjIOPxxY8Yk6HbgO1H  
0nsOo+OsVRKJmNEbyW3PGpAAhGg9hwFOyj2KQCR/  
WjG4TeaWe5hIE5LAIBG8bvUwi3sc6IN390bIGMBxD9S5RzINxD3U6x4Ld18s4L6awt64  
LzdxX4/zDxj9/IEueAvhAAAAEIFTkSuQmCC) 1x,  
url(  
BMVEUAAAD/gID/gID/VVX/VVX/Tk7/YmL/YGD/VVXzUVH/XV32UIL/W1v2T0//  
WFj3UIL/UIL3UFD/WFjwTk7/U1P/U1PxTU3/V1fyT0//VFTzTk7/UILwTU3/VVX0UFD/  
VFT/VFT1Tk7/VVX/VFT/U1PyT0//VFT/U1PxTEz/UILuS0v/U1P/UIL/VFT0T0//U1P0Tk7/  
VFT/U1PuTU3/UILzTU3/U1P/U1PwTEz/UIL/U1PvTU3/U1P/U1PxTU3/U1PzTk70Tk7/  
U1PyTk7/U1PzTk7/U1P/U1P6UFD/UILzTk7/U1P/U1PyTk7/U1PtTEz/UILyTU3/U1P/UIL/  
UILxTk7/UILvTEz/U1PvTU3/U1P/U1P/UILxTEzXTU3zTU3/UiK7Ozu8Ozu8PDy9PDy  
+PDy+PT2/PDy/PT3APDzAPT3BPT3BPj7CPT3CPj7DPT3DPj7EPj7EPz/FPj7FPz/  
GPj7GPz/HPz/HQEDIPz/IQEDJPz/  
JQEDKQEDKQUHLQEDLQUHMQEDMQUHNQUHNQkLOQUHOQkLOZWXPQUHPQk  
LPZWXQQkLRQkLRQ0PSQkLSQ0PSZmbTQ0PTZmbUQ0PURETVQ0PVRETVaGjWR  
ETWRUXXRETXRUXXaGjYRUXZRUXZaGjaRUXaRkbaaWnbRUXbRkbbaWncRkdbRk  
bdaWneRkbeR0ffRkbfR0ffa2vgR0fga2vhR0fhSEjha2viR0fiSEjia2vjSEjibGzkSEjkSUnkb  
GzlSEjISUnlbGzmSUnmbGznSUnnSkroSkrobW3pSkrqSkrqS0vqi4vrS0vriYnri4vsS0vsi  
YntS0vtTEzuTEzvTEzwTEzwTU3w6OjxTU3x6OjyTU3y6Ojy6eny8vLz8/  
P0Tk71Tk72Tk72cnL3T0/3cnL4T0/4cnL5T0/5c3P6T0/7UFD8UFD9UFD/  
UIJWZWgAAAAYXRSTIMAAgQGDA0NEBUWFhwcHR0fHyAgNDQ3ODg9PT4+QkJD  
Q0ILS15fdHR1fHyEhIWGiliJiYuVlaoqaurrK+vuLm5u7u7wsLExMXGxsZM0tTU2dna2t/  
p7Ozt7fPz+fv+/v7+jD+tjQAACYhJREFUeAhS1cFqwJAcx/  
G1FR0ilqKIFFEUEHKJQKIIRFKGUilSKVCmiHrKpCDuPHcbA99xtjA1+b7HLjmMkaelu  
+TzBl18C/xtFURRFURSfQ6bc6g0ns8Uq3u3i1Wl2GfZa5cz/tKQafW  
+NX629fiN11Rij1p3v8af9vFszrpRTGWxAZTOoyK8pdpZgsOwUpeaUxgSMYlgkLac6BZ  
dpVUpO3QM3ry48JzdClqOc0By9vUVC27YurscMIUBoiprHlhCCWEJGyvsQxhfwk5oxBl  
qbCXM0m0AoYmuJrqgD4RyDvyftQgl3zduTDSBFcMvXU4ggSVTg2ieCNFGW4/8EkCh  
g/keGC6lcg61Hc8Dg7cc76DkaU5ANnqAnliKb6V4QrqB7liLcCExyMfiCHp4/QC3O0/  
boPniDDi8MRb5OGWSBN+hwOL4yFFi0PSYB/  
OLH0+UTtlhJ9WAhEgSdTufLF2iFNI92ByRa6PzIUPTNih21uFG1cQC/  
Kewn6GWh0BdeSu1NKRRKS2+EUuiF2CKI7UWR3WabNHFmk84k2WSTndmsGyfZMc  
42ziYmJhozbo0hXSVsMYRYFhEEFWt7Uau1V3q+hf+Ts2ZXMsmMnHk+wY//ec55zjn/d/  
Heuc4DmqUg/4pr0XXnO+1Zwp2QP5B1LTr  
+D4IHKC3aQ8BFHxn3bXI6U37OuFPKBAUI4U/3b6yHf4TCIqrh3w  
+vwCQVHQrmv4TcYFwJ3THHxIkSYqZLkUXpv7/

EC4QTmrf3J1QSAwDtFh2KZr2f3SOGzQXCNwNCblKx5LpqjvRuSn/dbclbw/  
5fYGQIN6T5EQ6rX7qSnRr8q/fcYLi2vaz876Allr34oIESIEzzb+lizo+EfQmJwie  
+WAQLXQvGk  
+mIEwma7kRvTHxv5dwJ3R7PiAloixHk8mUmnErmvR7fJoXNHsbp3RQkMLRaHw5BY6  
maV+4EJ2e8PK5xg2apaCliJZOJJfVbDarZXNtZ9E1+zfREcINoi0UXJBjMbpiq/  
BouZzWcRYdsQWd4AfNYY4tLOAQiidVdVWjnpuyf  
+UoOmELusQP8gfQ0wuynEBCqwDldHh0w1F0yc5z6CY3aM6POSZIsryYSCtolayXXsg  
bht51EN08ZAM6SrhBPI9IFDDpFxfT6QxaGvkUUEbR6DuljtqATvGD/  
L67oihF5MTSUIrNaDoKHN0oFs2vydQ6ZXs141+yu4IQFumeX1LX1vKaRj2GUQSo9M1  
/vqZd9SAhP0BhKYEOUjJrWr6gF3Tq2TBNs/SYTKmr454Zwg+ax6AXo/  
F4Mqkqa3l4DJYPOGalskum1MwY6LAHIAQkShIGq6KghfI6Wy8EVCqVKpXyt2RyHR4  
D/c8DUCAgRiJR3Dxw9UBLG6yfTXjMUqVarX1HJtaxMdBjD0BBQYhEpEQqqWBs5KjI  
+CeeKkDTRCfHQGc8ANGAZDmeSuGYph7WP0UmqtUatcb3ZEKd4X9v2CUUBAhjQ1I  
ZwdgoIB54hhzkU6k1GvXmD67fHpc8AAhSCQMjXQEIF1f3yiAM/  
SUwUFA9Uaz2fzR7TS77AVIACiWximEMVYoDNeL5VODh3KshmUvujwGuulBKBSOyA  
Ap6iruHRs4DunuMoftzECWtbVI/URs6soY6C0PQO+KmPRoIUx6BASQud8/  
aCDLagLUav1s9wM6BrrhAejJe1EJPa2q2bymb2zQfEbx1Fk  
+8LTaNqlbXoJePt8XxZMJVVWYGKvY88P1QjtXWf8AZG21HrTbrV  
+cQTxL9ur5vuj9ZAoLltHzBlasshcQPA14miyfdvTeeawZJxN/  
fuTA6LIJrVADKKFRQ81RryoR29hQKo1elsbz9zbOrLxCPRB  
+oqNplxv1iulPfyozrhBoMG+WwD1P3VadtJF6JPqRzo3Afoio7EJt1ixaLp418HnZR/  
xZdHAOdJ96JNGx6c7NC46Ee1Kh/Oggl+XS3d3ZekAN13nm4cog  
+Wsce29xkGx7FNrw1BIHzcHunu7Pte/Ri6nA9SbwTPf14vWCW2X5nG77J8kE/  
dxAPDajX6/V/I6N6bQx0jHgp  
+sQslsusf5qsfSilctoUNOT0+oN90THuK6yD6DNcERusfxoW2+5DD2sfgPr9/mDw  
+OXkK+wM8Vb0eam6Ny8sxnkAUBugIQee3mAw6I9EM/  
bPIG7R0wOien00MFp0YAw5XfRzr9vrPRr0wdnd3X016RmEh6IH9ccB0ZeUM1qvv7s53  
9YmgiCMx5S  
+KP4RFUREBKFaFSqiSLWCVVSwBdEKlr5QP6lgtYrYiNe7unpudleWpHlrxdRv4cwOm  
yYVmSudg6XzCX7MPnN5sjsz+P0BoCXlz+DAsizPikB0T  
+avNEP0mvwP8dD3Bw5smWgQBzJUFCFHV5jLBiGiN+R/IPD3Ar/  
PpGcssCz7nCNPdQGXTYw1zEiRG/j95n0PBB00DPyKOA9QZdx/  
A2XyBH5H9IPwCEcg4FFs9LIWtaqwfMIZ4g0TvSc8BBQbeiniE/GeSnhNDmotyIJ09E  
+UEBYX0BzUfAQQGhfABIK21OyV0L80QfyP+8h1hGnhbVF  
+AUOfJobZ6NSV6c80Tkf0jP+H2m+qIEKeAxVyWeFpj4uUXUXQk/YLG  
+SM5ZEWhK5DHHG/+Jx4JEv4ZytEL+B3FIPwWEWqPzMuaR8PMUT9T9tBR/  
UGO5F0E+CnisPS/zgMcTdbelVsn/oH5Az3mQDyWobV8cEX/i5InWV1sD/  
WR4XsRjMEE3+UfgWojl/  
2SxvEJ9tYHHHZN7JueJ1iNQ7yv5n3BeOeGUxgLPrRoacXii3rfof0J  
+FMonCNq4E7KtFjwR8Qz5nyhnAzTO3ZVvRuGJel9G/  
A8dmGnjgbnTDSZuyBNhfkB8T6mNBhxnbn8u39DEE33/x/9oAwFE/vn+Bhvn/  
kjH73y7/9EWcaz3Z+tpiuOiX2zzP8biiXn/sCnfNliRaMT/AJBzwPPypHxjZVUiNex/  
qL6cvyTfelqdqBzyP6Sfzv1mXc25IYj0wP+0rYPwTw/W1r5cjchE/wM8oOhXZ  
+pr8K5OpCk9cF7+co0t8JWJgn5MILq2L4EhgT7yBBx/eyyJMYq+Czw/OgvjiQya9G3lz  
+JEMqM4/Y71ncXDCQ0rbXb8wkRS41ybd8bTH3hLfyQw/aHJusZKDyU/  
eLsHRpMpSVO7H96easqOt8/ujmf2wF5fAJDgioQEIOjQmo0deYAnF46mv4gk/

VUtKS2zkV/3Mzk9Mzcf1/3Mz81MTwqt+/kLc5W5R5JoGz0AAAAASUVORK5CYII=) 2x);  
}

</style>

<style>/\* Copyright 2017 The Chromium Authors. All rights reserved.

\* Use of this source code is governed by a BSD-style license that can be

\* found in the LICENSE file. \*/

.captive-portal .icon {

background-image: -webkit-image-set(

url(

5HvMAAAEyEIEQVR4Xu3afWhVZRwH8LvdmjVSjKyQbKazia1k1sL6YzXoZdy28/

J8f5XRC7hwEVk0hZQCXY2SipCgYgappXMq5Ya9ELZqoEM3VHqxBkapm21jOBWmb

Lfudk+0Peeu2+mc3zk9514WbN9/zz18ds75Pc/vPM+JTP9NZzrT

+b9Fn2nGxHrainb8ggEaRgLn0IOjaMFroloU/

Ssg1yxBLZpxBD00QufpODrRiFX6liUKXY8XcRAJsryC3nRL7EpsQJ/r0V3iSS0/

MKU4D6uoQwLY/P1q4i0aYX9xVqyPRH1jbr8UtfSbE8CDqIK60wBjOEb7aDf24CBO/

gN12LjJF4dM

+dPAoNgMdNkUxLENQItAFViJdbTWfFTcaRbiMXyCpE1CM4vR5uAzdwB/

hcxCDI5jXqGFeAadNJoGH0ErVhg3YPs46Ad9JldNS3GCAbDPkFGG/

eJGWktn3PA4IUB3oxPzGY4AXSBLCSTLgfZxx2MLV1Q5tFHeWxWQDNalHulWeo4q9

GW4VVSshng5J0KgZ8+bkYpcEqIJk6H2y0CiW4H58gGMYQhyn8DIqUUltZIk1XF01kKU

W5yhm3gVB3Q76MDaRyXBQR1aIIJnyi6jNZUgs8uTQfUiGDJlxZ9OPfxEwhAO0F70To5

NR5skxrsUgWaGDZLQCHBHvxnKxWjwuivRS7McK7na1kpUZkEwUzXbFiZfZcVk8pE6

xZ3u34AV5TFJUsv2ObBHU0+Q5wu0eJ21krw

+eD4lzBvOYf7wLeyl5DEfLx2l1DPqoSXK8Sltis/gm41mlago/

9P2UApklZE0pkHhdAv6YliB8NwEQL9HvUwCkzZH15g5Gw8jmR0QtsizvOoACUjAIdl

OZQUkHpCgDqd1gwRslhX3djZAmDsBwpATJPtDUZPqGFuyUWU453IWu7fFHZOvhjiQ

+aqiDheQ/bJjXpf2oP+caRA

+cgPJpkxckeZfiHMGmirC8gee4rz0qF6KV3lJEi86R8kg0oazQ6lv2Uyoiajt4x/

qJ1BfcCe6JQZC/BQ82XvDLYHJPUEK3tmYHTmlotpbzCSwsDomDpc

+mD6VgHETB3M5Or2c5xUBvGTq7P98DjBEjqrBOLbD2eDZj7o+aJdhnhooJxUbd/r1cJ

+yl1mJMMBGWUSP6zlezT5SlirGNKacEB4bwKEXcxrEN7gThUGSCtILaZXMC+KiGN

+5kHYITndkSj7Ko3GTIPMEvtJFKt9LDYgiZtVQTLnqUPUOJcV8KWE95df4m85ps0+CQ

/ig68xN72lscF4yveCFTa5g9AXnDR5lczC1Jr

+4UhuCsAv6YIH3EDURFbgPCG3Y2altmPG9FIJ8LnoOYLbXNDzgi

+Myvkqii9sluocAHZZuN9tAKi6hnaiBwka9f942y2rzFcut4tZOO

+3r5J6xcVmYEEKc9xIPuC3FhDHynBA6LQ5GGA2ftnNI82RqDrIDgb1ZU4AnxzUTc7raD

eWhwPCUW0BA/C7gYeP9cWqILSUX8YAgmxxloF3tYL/DkK9Y/znwm8CI4lvRLV

+dVAQThOpOPht8p+ogZ4W9/jijFFD5eXhWPgPCXjMh/

pSVYD6pxaywPGOKOIBWfkYhXYKozgvy5/

rYBva8avzcx19sSrgT2jJnjJMVMiUAAAAAEIFTkSuQmCC) 1x,

url(

QVR4Xu2de2wUxx3Hfxhj5dkSlgOhpICDWiIH1AjTGEwdnaCNdbi+2/n

+EgIKAvSBopRIKa+KSC5tCU

[illegible]

ARpgpl8xTZBMpOOMq2ICLF80gS/  
tEtQLF9qx2TqHXlfcWmbewShM1Cz3HNeTA6iselk6DicXKuhxL9BXUb00kQ/  
ij0xJ2x2ksy0RodlS6CoqPQKgRVmFnUuyFdBPEGqTw6yciycHyOMekgCGMSipxV5jYW  
els08G3h+HF+w5mV+jQvsR8lVmBwawoU2SNIXjjJTCmAogTB1WY3NznlfYm//  
6sYsd5tJ83wbXxKCnKmkx6Jx0BgBXkE//  
BbkHsdDy9lPdhqfIMldHg4KUCkc2+OFnDjrk  
+IXRpI9alq3z6cJh5AEnty8E4aQoVMltAU9LmWqJ0U3e9OvjeLcvFDmzqEdQmSCzLpx  
JAoPbBGKxtyoUWv0XhL9RD2Cx1IM4HiWB2Y0m272N/  
mp2Hwg63uPbOLGQuZYExrcqxUVMJA/gZ9zudw8RJYCJiaVw7Hf9eelvdovG6HiP  
+4J86K+ixBZTNEo9al7dT+7ob5eMkx43GR7kRPhVHMFvsKX/  
RU0Y2Ti1BBxl3JA6lbbtUKRJSGL0FObqA/ryAWjW7bjZHS8rYKi43HyJj1bXMIR85v  
+oxETbRSEiWi8Sc9ut6jfl2MjcNGZbpsgZ3oPTVyHoneTBjoHj7TzIpsE8aKbJxOo1zr1Sf/  
oGIQWDrND0PWgQujRylcaO/zotJqilcjP65DG02Py+Cx0YEX4NtsEYbfO2GP  
+ALZTKLJJELbov7nMH+H3FsbYIAhxrNOY9/h5CCQ+5w3RUf0rCM26QYXPx4iilTcWj  
+svQdhvliT1/SBadGBzLL+vBSHOa00mNPw/yvgEL  
+ecPhNUJ7OF9jxHS5MfMYF9vMh5mAb5KqiFI4YzyVYwGttSGDq3qQVqQu4Q84KwDa  
PJDvQP5EcH6vltrDemp1qEzHYTK+jjAwOqYgU00lhOQoX/  
274jjgrR3zPgdsYvQb1vcupR4oylgQ8mo8xwb8d5lGEypROhLDVNrUSN3nan6ECNWq  
mmhbloXQnf5US4lKvR2CsxjVzNpU5ELFhKd5yhsTw1F6u5nKtQw7VoQDO3cRua0cC1  
qOEqlsdqNTEw5wwla/gvpXzJeo7GTncAAAAASUVORK5CYII=) 2x);  
}

```
@media (prefers-color-scheme: dark) {  
  .captive-portal .icon {  
    filter: invert(1);  
  }  
}
```

</style>

```
<style> /* Copyright 2021 The Chromium Authors. All rights reserved.  
 * Use of this source code is governed by a BSD-style license that can be  
 * found in the LICENSE file. */
```

```
.https-only #main-content a {  
  color: var(--google-blue-700);  
  text-decoration: none;  
}
```

```
.https-only .icon {  
  background-image: -webkit-image-set(  
    url(  
VEUAAADcRTfcRDfdRET/gIDcRjr/  
ZmbjVTncRDfcRTfcRDfdRDzgSTncRDjeSDvcRTjbRDfbRDjeRzvcRjfbRjjcRTjcRTjcRTf  
dRTfcRDjdRTjcRTjbRDjbRTjbRTjbRTfcRjjdRDrcRjfbRTjZQzfcRDjZRDfZRzbWQzXXRD  
XXQzbXQzbWQjXYSDvWQjbbRDfOQDPSQTTUQjXCPDDNPzPJPjLGPTHVQjXMPzP  
RQTTWQjXLPzPDPDHYQzbAOzDTQTXHPTLIPjK8Oi++Oy/
```

FPTHEPThPQDTQQDTUQTxBPDDKPjK/OzC9Oi/////

PQDPRQDS3OS66OS7TQTTEPDHXQjbMPjMBhLaWAAAAL3RSTIMA4tgPAhYFCcL9  
8B4x9ie1+s49WICbqXNKZY3pujqcgVdLZnL2qKg9zmXpjfontV8LANsAAAJrSURBVHhe  
7ZTndulwFAY3ARlgBAg9vW1v173ROylby/u/yso2Fx3MNaxs9h/zAHM+Sfa8+M/s2LFjx  
+3tdjwH+/

sHWxHVAerb8KSyANnUFkRXwLiK78llgJHJxRalwSMd11OGOeV4nsM9FO0dxhJdw4Lr  
OJ6jYy46PoohqgEHatE9JViiFNWTPiEITpIRRXcQ4C6aJ3EJAS4TkUQXsMJFFE+  
+CCsU8xFEBSAoiHsaQNIQ7yuQCFe3DiHUhtKllzdKoRSFe0r8sXDAkSoumkligYaIOklf  
eWi56EESFm8r1w0fFI4epWgBA9qOMpmirCfeWijtoa9WSx6taAELFBRI/  
vilS3BJRlBk9/VFTsLrifUXRuNfXLU0y/7m6p0CKxqN+v6IJU/  
k3eJxu7Os5LWKDHI1tYstKG1zON1X3DGiRMR80Mx3fdCbc1+bQe3o2SJrYXcV0fFMx  
L9xXiz0987BBtux65qaCeF8IHCR3FabBTQ3xvk4M1yN5B/  
Mw2+urew8hTP1BM38Qnu5evK8gMw+7lcfH9E3ZIEBfMSO//Kf35+Cm6ua  
+rhbSYDeEa9CUyW3qK1Hlj5DBz8dWd0bWCd6Ult/uMPER+BmbV/JHrVG/  
a9MsEyvV5fsK50R3frmBFXtCtVXmt73H4PhQ4t9k9rkJ55tYXwZrO4rCEUfPHfUEcuaZC  
/umw97TfaVpslu2tCb2lRWnBlKFtf+huwrjaa6Pxx7RfgW7nubJPtKI/X0puQO4k/Pfe/  
ovtLY7KbxVwve0/sE3VeaLoslBkEDvt8Hoq/hKGwQYvoq5OMnoq/hLABgc/  
FVn33PX7pAfE5QHR6fAAAAAEIFTkSuQmCC) 1x,  
url(  
BMVEUAAADcRDf/ZmbcRjrjVTn/  
glDdRETdRDzZQzbXQzXWQzbXQjbWQzXZRDbbRDnWQjXWQzXYSDvbRTjcRTjbRT  
fcRjfcRTjcRTjdRjncRTfdRTndRTfdRDnRTjcRDnBRDfBRDjBRjfcRjfbRTjcRTjdRTjbRjfcR  
TjcRDjRjncRTncRTndRDnBRDfBRDjBRjfcRjfbRTjcRTjdRTjbRjfcR  
WQzXeRzvbRDjXRDXXQzbXQzbbRDfeSDvWQjbVQjXIPjLOQDPXQjbCPDDNPzPUQ  
TXRQTS5OS7QQDTUQjW3OS7SQTTPQDTFPDHJPjK2OC26OS7HPjHOPzPLPjLMPj  
PRQDTGPDHTQTTEPTHLPzPGPTG7Oi/HPTLKPjLTQTXyQza9Oi/  
MPzPFPTHDPDHBPDC/OzC+Oy+8Oi/AOzDWQjX////  
bRDd3undHAAAAQnRSTIMA2AUWCQIPHj39wvbO8DH64ifqqYFmtrVMc1IKS5x0nY6P  
WKqbjYDpZXWCZ1py8Jv9McJXV  
+KA9qioPc5l6Y36J7VmcHe8AAAFWUIEQVR4XuzWS4rCQBSG0euz56lSgiaEjHwgGh  
AhDnRF3/6HDY1la5WPjP4a3LOKY28555xzzjnnnHPOuSyzpPR7vb6lZAUrS8hgB7uBp  
aMEKC0Zhz3A/mCpaPjTWCK23GwtCcMjN8ehpWDN3doS8HPi7vRrejX/  
1CbX8qA1sdGZB  
+eRaW14sjGp8YQnk7EpVQQqE7peCFyupjMnYm4yGVGZ7q1EyTZbEEche2uUblMIL  
5W6t4Zkm22lkm02561c89aQbLNTpPgq3hqSbbbmo1r41rhW8NaAaLMzvjITvDUg2WzF  
lyrBWwOCzc6Jkm12QQcL3Vtlmy3opFC9VbbZJR0tNW  
+Vbbahs0b41rhc8FbVZqdEyTb724t5/bYNA3G4e  
+80NYl0gGFkvaR779KKZUWuFKe7nllsT5X//2M5VMZiZB9DQj74xW8ffrwjP90Mb/  
07Vf5CbXYJg0BtO4toKS9vhYHGY1vDZg28FQY6tBZls8tYBehwNLtyt1nhrTDQaDQcW  
Aux2SJWAXpOBpWMWSvm4q0w0Gg4nFQqFTd/m72HIYBYQJV  
+w83bZu9jRaDJYEB4osjJ02aFt8JASUBRq  
+PlarMrWBGI8lQajVanXA5kopUcvBUEGrCAWhSoXs3PZtewKhA/  
MMbTbcpEa7l4KwwURZSHANnVnGz2CVYGmg6oZ1u1XGy2hNWBCA8BogE1m7ZI  
+ShNVMrdW2Wg/v  
+Amr2eRYCcGLBZU2+FgcSBESDfdZxdwGbNvBUGihKgnk1OjPAEwS5gsybeCgNNdT

```
QLyAtqtRCwWQNvhYH4ndjtNnlAnlet1ulQsFI9b4WBpgNyaUCEJ45DwGa1vRUGanU6n
McmB+ZSnlosES3nvm/tUpGm1tFPd5DDAYKFBjGpzRaxSjW5J0o8/MAQ4ZEyKua/b
+0Np175bIMERDuaECFBZGqzBaxY9iAjIMbDK01U0OVZxcplE6BljLzFRixgQDwflCJaX
cC+1ToKyOYHFvCOljPiNmvrTBRI
+oQoGTk2Z1YQyleiWhIEftWnx8Yf8RcyiMCEkyhic2u4xOWSw9MBBQENTQFI83a
+iL2rdgpJ1rms45mByYzhhDNwt6qTtTIQC7r6FT/CLRQ02ZLWKc8OmK
+LzooCykhKpl4q7p+7B/d0SjNggRbqGOzm1gPqL3PX3niZakOQsenf1PDWzWAxr
+JBtEDQxnnJTISNmvrufBK75t45bORBNGSobcqb9DqBCjdQOI5E370xthbYaDRiljRDx
KQwJk9a+o2u431gYZERBo/kcBlfvJ/TrSt6K1b+kDUHMkra2V3j5zRlprNbmADILbQ65S/
z2ggyY82zL0VXsdQnnLdhSOKQzWbLWADIMpDgOrd3q958QiigrG3wusYzmNbXmY4s
h+tangrVJ2Dgy97X9v0CmlLzzIHcj3ZPTL
+h6DN7mhYR5nxHI4mtKNbLCamaX9QDDKFO6C36hDttcdJQFGLLeTWRIlupocGOj62c
Bb9WqesLTFwfm000MQgqz9IDLW
+Hve35HM9Fnqw9HetBkNsF6+Yaet8Jf0+xbka0XbYspSMlg
+5D8/8psnqdYv3qso1vsS9Hy6SaGQ6AYHP9ngLdqlIVpilB8RygRQjGEdOsc4K26RGzk
6YTxjhbDDdzXcfwC8Fbd8glPnR4Y62gBAM/
a1WybfYVNyyUBiZFPXYCAH70GvFW7nFRHH7Egyl8uAd6qXZ7NAqoilG6ZKuBH184D
3qpdAQIIWp0p9dE7wFv1q8Y6+njLoPI
+9P4C4K0GRKSjgTyywvoAeKtBxVWU6YhorovcvA14q0HtouwU0Fw/+jzN8w/cQ/
zg6ug2/QAAAABJRU5ErkJggg==) 2x);
  filter: grayscale(1);
}
```

```
@media (prefers-color-scheme: dark) {
  .https-only .icon {
    filter: invert(1) grayscale(1);
  }
}
```

</style>

```
<style>/* Copyright 2020 The Chromium Authors. All rights reserved.
* Use of this source code is governed by a BSD-style license that can be
* found in the LICENSE file. */
```

```
.insecure-form #main-content a {
  color: var(--google-blue-700);
  text-decoration: none;
}
```

```
.insecure-form .icon {
  background-image: -webkit-image-set(
    url(
SR0IArs4c6QAAB21JREFUeAHtXF1IHFcU9ie2bovECqWxeWyLjRH60BYpKZHYpoFC
U60/
xKct5ME3QaSpT6WUPEICEXyTUplojfgTUwshNpBgqZVQ86hGktdgSsFGQqr1t9+nd2
WZPefO7LjrjYzCJmZc8495zvf3LI3Zu
```

+dzcoKt5CBklGQgZCBklFMZSB7r4G3tLS8sLCw8D7ivo1Ssrn5WYL9AZSC7OzsAuyzl  
HuCHcsjyOawZ7lbVFT0W09Pzz843rNtTwhqaGh4ZXV1tQFZFYZSDgKe85MhyFpBvTso  
V/Py8q5g  
+9OPn0TqpJSgurq6CpBxFuUEQO1LBJgH2zUQdgPlwuDg4LgHe18mKSGovr7+2Pr6+j  
kgOuILVeKVJnJzc78eGBi4nXhVe42kEtTY2Fi8vLz8HVrMKXvY1GjRmvrz8/Pb+/  
r65pMVIWkEodV8vLGx8SPI2Z8scH78gKTFnJyc02hN1/3Ud9ZJCKG1tbVfwnEnyMlxBp  
DOkcQybG9ifwv6OezvRyKRv5eWljhyZeG4AMcvweYNnHKkq4TNcezzqXfbYLSBm46h  
oaELbrZu+l0R1Nra+vz8/HwPgH/uFgj6xwA  
+inINt8Evvb29Tz3U2TFpamp6EbfvR4hVhXISisldpXKAWJeLi4tburu7/1VMXMW  
+Ccll9TKA/oTyni0KQC5B34V9J0abRZutVx1i70fcDti3YR  
+x1UPcSZRPEfsvm52m80WQaTm3beQA1Dr0F9EffANwDzUAu5GDqIPo975FrGbEytV  
8QT  
+JlnTMT0vyRRD6nEsAZLutOlPUDw8P86Eu5VtNTU05goygFGvBQNJI9ElfaHpNrrKuV  
WCHDHLOanoAmUKr  
+QBgZjWbZMtnZ2cflpWV9cPvUZRXF9vHT58+OnMzMzvil4UJ0QQh3KQ8wM8iS0P5P  
SjVOGWWWhCjpVCIXJ+AgD6EeA2ITAoFbB  
+CyKnp6en7kl6SiYIKhuYhcBYEic85JAethu9bad/Qyq8Ap/  
iwCpyLGEUPeX2Y9PTcwozNE7JGzhQCn0k7MwYAsaBMSXh4gZmLpJNknIqQebe6JT  
mAbB59zru7GanQyW5KvtHJe8ln1TUj3B/  
QiR033t0qvby7eWpB5sUzDgeu0jqE1bshJ85pkgQGU7XBG0dVy8lp6EoQrkQFKolv5Wi  
uF/dqKHcC93JObMS02B4xuSnqbbErQQggDum4Mkt8CLR6D4CSGIIvgqLIFmtrJYi/  
BMIJf+yStq4g3lpOoAZj1POc+bGHCvdVGyLaGVI5TQMpV8C+eLZGXUS9L3B+lJAuc/  
8FCyotkVS8jvGcFwNInfOoweQj  
+LKJOXFKz53M1pFMdn2xlpno1Hklr0e8XdysYXRp9qCOPsAPd9x4jYQdC1OGHCBBX  
O5yVXMQCWIUzNgPG72AYGW  
+XuO6C3AQmImdidE5mimoZyqrXOVIG5bxW3weHNRH/  
sinOSBgExE7sSWsyVtjaCSiRnuAraE7VkhiiZBbuYK8GrBIFtsRKC3AtU1gmA0bBrudK1  
bRQ7oMR  
+oMh9i1PxLqaA0bBrueotCAG25smdgTj74JRlyrkFu5gr81JvMTRHsVJ0aiZTSInFqWHX  
crUSFOv4WT5WWxA6rq1JPCc5nNRzyjLIXMOu5cq8VIKgeWnijGemEOLEacEu5sr6Nol  
eOQPwHGxzOjgjNwt2MVcmqRKEjmtOYUF8PIJsgyYWsVty1QICZiJBuAqVQcvaKx4Ldj  
FX+IVbEHR3pcBg  
+zgXEki6IMulmdgVjGKutFUJ4oJJOFxxOsRVyOcqC6c86OdmZUjc8hnmyFw1/  
CpBZjWpOLcOkqo0h0GVWzDfsa2cVQkyiV6VEkaw5gRECCrJft0y4iVmBUcYo5RWyt  
BXGoLw7Woccy  
+EAE7Ys4DfWiwFgog10yOgmpbZCWI65Bxj44ptdtwZQ4qusCIDcY2CRByu  
+G21tpKEJ3CyXnJOa5KhluxJF2QZMRlRlIm5Oa6htGVIMwljMP5hBKg2SxektRplxEbS  
GhWgEyY3BT1ttiVIJpxkbbkBVeG64tGgnirGUwjBmMcfC0np6Hn1RMua264/  
OUorog4xesMmupzkBMBMb  
+ivCPFAIbPa5k8tSAGwbRJOxyLk4UEgsKVZ4HYiMVCDhdQtXsF6rkF0aFZTf8zgovE8s  
qgnElXSzlth+SckggAtg0sZvgkkVX4Ca1R5Nq+0tJSfq  
+lvWpwbeAJrBW8zjWDEshUydjngJgxFA0bR+SvcPEuJYIhoRYUdYz  
+6JIZBizeKIEitD2X9+NqTGp6yluhn8Aw+70ZTSym/lX0zRiMxZiaJ2IIZk1vk/  
tqQQQlC0GnCDZmqQs/ZnFjyOjRJ/n+HArNn1PZDzipF5234uyD  
+YH9dXS6b6Jk5udQsfz9Xz  
+o89VJxxITPeazBR7ADqFF8JuJtGyMTQyJPOe4AfXdSdscm4Xn52AjLh



+21fWpy4yPep3JYaSrQP+Rys/  
Cx9BqzuPhb9wZO1nnKWIBTnDhHws4GbGcZ9pfU1hSCVUhAyEDIQMhAyEDAWfgP5  
qNU5RLQmxEAAAAAEIFTkSuQmCC) 1x,  
url(  
SR0IArs4c6QAAEp1JREFUeAHtnVuMFkUWx2dgRiBhvUxQSZTsw25wAUPiNQTRgFkv  
8YIbZhBcB8hK2NVkXnxRY0xMDFffJkHsyxskBFRGIJ4iWjioLJqdL3EENFZ35AELxnR  
HZFFBtjff+gePsbv0qe6+vv6+6Y66XR39aIT5/  
zPv6urq6q7m5rCEhAICAQEAgIBgYBAQCAgEBAICAQEAgIBgYBAQCAgEBAICAQEA  
gIBgYBAQCAgEBBoTASaG9Ot8l6tWLFi4sGDB3+P1HStx44d0/  
a85ubmyWwnHz9+fHgbHTdxPEj6IMfD2+j423HjxvWTPryeeeaZX65fv/5/HI  
+pZUwQ6l477vjD0NDQAgiwgOBfynYa23E+l43OY+jcy/Zjtn0tLS19zz///Oc  
+y8ijroYkUEdHxxSCuBDaF7DOZ/  
+CWoAPmb6m3J2sfexv37Jly3e1sCPLMhuGQF1dXRP2799/  
G2TpBLCbWFuyBM5B9xB5XoVIPVOnTn2xu7v7sIOO3GWpewJR21xJG  
+ZukF3MenbuEC5u0A8kb6YNtY5a6YPilvWRWrcEWrx48Xyl8xA1znX1AXVxK6mR3oBI  
qzdV3qxbXd0tdUcgapybIY2IM6fu0C5jMER6j3U1NdlrZcRyd6puCARx5kCabtbLcoeiR4  
Mg0UesXRDpPY9qM1OVewltW7asjT6bJ0DgLGy5t9dTpl6j55/0Ld2/  
YcOGAU86M1GT24BQ0zS3t7evxOvHWNsy8T7/  
SkWeB3t7e9dSK4IUuVtySSBuV9NoId8LWnNzh1htDHqHhvd3Nb21qb40qV67Y0tXUz  
yMzxd3Urt8wk5AnlOwjZXmAibk0n52MtNDbRq1arWgYGBx4HlvmpAwy3hJ8rpJzD98Zg  
W+1+RPjh+/PjB0047bfDQoUMa+2o6/FTTJ//yyy+Tjx49OjxOhsxFJA+PobE/  
PJ5G3kmSrcLyZFtb2wNr1qw5UoWyKhaRCwltWbLklsaqthCEqypa7CggwqD/  
bbZ9bPsuueSSTx955JFjjupOyYaecbt3756Nbo21acztGraZEQR97zPW1vHcc899dYohN  
TioOYFo78ygvfMavl+Ygf8aQe  
+IhumZMWPGLGkT4YTMF8pp2bNnzzz86oRI7RS0oX3fyz78uoF20R7fii36akqggG/  
nZUA+12J0JVII8zrr08htA+BDleSzPM+t+YwDBw7cjo/LWa/3WRY  
+fs96Sy37jGpGIMhzM1foZgA9wweoAKnb0VbaL6uZRvGpD52+dTCtZDbtqlfQuwgy  
+XqA+ZmaaDEkqkkPdk0IRP/  
OnwFwPUCmHjGPIPNMa2vrY5s2bfrCd9Cz0Ld06dKLjxw58iC67/  
JEpCFItBwSqueujqkvVCRTVPC/gpQ/  
yfEgA7tm6deuHVUXNU2GLFi26nAvgKXy43INKkej2atdEvqrRRP6rzRPdtIKRB9APANa  
9s2bNuqpeySPAZL8kC/yKRGlpYVahK0wLi3i/  
0zVaiAcm8GVtos1VYMZoHfQL7O8p6fnW/  
9w1E5jZ2fnefQ7PQ0+N6axAnzUsJ5HTVSVp7OqEEj9PNzz3wWYNI/  
qqqlfZt7MEwCUy3GhNIFXXsjTTG/z/  
dQkj3KYppbeN3HixDkbN27cl9amSvkzv4Wph1mdhBiShjqzq85jPVfV4o5JHgZJv8IG  
+cpgm+BcePny4V9hLb5ZL5gTS8ARXVpoe5k8B9AqA/  
VeWQORJt3yVz9jk3B0hzKOhoUxdy/QWpsE/+j1edPWAK/lt1oUA  
+qOrjnrOR7vxLliwnfVaVz/oF7uN2/5Lrvkr5cusBsL5adzL11cyoNR5iLNt0qRJN45V8ggX  
+S4MhEUpnCqIKwaKRSU51/  
OZEIgrphnDn2Xr9MQlwFg7xuKbnqMDKQyEhSuJFIMoFpncbTIhUDST0Gk  
+D0C9xVWnyVNHR4M5Vo  
+FhTARNo4Yzl1i4pi9dDbvrlzmMPdTpmS0VDWYrx3Lt63SoWpqUpul2kQkml1OrsS5Ae  
ZYT/c9x9p7DRRNghChjx7Vx3Sbp0TgR5J1YQkjElwe8eOXE0b0+djxWgNxxhWio4h0Ms  
+pVJ6H6eWr2qM64IKlzkmeIq48+4jWsA5yvBuedHLQYIR4H57ng7O2Vla81EA22bhwy

A4tTD9eSPMYg1FxcWAKzB0Oaoxg5ZC2exRuBuCr0xuhlxYspnUrDcleGJ0pLhDPFEli  
GdHYUO1cuTTFsRMrJWM55lxCGaaKUaYE8BzQwyTZ0+zAV0qDCwizCzjyK7xKrUjB6l  
RA9zvoGj3kaASA81Gij6qWAZiJd2AlDq27FSjGz5ism74VANOjMTuD4hzNnzvx7MaNC  
WnlEhKGWtJ7jhKRLzIqVkJpA3E  
+vhNGmT6zgsD4Hd4+v12qKOTZW0oShsBSmFp8VM8XOkqeYbGoCYcjKYoorpD1Tzz  
MJK/hW9dMRIs9YC3aM3SnFpCKQPiuHER2naKxwoCtFE  
+AriXTRgSEqUMt1KEYGos6RTwVgfRNQRZPYu3tV7enjgFqZwfRJhuNZp5dhRDY7aT  
4qklhJplJ1UI29N7W8kkg5QVARdsuYPoo6TOizOBalDpU7qmCeBUsa/n9aU/  
ZwRzIFHYCmOjSTcplsY8l+LWsZSRjJBnlQem/  
Dj39liCnO3UcmzLJxTCmNhYXqFuiWK51sUO5xqlwhYYCxxE3nlmbGssSwujlW1ZbH  
GckR3GgKZejK5MnoZBKzphw5GvG7gHWEsrI0ummJZqNuJQNwz9ZKq6fcBjB73FBYc  
9rNDwlq1Yqn/ibhY5EQgusFNjOWK  
+Enf53ExMOSxlyCshbklp35GY5GPZZ0lhHGmwmD429X6uFPs2FjeCmthbsHAGtNYtx  
OBMO7SWEGLcb1JZELMv4QsGJujWlsqZIA  
+lkbxpneM8K4QKAY8SptrZgrpoqt1TwzgfSnP4xLnA/DftlHLA2GBfl0CAhzYZ9Ui2la/  
cUxaZZhucREKNCqz9palv4wbcMClx/  
ZCHO9XmVZrLFtypxAMNvqhMXhIFsGAQfssycQj/CmQuiTCAQqE  
+QsT1mxt8ZWtpvGspSB+  
+r5MFu7SZe6lFA9vReWfHjkTNgrtgbdw6lutzDTR7Mh21dWo4K8HwQcsDfFVla6EMj0  
CX9YbR3Y84Ne0KK7hRV7U2ydCASrTSxlkpPViRB6TwhYsbfG1oIAZDIRSH  
+98YRHUGNEwAF7U2xljvkWRrVoKiT  
+ZZLR9yDuAQEr9tbYykQzgTz4FVQ0EAJmAnGfNN2S9LO2BsKrrlyxYm  
+NrcAwE4g8JgLPt391hXoDGeuAvSm2gspMIoujX4T2UAxqStXrNhby  
+tEIDKZWOrYAfhXUcqsQ7Ym2LrSqDEUwRUAKzWD2rDUgMErNhXpQ1Eld8YsTAN  
vhp1B/HyCFixN/  
8BydwGqsYlB3lMwtmkCFhH162xlR1mApHHOsJrvQqS4hPkKiDALcyKvSm2Kj5zAlHGd  
GbHuZRTAZ5wuhwCEeb5lxBfO/  
8SZh8rZ3zhOdpMk3bv3j27MC3sZ4+AMBf2SutSTBXbpPKxnLlm0M8/  
MGxvrCDJFuMWJJEMLMv4QsGKumLr83MZMILmlcR9bXMW4QCALYB5krZhbYxqb6EQ  
gjDO954Vx13BPNk  
+fjY0MWxsCwlqYW3JZYxrrdiJQS0uLiUAYN2nPnj3z4kLDNlsEhLUwt5RijWms24lAfAnrc  
xj+dawkyZY+iVSfUktSRpA5gYAVa8VSMXXBz4IAUUH6W0zihSuinc/  
CnJ44QxB0QkAYC2tjZIMsC3WnlZDpNkahGpX/  
U2HhYT8TBISxdQaENZYjhjsTiGpvO1qGRjQl2OHKWJ5ALlikQACMVxizD0WxNGY7le  
5MID6l9h0qXrWUinPX8yWs0KloAc0gK2zB+I+GLBJ9NYqlMdsJcWcCKTvMNx+2jklO5h  
+zOHk2BjO5YOSsw0JoUxFo6tSpL6Lsh0KFCfYXLV269OIEckHEgECE6SJDFon  
+EMXQmO2keCoCdXd3H0bV5pPqKu9RxY47cuTlg5Ulg4QFAWEqbC15kN0cxdCY7aS  
4tcCTOaM95pCs+1Vi5YS7+JjB5ZXFgkQSBCls70oiWyjjGLtCFU7TOU5RQAPsA  
+6jb5ySWOFavwp5ngrTPCoAleC0MBSW1tpHMVPsEhRRViR1DSTtMNn8AxUcvvyzzz  
77a1nrwsmKCAhDYVIRcJSAS8xGqRg+9Elg/iC8E0a/V6yAcmk4vrqzs/  
O8cjLhXGkEhJ0wLC1R/lxipZgVP2tL9UlgFYIRZkdw/  
hze39bPQZptZgdpYRZhd44VDZdYISrDG4G4n76CYR+VKqhUOkDcyB  
+E7y91PqQXR0CYCbviZ0unKkaKVWkJ2xlvBFKxGNfF5rjNhKYmRo8fZRDwamu  
+sSovrlSZg//Hoxg5ZC2exfutg0fKtRR1d/Hiyqbuo2F3BVeHaZpIWY0NeBLyXAB5/  
o1rFzq4t47/oq10yFcyi9caSKUwMVu3o4GSJZY  
+cSHA7ACgs0qLjO0zkwYYgYILeQai2HgF0TuBNmzYIPK49jRrMHC7yyf3vaKSQ2XCR

NhgmUtG9INRbLx65/0WJutwtLm9vX0Xu3NdrOU+vY21g9vZUZf8jZaHmmc8mG5h1Vwfl  
+Wd3t7eeWBqbp9WKSx7DaQCZSjtmTvZfl/  
JgGLnBZQACzVRU1NU8ziTRzGIYuGdPMOxLhZAX2k8at7KFAON2DstOP8W60Jqoh  
+dFNR5JrV5uJC2s17r6gpfar2NTsOXXPNXyje+kkCa83Sz/4e/  
5/0GHXMc9fwW8G6aNWvWC7xpYPqsjGN5uckGefS0pTHGq1IY9SS3ru4U  
+StmzeQWVlhqW1vbA9Qi7xemGfdn67EVQMdMP5F8lc/  
g5NpgVjPifWFvxNosnkkjerQVS5YsuYj5Ku  
+S7vL4Gasb4l7+MNXXE4CTyf08LqhWW2rbZvUwQx51EqZ5EXPfxIkT52zcuHFf1r5Uh  
UBygqtKf3rexXpuGqcgzw6+Prq8p6fH/  
DGkNOVmnVcDo9HYInl4otA28PmedR7txj2F6VntZ9oGKjSaNsx3M2fOFIGWkt5aeM64/  
zv+MLwSXf/  
lav34zTffrOvaSPN5pkyZ8jdq6G1gc4kRi9HiP1NL3wh5Phl9lqvjqTVAsQPURDdTRb/  
AcZoqOlandsK9dM9/GCfU01YzCaktNBnMPJ+niJ  
+6xd8OebwNIBYp41dJVSeQLIBEd0Kip9INTSICcAw9z7S2tj62adOmL6Q/74smwEfzWu  
+CPD4eZESe5ZDn2Wr7XhMCycmoJtKE/DN8OB0RaSv9Hqt5z/  
tTHzp969B7W9GrN4s8EUcm6ra1uNo1T4xNzQgkAyDRHIB8mTVVwzp2Jt5CptdZVcNt  
A9hDcXottvio7wGoZ3056/U+bcBHNZhvwUfzbFBfdtSUQHICgGdwO3uN3TSP  
+KXwGATgXq7QHjo0d9FgHSol6DOdclR0iRX86oQ07eie7FN/  
pEvTX26APFV52iplf80JJMPUT8STlcZ70vS6lvJxOB0i/YT  
+t9n2se3Tf9UJtNpPqRc9SembhOhegO4FbK9ha/o+j8UI9L8/  
YcKE9mr081SyKxcEkpGrVq1qHRgYeJzd+yoZ7eM8QdDQSD+B7udK7o/2vyJ9UH/608/  
a4v9t6a83+nEJ7ZfJyE9G5iLkp1PDTGdfX0KdniVh0F  
+4PKke5jVr1hwpTKzVfm4IFAogAVgCs56AeG0XxfrrdQtRNAq  
+IsuBURdsckcgOUG7aBok0iOp03wiFyBynucdyHMn7Z29ebMzlwQSSNRampS2kt3HW  
NuUNgaX4dmdjKivpQbKZY+7j06sTOlqwOhh/gfzeNXGWMeaSwAzcf6Er  
+vkuzDIK3nke25roNGBifqMuqmZLht9rpGOlctHrF217Nux4Fk3Blqdgkg3Q6KHWf0nqc  
WqcrWFNO+xroY4VR3LSgtC3REodpintfk0tEWk6+K0etxCmjdoIK/  
29a56tTGoWwLFQFEjXQmJVrJ2kHZ2nJ7z7Q8QZwvrWmqc1J9YqaWvdU+gGLyurq4J  
+/fvv43jZZBJk7JSj/THuj1t9TVUvRS4QZ+VS/  
tlME82pVbTMAQqRIJaaQokWkjaAtb57F9QeL5a  
+xBGr2nvZO1jfzu1jb5s21BLQxJodIQglAZs5xNEjVVdynYaW69dGOg8hs69bD9m20e7Z  
ieEqelA52gcsjgeEwQaDZxe1jt48ODvSR8ex4JcGtM6n2ONmk  
+CANpqzGt4FJ3jQY41sq+txtAGSfsGkgyPoXHcT5/  
Nly7/2yJvWAICAYGAQEAgIBAQCAGBAICAYGAQEAgIBAQCAGBAICAYGAQEAgIB  
AQCAgEBAICAYEclvB/Q079+h6myXwAAAAASUVORK5CYII=) 2x);  
}

```
@media (prefers-color-scheme: dark) {  
  .insecure-form .icon {  
    filter: invert(1);  
  }  
}
```

</style>

<style>/\* Copyright 2019 The Chromium Authors. All rights reserved.  
\* Use of this source code is governed by a BSD-style license that can be  
\* found in the LICENSE file. \*/

```
.lookalike-url #main-content a {  
  color: var(--google-blue-700);  
  text-decoration: none;  
}
```

```
.lookalike-url .icon {  
  background-image: -webkit-image-set(  
    url(  
SR0IArs4c6QAAB21JREFUeAHtXF1IHFcU9ie2bovECqWxeWyLjRH60BYpKZHYpoFC  
U60/  
xKCt5ME3QaSpT6WUPEICEXyTUplojfgTUwshNpBgqZVQ86hGktdgSsFGQqr1t9+nd2  
WZPefO7LjrjYzcJmZc8495zvf3LI3Zu  
+dzcoKt5CBklGQgZCBklFMZSB7r4G3tLS8sLCw8D7ivo1Ssrn5WYL9AZSC7OzsAuyzl  
HuCHcsjyOawZ7lbVFT0W09Pzz843rNtTwhqaGh4ZXV1tQFZfYZSDgKe85MhyFpBvTso  
V/Py8q5g  
+9OPn0TqpJSgurq6CpBxFuUEQO1LBJgH2zUQdgPlwuDg4LgHe18mKSGovr7+2Pr6+j  
kgOulLveKVJnJzc78eGBi4nXhVe42kEtTY2Fi8vLz8HVrMKXvY1GjRmvrz8/Pb+/  
r65pMVIWkEodV8vLGx8SPI2Z8sch78gKTFnJyc02hN1/3Ud9ZJCkG1tbVfwnEnyMlxBp  
DOkcQybG9ifwv6OezvRyKRv5eWljhyZeG4AMcvweYNnHKkq4TNcezzqXfbYLSBm46h  
oaELbrZu+l0R1Nra+vz8/HwPgH/uFgj6xwA  
+inlNt8Evvb29Tz3U2TFpamp6EbfvR4hVhXISisldpXKAWJeLi4tburu7/1VMXMW  
+Ccll9TKA/oTyni0KQC5B34V9J0abRZutVx1i70fcDti3YR  
+x1UPcSZRPEfsvm52m80WQaTm3beQA1Dr0F9EffANwDzUAu5GDqIPo975FrGbEytV  
8QT  
+JlnTMT0vyRRD6nEsAZLutOlPUDw8P86Eu5VtNTU05goygFGvBQNJI9ElfaHpNrrKuV  
WCHDHLOanoAmUKr  
+QBgzJWbZMtnZ2cflpWV9cPvUZRXFf9vHT58+OnMzMzvil4UJ0QQh3KQ8wM8iS0P5P  
SjVOGWWWhCjpVCIXJ+AgD6EeA2ITAoFbB  
+CyKnp6en7kl6SiYIKhuYhcBYEic85JAethu9bad/Qyq8Ap/  
iwCpyLGEUPeX2Y9PTcwozNE7JGzhQCn0k7MwYAsaBMSXh4gZmLpJNknIqQebe6JT  
mAbB59zru7GanQyW5KvtHJe8ln1TUj3B/  
QiR033t0qvby7eWpB5sUzDgeu0jqE1bshJ85pkgQGU7XBGODVy8lp6EoQrkQFKolv5Wi  
uF/dqKHcC93JObMSO2B4xuSnqbbErQQggDum4Mkt8CLR6D4CSGIIVgqLIFmtrJYi/  
BMIJf+yStq4g3lpOoAZjl1POc+bGHCvDVGyIaGVI5TQMpV8C+eLZGXUS9L3B+lJAuc/  
8FCyotkVS8jvGcFwNlnfOoweQj  
+LKJOXFkz53M1pFMdn2xlpno1Hklr0e8XdysYXRp9qCOPsAPd9x4jYQdC1OGHCBBX  
O5yVXMQCWIUzNgPG72AYGW  
+XuO6C3AQmImdidE5mimoZyqrXOVIGg5bxW3weHNRH/  
sinOSBgExE7sSWsyVtjaCSiRnuAraE7VkhiiZBbuYK8GrBIFtsRKC3AtU1gmA0bBrudK1  
bRQ7oMR  
+oMh9i1PxLqaA0bBrueotCAG25smdgTj74JRlyrkFu5gr81JvMTRHsVJ0aiZTSInFqWHX  
crUSFOv4WT5WWxA6rq1JPCc5nNRzyjLIXMOu5cq8VIKGEwnijGemEOLEacEu5sr6Nol  
eOQPwHGxzOjgJNwt2MVcmqRKEjmtOYUF8PIJsgyYWsVty1QICZiJBuAqVQcvaKx4Ldj  
FX+IVbEHR3pcBg
```

+zgXEki6IMulmdgVjGKutFUJ4oJJOFxxOsRVyOcqC6c86OdmZUjc8hnmyFw1/  
CpBzJWpOLcOkqo0h0GVWzDfsa2cVQkyiV6VEkaw5gRECCrJft0y4iVmBUcYo5RWyt  
BXGoLw7Woccy  
+EAE7Ys4DfWiwFgog10yOgmpbZCWI65Bxj44ptdtwZQ4qusCIDcY2CRByu  
+G21tpKEJ3CyXnJOa5KhluXJF2QZMRlRBlm5Oa6htGVIMwljMP5hBKg2SxektRplxEbS  
GhWgEyY3BT1ttiVIJpxkbbkBVeG64tGgnirGUwjBmMcfC0np6Hn1RMua264/  
OUorog4xesMmupzkBMBMb  
+ivCPFAIbPa5k8tSAGwbRJOxyLk4UEgsKVZ4HYiMVCDhdQtXsF6rkF0aFZTf8zgovE8s  
qgnElXSzlth+SckggAtg0sZvgkkVX4Ca1R5Nq+0tJSfq  
+lvWpwbeAJrBW8zjWDEshUydjngJgxFA0bR+SvcPEuJYIhoRYUdYz  
+6JIZBizeKIEitD2X9+NqTGp6yluhn8Aw+70ZTSym/lX0zRiMxZiaJ2IIZk1vk/  
tqQQXlC0GnCDZmqQs/ZnFjyOjRJ/n+HArNn1PZDzipF5234uyD  
+YH9dXS6b6Jk5udQsfz9Xz  
+o89VJxxITPeazBR7ADqFF8JuJtGyMTQyJPOe4AfXdSdscm4Xn52AjLh  
+21fWpy4yPep3JYaSrQP+Rys/  
Cx9BqzuPhb9wZO1nnKWIBTnDhHws4GbGcZ9pfU1hSCVUhAyEDIQMhAyEDAWfgP5  
qNU5RLQmxEAAAAAEIFTkSuQmCC) 1x,  
url(  
SR0IArs4c6QAAEp1JREFUeAHtnVuMFkUWx2dgRIBhvUxQSZTsw25wAUPiNQTRgFkv  
8YIbZhBcB8hK2NVkXnxRY0xMDFffJkHsyxskBFRGIJ4iWjioLJqdL3EENFZ35AELxnR  
HZFFBtjff+gePsbv0qe6+vv6+6Y66XR39aIT5/  
zPv6urq6q7m5rCEhAICAQEAglBgYBAQCAgEBAICAQEAglBgYBAQCAgEBAICAQEA  
glBgYBAQCAgEBBoTASaG9Ot8l6tWLFi4sGDB3+P1HStx44d0/  
a85ubmyWwnHz9+fHgbHTdxPEj6IMfD2+j423HjxvWTPryeeeaZX65fv/5/HI  
+pZUwQ6l477vjD0NDQAgiwgOBfynYa23E+l43OY+jcy/Zjtn0tLS19zz///Oc  
+y8ijroYkUEdHxxSCuBDAF7DOZ/  
+CWoAPmb6m3J2sfexv37Jly3e1sCPLMhuGQF1dXRP2799/  
G2TpBLCbWFuyBM5B9xB5XoVIPVOnTn2xu7v7sIOO3GWpewJR21xJG  
+ZukF3MenbuEC5u0A8kb6YNtY5a6YPilvWRWrcEWrx48Xyl8xA1znX1AXVxK6mR3oBl  
qzdV3qxbXd0tdUcgapybIY2IM6fu0C5jMER6j3U1NdlrZcRyd6puCARx5kCabtbLcoeiR4  
Mg0UesXRDpPY9qM1OVewltW7asjT6bJ0DgLGy5t9dTpl6j55/0Ld2/  
YcOGAU86M1GT24BQ0zS3t7evxOvHWNsy8T7/  
SkWeB3t7e9dSK4lUuVtySSBuV9NoID8LWnNzh1htDHqHhvd3Nb21qb40qV67Y0tXUz  
yMzxd3Urt8wk5AnlOwjZXmAibk0n52MtNDbRq1arWgYGBx4HlvmpAwy3hJ8rpJzD98Zg  
W+1+RPjh+/PjB0047bfDQoUMa+2o6/FTTJ//yyy+Tjx49OjxOhsxFJA+PobE/  
PJ5G3kmSrcLyZFtb2wNr1qw5UoWyKhaRCwltWbLklIsaqthCEqypa7CggwqD/  
bbZ9bPsuueSSTx955JFjjupOyYaecbt3756Nbo21acztGraZEQR97zPW1vHcc899dYohN  
TioOYFo78ygvfMavl+Ygf8aQe  
+lhumZMWPGlgKt4YTMF8pp2bNnzzz86oRI7RSo0X3fyz78uoF20R7fii36akqggG/  
nZUA+12J0JVII8zrr08htA+BDleSzPM+t+YwDBw7cjo/LWa/3WRY  
+fs96Sy37jGpGIMhzM1foZgA9wweoAKnb0VbaL6uZRvGpD52+dTctZDbtqlfQuwgy  
+XqA+ZmaaDEkqkkPdk0IRP/  
OnwFwPUCmHjGPiPNMa2vrY5s2bfrCd9Cz0Ld06dKLjxw58iC67/  
JEpCFItBwSqeujqkvVCRTVPC/gpQ/  
yfEgA7tm6deuHVUXNU2GLFi26nAvgKXy43INKkej2atdEvqrRRP6rzRPdtIKRB9APANa

9s2bNuqpeySPAZLt8kC/yKRGlpYVahK0wLi3i/  
0zVaiAcm8GVtos1VYMZoHfQL7O8p6fnW/  
9w1E5jZ2fnefQ7PQ0+N6axAnzUsJ5HTVSVp7OqEEj9PNzz3wWYNI/  
qqqlfZt7MEwCUy3GhNIFXXsjTTG/z/  
dQkj3KYppbeN3HixDkbN27cl9amSvkzv4Wph1mdhBiShjzq85jPVfV4o5JHgZJv8IG  
+cpgm+BcePny4V9hLb5ZL5gTS8ARXVpoe5k8B9AqA/  
VeWQORJt3yVz9jk3B0hzKOhoUxdy/QWpsE/+j1edPWAK/lt1oUA  
+qOrjnrOR7vxLliwnfVaVz/oF7uN2/5Lrvkr5cusBsL5adzL11cyoNR5iLNt0qRJN45V8ggX  
+S4MhEUpnCqIKwaKRSU51/  
OZEIgrphnDn2Xr9MQlwFg7xuKbnqMDKQyEhSuJFIMoFpncbTihUDST0Gk  
+D0C9xVWnyVNHR4M5Vo  
+FhTARNo4Yzl1i4pi9DbvrlzmMPdTpMs0VDWYrx3Lt63SoWpqUpul2kQkml1OrsS5Ae  
ZYT/c9x9p7DRRNgHchjx7Vx3Sbp0TgR5J1YQkjElwe8eOXE0b0+djxWgNxxhWio4h0Ms  
+pVJ6H6eWr2qM64IKlzkmlq48+4jWsA5yvBuedHLQYIR4H57ng7O2Vla81EA22bhwY  
A4tTD9eSPMYg1FxcWakzB0Oaoxg5ZC2exRuBuCr0xuhlxYspnUrDcleGJ0pLhDPFeli  
GdHYUO1cuTTFSrMrJWM55lxCGaaKUaYE8BzQwyTZ0+zAV0qDCwizCzjyK7xKrUjB6l  
RA9zvoGj3kaASA81Gij6qWAZiJd2Aldq27FSjGz5ism74VANOjMTuD4hzNnzvx7MaNC  
WnlEhKGwTJ7jhKRLzlqVkJpA3E  
+vhNGmT6zgsD4Hd4+v12qKOTZW0oShsBSmFp8VM8XOkqeYbGoCYcjKYoorpD1Tzz  
MJk/hW9dMRIs9YC3aM3SnFpCKQPiuHER2naKxwoCtFE  
+AriXTRgSEqUMt1KEYGos6RTwVgfRNQRZPYu3tV7enjgFqZwfRJhuNZp5dhRDY7aT  
4qklhJplJ1UI29N7W8kkg5QVARdsuYPoo6TOizOBalDpU7qmCeBUsa/n9aU/  
ZwRzlFHYCmOjSTcplsY8l+LWsZSRjJBnlQem/  
Dj39liCnO3UcmzLJxTCmNhYXqFuiWK51sUO5xqlwhYYCxxE3nlmnbGssSwujlW1ZbH  
GckR3GgKZejK5MnoZBKzphw5GvG7gHWEsrI0ummJZqNuJQNwz9ZKg6fcBjB73FBYc  
9rNDwlq1Yqn/ibhY5EQgusFNjOWK  
+Enf53ExMOSxlyCshbkp35GY5GPZZ0lhHGmwmD429X6uFPs2FjeCmthbsHAGtNYtx  
OBMO7SWEGSLcb1JZELMv4QsGJujWlsqZIA  
+lkbpneM8K4QKAY8SprrZgrpoqt1TwzgfSnP4xLnA/DftIHLA2GBfl0CAhzYZ9Ui2la/  
cUxaZZhucREKNCqz9palv4wbcMClx/  
ZCHO9XmVZrLFtypxAMNvqhMXhIFsGAQfssycQj/CmQuiTCAQqE  
+QsT1mxt8ZWtpvGspSB+  
+r5MFu7SZe6IFA9vReWfHjkTNgrtgbdw6lutzDTR7Mh21dWo4K8HwQcsDfVla6EMj0  
CX9YbR3Y84Ne0KK7hRV7U2ydCASrTSxlkpPViRB6TwhYsbfG1olAZDIRSH  
+98YRHUGNEwAF7U2xljvkWRrVoKiT  
+ZZLR9yDuAQEr9tbYykQzgTz4FVQ0EAJmAnGfNN2S9LO2BsKrrlyxYm  
+NrcAwE4g8JgLPt391hXoDGeuAvSm2gspMIOujoX4T2UAxqStXrNhY  
+tEIDKZWOrYAfhXUcqsQ7Ym2LrSqDEUwRUAKzWD2rDUgMERNhXpQ1Eld8YsTAN  
vhp1B/HyCFixN/  
8BydwGqsYlb3IMwtmkCFhH162xIR1mApHHOsJrvQqS4hPkKiDALcyKvSm2Kj5zAIHGd  
GbHuZRTAZ5wuhwCEeb5lxBfO/  
8SZh8rZ3zhOdpMk3bv3j27MC3sZ4+AMBf2SutSTBXbpPKxnLIm0M8/  
MGxvrCDJFuMWJJELMv4QsGKumLr83MZMILmIcR9bXmw4QCALYB5krZhbyxqb6EQ  
gjDO954Vx13BPNk  
+fjY0MWxsCwlqYW3JZYxrrdiJQS0uLiUAYN2nPnj3z4kLDNIsEhLUwt5RijWms24IAfAnrc  
xj+dawkyZY+iVSfUktSRpA5gYAVa8VSMXXBz4IAUUh6W0zihSuinc/

CnJ44QxB0QkAYC2tjZIMsC3WnlZDpNkahGpX/  
U2HhYT8TBISxdQaENZYjhjsTiGpvO1qGRjQl2OHKWJ5ALlikQACMVxizD0WxNGY7le  
5MID6l9h0qXrWUinPX8yWs0KloAc0gK2zB+I+GLBJ9NYqIMdsJcWcCKTvMNX+2jklO5h  
+zOHk2BjO5YOsSw0JoUxFo6tSpL6Lsh0KFCfYXLV269OIEckHEgECE6SJDfOn  
+EMXQmO2keCoCdXd3H0bV5pPqKu9RxY47cuTlg5Ulg4QFAWEqbC15kN0cxdCY7aS  
4tcCTOaM95pCs+1Vi5YS7+JjB5ZXFGkQSBCls70oiWyjjGLtCFU7TOU5RQAPsA  
+6jb5ySWOFAVwp5ngrTPCoAleC0MBSW1tpHMVPsEhRRViR1DSTtMNn8AxUcvvyzzz  
77a1nrwsmKCAhDYVIRcJSAS8xGqRg+9Elg/iC8E0a/V6yAcmk4vrqzs/  
O8cjLhXGkEhJ0wLC1R/lxipZgVP2tL9UlgFYIRZkdw/  
hze39bPQZptZgdpYRZhd44VDZdYISrDG4G4n76CYR+VKqhUOkDcyB  
+E7y91PqQXR0CYCbviZ0unKkaKVWkJ2xlvBFKxGNfF5rjNhKYmRo8fZRDwamu  
+sSovrlSZg//Hoxg5ZC2exfutg0fKtRR1d/Hiyqbuo2F3BVeHaZpIWY0NeBLyXAB5/  
o1rFzq4t47/oq10yFcyi9caSKUwMVu3o4GSJZY  
+cSHA7ACgs0qLjO0zwwYYgYILeQai2HgF0TuBNmzYIPK49jRrMHC7yyf3vaKSQ2XCR  
NhgmUtG9lNRbLx65/0WJutwtLm9vX0Xu3NdrOU+vY21g9vZUZf8jZaHmMc8mG5h1Vwfl  
+Wd3t7eeWBqbp9Wksx7DaQCZSjtmTvZfl/  
JgGLnBZQACzVRU1NU8ziTRzGIYuGdPMOxLhZAX2k8at7KFAON2DstOP8W60Jqoh  
+dFNR5JrV5uJC2s17r6gpfar2NTsOXXPNXyje+kkCa83Sz/4e/  
5/0GHXMc9fwW8G6aNWvWC7xpYPqsjGN5uckGefS0pTHGq1IY9SS3ru4U  
+StmzeQWVlhqW1vbA9Qi7xemGfdn67EVQMdMP5F8lc/  
g5NpgVjPifWFvxNosnkkjerQVS5YsuYj5Ku  
+S7vL4Gasb4l7+MNXXE4CTyf08LqhWW2rbZvUwQx51EqZ5EXPfxIkT52zcuHFf1r5Uh  
UBygqtKf3rexXpuGqcgzw6+Prq8p6fH/  
DGkNOVmnVcDo9HYInl4otA28PmedR7txj2F6VntZ9oGKjSaNsx3M2fOFIGWkt5aeM64/  
zv+MLwSXf/  
lav34zTffrOvaSPN5pkyZ8jDq6G1gc4kRi9HiP1NL3wh5Phl9lqvjqVAsQPURDdTRb/  
AcZoqOlandsK9dM9/GCfU01YzCaktNBnMPJ+niJ  
+6xd8OebwNIBYp41dJVSeQLIBEd0Kip9INTSICcAw9z7S2tj62adOmL6Q/74smwEfzWu  
+CPD4eZESe5ZDn2Wr7XhMCycmoJtKE/DN8OB0RaSv9Hqt5z/  
tTHzp969B7W9GrN4s8EUcm6ra1uNo1T4xNzQgkAyDRHIB8mTVVwzp2Jt5CptdZVcNt  
A9hDcXottvio7wGoZ3056/U+bcBHNZhvwUfzbFBfdtSUQHICgGdwO3uN3TSP  
+KXwGATgXq7QHjo0d9FgHSol6DOdclR0iRX86oQ07eie7FN/  
pEvTX26APFV52iplf80JJMPUT8STlcZ70vS6lvJxOB0i/YT  
+t9n2se3Tf9UJtNpPqRc9SembhOhegO4FbK9ha/o+j8UI9L8/  
YcKE9mr081SyKxcEkpGrVq1qHRgYeJzd+yoZ7eM8QdDQSD+B7udK7o/2vyJ9UH/608/  
a4v9t6a83+nEJ7ZfJyE9G5iLkp1PDTGdfX0KdniVh0F  
+4PKke5jVr1hwpTKzVfm4IFAogAVgCs56AeG0XxfrrdQtRNAq  
+IsuBURdsckcgOUG7aBok0iOp03wiFyBynucdyHMn7Z29ebMzlwQSSNRampS2kt3HW  
NuUNgaX4dmdjKivpQbKZY+7j06sTOlqwOhh/gfzeNXGWMeaSwAzcf6Er  
+vkuzDIK3nke25roNGBifqMuqmZLht9rpGOlctHrF217Nux4Fk3Blqdgkg3Q6KHWF0nqc  
WqcrWFNO+xroY4VR3LSgtC3REodpintfk0tEWk6+K0etxcmjdolK/  
29a56tTGoWwLFQFEjXQmJVrJ2kHZ2nJ7z7Q8QZwvrWmqc1J9YqaWvdU+gGLyurq4J  
+/fvv43jZZBJk7JSj/THuj1t9TVUvRS4QZ+VS/  
tlME82pVbTMAQqRIJaaQokWkjaAtb57F9QeL5a  
+xBGr2nvZO1jfu1jb5s21BLQxJodlQglAZs5xNEjVVdynYaW69dGOg8hs69bD9m20e7Z  
ieEqelA52gcsjgeEwQaDZxe1jt48ODvSR8ex4JcGtM6n2ONmk

```
+CANpqzGt4FJ3jQY41sq+txtAGSfsGkgyPoXHcT5/  
Nly7/2yJvWAICAYGAQEAglBAQCAgEBAICAYGAQEAglBAQCAgEBAICAYGAQEAglB  
AQCAgEBAICAYEcIvB/Q079+h6myXwAAAAASUVORK5CYII=) 2x);  
}
```

```
@media (prefers-color-scheme: dark) {  
  .lookalike-url .icon {  
    filter: invert(1);  
  }  
}
```

```
</style>
```

```
<style>/* Copyright 2017 The Chromium Authors. All rights reserved.  
* Use of this source code is governed by a BSD-style license that can be  
* found in the LICENSE file. */
```

```
body.safe-browsing {  
  --google-red-900: rgb(165, 14, 14);  
  --google-red-700: rgb(197, 34, 31);  
  --google-red-600: rgb(217, 48, 37);  
  --google-red-500: rgb(234, 67, 53);  
  --google-red-100: rgb(250, 210, 207);  
  --google-red-50: rgb(252, 232, 230);  
  background-color: var(--google-red-600);  
  color: white;  
}
```

```
.safe-browsing :-webkit-any(  
  a, #details, #details-button, #proceed-button, h1, h2, p, .small-link) {  
  color: white;  
}
```

```
.safe-browsing button {  
  background: white;  
  border-color: white;  
  color: var(--google-red-700);  
}
```

```
.safe-browsing button:active {  
  box-shadow: 0 1px 2px 0 rgba(165, 14, 14, .3),  
    0 2px 6px 2px rgba(165, 14, 14, .15);  
}
```

```
.safe-browsing button:hover {  
  background: var(--google-red-50);  
}
```



```
.safe-browsing .secondary-button {
  background-color: var(--google-red-600);
  border-color: var(--google-red-100);
  color: white;
}
```

```
.safe-browsing .secondary-button:active {
  box-shadow: 0 2px 3px rgba(0, 0, 0, .5);
}
```

```
.safe-browsing .secondary-button:hover {
  background-color: var(--google-red-500);
  border-color: var(--google-red-50);
}
```

```
.safe-browsing .error-code {
  display: none;
}
```

```
.safe-browsing .icon {
  background-image: -webkit-image-set(
    url(
+VBMVEUAAAD//////////9/f3///9/f36+vr8/Pz7+/v7+/v7+/v7+/v7///
r6+vn5+fk5OT5+fn19fX4+Pju7u7v7+/o6Ojx8fH09PTy8vLz8/Pj4+P39/fm5ubp6eni4uL8/
Pzw8PDt7e329vbs7Ozg4ODh4eHe3t7l5eX6+vrD3d3q6urf39/c3NzbRDf7+/
vb29vW1tbZ2dk+D9arAAAALXRSTIMA/
elxBfACHj3YwhYJDyfOtr2WHObgEqpjellgY6c6mZLV3L2qKjOZemN
+rUv7NpAAACcUIEQVR4Xu2U53baQBQGAXiMaQZT3Wt6rnql9uKW9v4Pk5XEspG4
WFmJ/
GMeYM635cy7/8yePXv23N7uxnNwdHSwE1ENoLYLTz4NkM7vQHqFhKv4nlQCCIIUbF
EOXHJxPVVYUY3nKSSpKFmIjBqANRdxPlcnTHRyGEPUAga0onuy4CMb1ZM5Bh/
HmYiiGwhwE81TviYA18VlokY4DKKp1SBDSqICklyJT5PU1AafL3FVC4q1uDLdS4+4r
CXd06bKXO21fKdxcNKFzVzUFQ9KoDJRehr0z0tBgBpcrfVyZavAxM7uo2ABHNO8s2N
TW4+8pEs4HUMxS+6rYAEZFBI+Y8FQ3C5holmGMNVVvLp4X1FRZ2n0BU035X
+u7jmgomWv/zzWR23IB7ich/
b1DBeRQVNNty1FHYLDWVh1TwEXic4gs6105UfPdMrd1/
HCffrnqTayrYkqP3qmSom7r9q84xyMDLIUx7Mylfn7OvIGmbYyUYmGbmry99UWnYOZ7
sFc5OHb1b2HLbS9QV1vED3dPX9fQXFuWiEeF9kzpfMcfaUZkbyn9+bQTfXwvm4W0iA
3RNdQUyIV1lekkDOhSx3sdGh17+At0ctA0plDpr/gLqSvSI
+kXt8mAj9DVI20r3hGBM0iOwKmBtZXHOPX79fFXCJ/
Ure79Gxr40Owuu9hK8LPp1mnR7JmWuv7ZsM+oH3FGZObFoWx3iY/
UI1J2LVn0b7i6B2RZM1WumwL833MoH3FMQ1hPHIGsadnkv94X3EsctPE4xK8p89FvK
84ExIRuoUa6Pt98fU1DJUMosj+3/S15OtrGOsB9LnYqm+e5w
+es4JAhYQDdgAAAABJRu5ErkJggg==) 1x,
```

url(  
BMVEUAAAD//////////6+vr7+/v7+/v7+/v9/f36+vr7+/v7+/  
v//////////9/  
f3////////8/Pz////5+fn////////6+vr////////8/Pz7+/v7+/v////k5OTh4eHf39/  
e3t7g4OD6+vrBdF5+f7v7+/x8fH09PTY2Njc3Nz39/fq6ur4+Pjz8/Pn5+f7+/  
vZ2dnW1tbt7e3U1NTV1dXl5eXr6+v19fXy8vLs7Oz29vba2trb29vo6Ojm5ubu7u7d3d38/  
Pzp6enj4+Pi4uLw8PD///  
+BQ30nAAAAQHRSTIMAHhYFDzEJAifiwvb98M76PdjqtoGpZlI0c46qj1icTJ1KS42A6W  
Wb2WdadYK1cvCb/cJXV+KA9qioPc5l6Y36uRjySgAABXVJREFUEf7s1kuKwIAUBNDS  
+JtqggkkmODIH4gPRlgD11P730BDI6HV260Z1Wu4ZxUH/  
4tzzjnnnHPOOeecq2tEJdlsEsRkT  
+4RkeGRPA4Rj4YkG0RjfCLJ0xixCPwWEIkd73alwuTMu/  
MEMTiwc4AeBhd2LgPolfyhhFzLBy3Eplc  
+uE6hteWTLaRGcz5ZjKBu8UUFodmNL24z6KQ0pJCpaaohksxpmifQKGhjlXurSbbZhr  
9qRG81yTYbaJtNuefcsFbTbLNznwjE7zVINtsybdK4VttreytHdlml/  
zIUvTWjmyzFT9USd7akW02pUm22TV7WOveKttsWV4K1Vtlm12xp5XmrbLNBvYWhG  
+15YK3qjab0STb7Fcv5tmcOBKE4Q2Xc7KrTJW3yhs/  
2RsuZwvBjsQShPd0JxCyJVmA5v//getpgYextLSYKdHf  
+PbUO93No36Ht77eqHCvNnvASaBLp4rooClvpYGurpx92Sx6Kw3Uv873ZbOHvA7Qtd  
12m7ZZ6a00UN  
+225a7F5tt8XpA7Y61rJi1ViPeSgPZ7XZnuVh4zdsseisNhAEtZgOvaZt9yGsCdSwLeJKk  
16DNSm  
+lgSCghQAax9NGbfal1wQSD7YlgnE8H0a3iY4a8FYSyMKAXgJoMqBs1txbaSDs6AR5  
Zv6Asllzb6WBIKAkiefzCQA5A8Jmjb2VBtoMKHXcUSM2e8x3ADoPAAh4Jr7v5G7IVKLjh  
r21DHS  
+DshP0xyAel3CZk29lQa6ebAUeLxe74KwWTNvpYGSFVDqOK4LPFF0QdisibfSQBsdj  
QFNo9Eol2zWxFtpINiJw+Fs5hcBTaeDwaibETar76000GZAnggleLqSyNxmd/  
luQOM4LngceDBP8ACQSnTY+L111geRtqxlMfLDAsiFkV8H1GWSyNBmW7xO  
+eiJyo7O87B4MCAa1lxazd9bUwglgFYd7a8Dwo4GlqYQgc02f291kEcd  
+YJHVmZusye8djkAlKxHHnjEiEUIVOTDFKKTPdxbXdnRjghl7GjgUSqTNmvurTRRkMQ  
AtBp53lkjJuNRiY72cW8NiwdLHdyJgmcdkGKTGTGz2Gd  
+xPOhoGVAUjZgCo8zas33cW3lviFqWhyF2ND7YbaZM2qy5t9JEs8ITPexoJIFksUzTZo  
+5Tk3FilWh7KAKpEzarLm30v/2ckczlYXdsGU6NvuY6wFdJjn+y4OXKR3EytP/  
WMNbNYD6Y9Ag8WCs/F4yo51t9oBrAoGJzJV/  
+WokJDow9NbaF7QJAKkNpMyb9KPfjL2VBrLtnLpQ0DIYNTKdrHZp1wfqA0iEjhMYsisl  
N9l9LSmtz7RBxLmGCSxW9k9pYye1LPZR9wACNQaglbtagaSfvTI3Fvp  
+xDwgDd6ZERAvm9mT7kB0HIZJln4Ekp7W+KRRKfG3kqfYyAg4HHy6ZZwpB  
+daHgrVfG///  
3z5u1l38aODsbFrcH1opJ7VPgRabNnfPeal8+13REdPV4FBA4CRkQyZWekt2rU8M3IV  
REQns9AG4XmgxLRjc0+v0d4q1ZNVjznNw8WgsTiviaRXmh5K/19jzsaL  
+biwVwhsbeXdlUfffa+nrfSX9P4rYjnRcdFBWGSQ+KUf3/  
ybp7vuX6lFp7woaN9eDBxiUElWCDH/  
wfCWzUrXwTrgLCj14rGuhTSxx8Q3qpZLnSQ4MIXHS2He8u  
+xvqJ8FbdCmEFTXxcQdFAJkPP2qfVNvsLNy1vjg9WBKQsQMkpfiW8Vbt6SkdLFkb50  
VeEt2rX1MGA4MHUIhkRfvTFh4S3alcEHaSczmr20R+Et+rXCdu63DJsux/

```
9+R7hrQZE0NFEHIVh/UV4q0F1B6zSEdIWF/ISsdn/  
AccLgKctjBheAAAAAEIFTkSuQmCC) 2x);  
}
```

```
@media (min-width: 240px) and (max-width: 420px) and  
  (min-height: 401px),  
  (min-width: 421px) and (min-height: 240px) and  
  (max-height: 560px) {  
  body.safe-browsing .nav-wrapper {  
    background: var(--google-red-600);  
    box-shadow: 0 -12px 24px var(--google-red-600);  
  }  
}
```

```
.safe-browsing-billing .icon {  
  background-image: -webkit-image-set(  
    url(  
VEUAAADcRTfcRDfdRET/gIDcRjr/  
ZmbjVTncRDfcRTfcRDfdRDzgSTncRDjeSDvcRTjbRDfbRDjeRzvcRjfbRjjcRTjcRTjcRTf  
dRTfcRDjdRTjcRTjbRDjbRTjbRTjbRTfcRjjdRDrcRjfbRTjZQzfcRDjZRDfZRzbWQzXXRD  
XXQzbXQzbWQjXYSdvWQjbbRDfOQDPSQTTUQjXCPDDNPzPJPjLGPthVQjXMPzP  
RQTTWQjXLPzPDPdHYQzbAOzDTQTXHPTLIPjK8Oi++Oy/  
FPTHEPthPQDTQQDTUQTxBPDDKPjK/OzC9Oi/////
```

PQDPRQDS3OS66OS7TQTTEPDHXQjbMPjMBhLaWAAAAL3RSTIMA4tgPAhYFCcL9  
8B4x9ie1+s49WICbqXNKZY3pjuqcgVdLZnL2qKg9zmXpjfontV8LANsAAAJrSURBVHhe  
7ZTndulwFAY3ARlgBAg9vW1v173ROylby/u/ys02Fx3MNaxs9h/zAHM+Sfa8+M/s2LFjx  
+3tdjwH+/  
sHWxHVAerb8KSyANnUFkRXwLiK78llgJHJxRalwSMd11OGOeV4nsM9FO0dxhJdw4Lr  
OJ6jYy46PoohqgEHatE9JViiFNWTPiEITpIRRXcQ4C6aJ3EJAS4TkUQXsMJFFE+  
+CCsU8xFEBSAoiHsaQNIQ7yuQCFe3DiHUhtKIlzdKoRSFe0r8sXDAkSoumkligYalOkIf  
eWi56EESFm8r1w0fFll4epWgBA9qOMpmirCfeWijtoa9WSx6taAELFBRI/  
viIS3BJRlBk9/VFTsLrifUXRuNfXLU0y/7m6p0CKxqN+v6lJU/  
k3eJxu7Os5LWKDHi1tYstKG1zON1X3DGiRMR80Mx3fdCbc1+bQe3o2SJrYXcV0fFMx  
L9xXiz0987BBtux65qaCeF8IHCR3FabBTQ3xvk4M1yN5B/  
Mw2+urew8hTP1BM38Qnu5evK8gMw+7lcfH9E3ZIEBfMSO//Kf35+Cm6ua  
+rhbSYDeEa9CUyW3qK1Hlj5DBz8dWd0bWCd6Ult/uMPER+BmbV/JHrVG/  
a9MsEyV5fsK50R3frmBFXtCtVXmt73H4PhQ4t9k9rkJ55tYXwZrO4rCEUfPHfUEcuaZC  
/umw97TfaVpslu2tCb2lRWnBlKFtf+huwrjaa6Pxx7RfgW7nubJPtKI/X0puQO4k/Pfe/  
ovtLY7KbxVwve0/sE3VeaLoslBkEDvt8Hoq/hKGwQYvoq5OMnoq/hLABgc/  
FVn33PX7pAfE5QHR6fAAAAAEIFTkSuQmCC) 1x,  
 url(  
BMVEUAAADcRDf/ZmbcRjrjVTn/  
gIDdRETdRDzZQzbXQzXWQzbXQjbWQzXZRDbbRDnWQjXWQzXYSdvbRTjcRTjbRT  
fcRjfcRTjcRTjdRjncRTfdRTndRTfdRDrbRTjcRDnbRDfbRDjbRjfcRjfbRTjcRTjdRTjbRjjcR

TjcRDjcRjncRTncRTndRDnbRTjcRDfZQzbcRTfgSTncRDfcRjjZQzjcRTfVRDbcRDjcRDj  
WQzXeRzvbRDjXRDXQzbXQzbbRDfeSDvWQjbVQjXIPjLOQDPXQjbCPDDNPzPUQ  
TXRQTS5OS7QQDTUQjW3OS7SQTTTPQDTFPDHJPjK2OC26OS7HPjHOPzPLPjLMPj  
PRQDTGPDHTQTTEPTHLpZPGPTG7Oi/HPTLKPjLTQTXYQza9Oi/  
MPzPFPTHDPDHBPDC/OzC+Oy+8Oi/AOzDWQjX////  
bRDd3undHAAAAQnRSTIMA2AUWCQIPHj39wvbO8DH64ifqqYFmtrVMc1IKS5x0nY6P  
WKqbjYDpZXWCZ1py8Jv9McJXV  
+KA9qioPc5l6Y36J7VmcHe8AAAFWUIEQVR4XuzWS4rCQBSG0euz56lSgiaEjHwgGh  
AhDnRF3/6HDY1la5WPjP4a3LOKY28555xzzjnnnHPOuSyzpPR7vb6lZAUrS8hgB7uBp  
aMEKC0Zhz3A/mCpaPjTWCK23GwtCcMjN8ehpWDN3doS8HPi7vRjejX/  
1CbX8qA1sdGZB  
+eRaW14sjGp8YQnk7EpVQQqE7peCFyupjMnYm4yGVGZ7q1EyTZbEEche2uUbLMIL  
5W6t4Zkm22lkm02561c89aQbLNTpPgq3hqSbbbmo1r41rhW8NaAaLMzvjITvDUg2WzF  
lyrBWwOCzc6Jkm12QQcL3Vtlmy3opFC9VbbZJR0tNW  
+Vbbahs0b41rhc8FbVZqdEyTb724t5/bYNA3G4e  
+80NYl0gGFkvaR779KKZUWuFKe7nllsT5X//2M5VMZiZB9DQj74xW8ffrwjP90Mb/  
07Vf5CbXYJg0BtO4toKS9vhYHGy1vDZg28FQY6tBZls8tYBehwNLtyt1nhrTDQaDQcW  
Aux2SJWAxpOBpWMWSvm4q0w0Gg4nFQqFTd/m72HIYBYQJV  
+w83bZu9jRaDJYEB4osjJ02aFt8JASUBRq  
+PlarMrWBGI8lQajVanXA5kopUcvBUEGrCAWhSoXs3PZtewKhA/  
MMbTbcpEa7l4KwwURZSHANnVnGz2CVYGmg6oZ1u1XGy2hNWBCA8BogE1m7ZI  
+ShNVMrdW2Wg/v  
+Amr2eRYCcGLBZU2+FgcSBESDfdZxdwGbNvBUGihKgnk1OjPAEwS5gsybeCgNNdT  
QLyAtqtRCwWQNvhYH4ndjtNnlAnlet1uIQsFI9b4WBpgNyaUCEJ45DwGa1vRUGanU6n  
McmB+ZSnlosES3nvm/tUpGm1tFPd5DDAYKFBjGpzRaxSjW5J0o8/MAQ4ZEyKua/b  
+0Np175bIMERDuaECFBZGqzBaxY9iAjIMbDK01U0OVZxcplE6BljLzFRixgQDwflCJaX  
cC+1ToKyOYHFvCOljPiNmvrTBRI  
+oQoGTk2Z1YQyleiWhlEftWnx8Yf8RcyiMCEkyhic2u4xOWSw9MBBQENTQFI83a  
+iL2rdgpJ1rms45mByYzhbDNwt6qTtTIQC7r6FT/CLRQ02ZLWKc8OmK  
+LzooCykhKpl4q7p+7B/d0SjNggRbqGOzm1gPqL3PX3niZakOQsenf1PDWzWAxr  
+JBtEDQxnnJTISNmvrufBK75t45bORBNGSobcqb9DqBCjdQOI5E370xthbYaDRiljRDx  
KQwJk9a+o2u431gYZERBo/kcBlfvJ/TrSt6K1b+kDUHMkra2V3j5zRlprNbmADILbQ65S/  
z2ggyY82zL0VXsdQnnLdhSOKQzWbLWADIMpDgOrd3q958QiigrG3wusYzmNbXmY4s  
h+tangrVJ2Dgy97X9v0CmlLzzlHcj3ZPTL  
+h6DN7mhYR5nxHI4mtKNbLCamaX9QDDKFO6C36hDttcdJQFGLeTWRIlupocGOj62c  
Bb9WqesLTFwfm000MQgqz9IDLW  
+Hve35HM9Fnqw9HetBkNsF6+Yaet8Jf0+xbka0XbYspSMlg  
+5D8/8psnqdYv3qso1vsS9Hy6SaGQ6AYHP9ngLdqllVpilB8RygRQjGEdOsc4K26RGzk  
6YTjxhbDDdzXcfwC8Fbd8glPnR4Y62gBAM/  
a1WybfYVNyyUBiZFPXYCAH70GvFW7nFRHH7Egyl8uAd6qXZ7NAqoilG6ZKuBH184D  
3qpdAQIIWp0p9dE7wFv1q8Y6+njLoPI  
+9P4C4K0GRKSjgTyywvoAeKtBxVWU6YhorovcvA14q0HtouwU0Fw/+jzN8w/cQ/  
zg6ug2/QAAAABJRU5ErkJggg==) 2x);  
}

.safe-browsing-enhanced-protection-message {

```
background-color: rgba(0, 0, 0, 0.2); /* black 20% opacity */
border: 1px solid rgba(0, 0, 0, 0.3); /* black 30% opacity */
}
```

```
.safe-browsing-enhanced-protection-message .icon {
  background-image: url(
ZHRoPSlyNCIgaGVpZ2h0PSlyNCIgZmlsbD0iOZGRil
+PHBhdGggZD0iTTkgMjFjMCAuNTUuNDUgMSAxIDFoNGMuNTUgMCAxLS40NSAxL
TF2LTFlOXYxem0zLTE5QzguMTQgMiA1IDUuMTQgNSA5YzAgMi4zOCxLjE5IDQuN
DcgMyA1Ljc0VjE3YzAgLjU1LjQ1IDEgMSAxLjE5IDAgMS0uNDUgMS0xd0yLjI2Yz
EuODEtMS4yNyAzLTMuMzYgMy01Ljc0IDAtMy44Ni0zLjE0LTctNy03em0yLjg1IDExLjFs
LS44NS42VjE2aC00di0yLjNsLS44NS0uNkM3LjggMTluMTYgNyAxMC42MyA3IDljMCO
yLjc2IDluMjQtNSA1LTVzNSAyLjI0IDUgNWwIDEuNjMtLjggMy4xNi0yLjE1IDQuMXoiL
z48L3N2Zz4K);
}
```

</style>

<style>/\* Copyright 2017 The Chromium Authors. All rights reserved.  
\* Use of this source code is governed by a BSD-style license that can be  
\* found in the LICENSE file. \*/

```
.ssl .icon {
  background-image: -webkit-image-set(
    url(
VEUAAADcRTfcRDfdRET/gIDcRjr/
ZmbjVTncRDfcRTfcRDfdRDzgSTncRDjeSDvcRTjbRDfbRDjeRzvcRjfbRjjcRTjcRTjcRTf
dRTfcRDjdRTjcRTjbRDjbRTjbRTjbRTfcRjrdRDrcRjfbRTjZQzfcRDjZRDfZRzbWQzXXRD
XXQzbXQzbWQjXYSDvWQjbbRDfOQDPSQTTUQjXCPDDNPzPJPjLGPthVQjXMPzP
RQTTWQjXLPzPDPDHYQzbAOzDTQTXHPTLIPjK8Oi++Oy/
FPTHEPTHQPDTQQDTUQTXBPDDKPjK/OzC9Oi/////
PQDPRQDS3OS66OS7TQTTEPDHXQjbMPjMBhLaWAAAAL3RSTIMA4tgPAhYFCcL9
8B4x9ie1+s49WICbqXNKZY3puqcgVdLZnL2qKg9zmXpjfontV8LANsAAAJrSURBVHhe
7ZTndulwFAY3ARlgBAg9vW1v173ROylby/u/yso2Fx3MNaxs9h/zAHM+Sfa8+M/s2LFjx
+3tdjwH+/
sHWxHVAerb8KSyANnUFkRXwLiK78llgJHJxRalwSMd11OGOeV4nsM9FO0dxhJdw4Lr
OJ6jYy46PoohqgEHatE9JViiFNWTPiEITpIRRXcQ4C6aJ3EJAS4TkUQXsMJFFE+
+CCsU8xFEBSAoiHsaQNIQ7yuQCFe3DiHUhtKIlzdKoRSFe0r8sXDAkSoumkligYalOkIf
eWi56EESFm8r1w0fFI4epWgBA9qOMpmirCfeWijtoa9WSx6taAELFBRI/
viIS3BJRlBk9/VFTsLrifUXRuNfXLU0y/7m6p0CKxqN+v6IJU/
k3eJxu7Os5LWKDHi1tYstKG1zON1X3DGiRMR80Mx3fdCbc1+bQe3o2SJrYXcV0fFMx
L9xXiz0987BBtux65qaCeF8IHCR3FabBTQ3xvk4M1yN5B/
Mw2+urew8hTP1BM38Qnu5evK8gMw+7lcfH9E3ZIEBfMSO//Kf35+Cm6ua
+rhbSYDeEa9CUyW3qK1Hlj5DBz8dWd0bWCd6Ult/uMPEr+BmbV/JHrVG/
a9MsEyV5fsK50R3frmBFXtCtVXmt73H4PhQ4t9k9rkJ55tYXwZrO4rCEUfPHfUEcuaZC
/umw97TfaVpslu2tCb2lRWnBlKFtf+huwrjaa6Pxx7RfgW7nubJPtKI/X0puQO4k/Pfe/
```

ovtLY7KbxVwve0/sE3VeaLoslbkEDvt8Hoq/hKGwQYvoq5OMnoq/hLAbgc/  
FVn33PX7pAfE5QHR6fAAAAAEIFTkSuQmCC) 1x,  
url(  
BMVEUAAADcRDf/ZmbcRjrjVTn/  
gIDdRETdRDzZQzbXQzXWQzbXQjbWQzXZRDbbRDnWQjXWQzXYSDvbRTjcRTjbRT  
fcRjfcRTjcRTjdRjncRTfdRTndRTfdRDrbRTjcRDnbRDfbRDjbRjfcRjfbRTjcRTjdRTjbRjjcR  
TjcRDjcRjncRTncRTndRDnbRTjcRDfZQzbcRTfgSTncRDfcRjjZQzjcRTfVRDbcRDjcRDj  
WQzXeRzvbRDjXRDXXQzbXQzbbRDfeSDvWQjbVQjXIPjLOQDPXQjbCPDDNPzPUQ  
TXRQTS5OS7QQDTUQjW3OS7SQTTTPQDTFDPDHJPjK2OC26OS7HPjHOPzPLPjLMPj  
PRQDTGPDHTQTTEPTHLpZPGPTG7Oi/HPTLKPjLTQTXYQza9Oi/  
MPzPFPTHDPDHPDC/OzC+Oy+8Oi/AOzDWQjX////  
bRDd3undHAAAAQnRSTIMA2AUWCQIPHj39wvbO8DH64ifqqYFmtrVMc1IKS5x0nY6P  
WKqbjYDpZXWCZ1py8Jv9McJXV  
+KA9qioPc5l6Y36J7VmcHe8AAAFWUIEQVR4XuzWS4rCQBSG0euz56lSgiaEjHwgGh  
AhDnRF3/6HDY1la5WPjP4a3LOKY28555xzzjnnnHPOuSyzpPR7vb6lZAUrS8hgB7uBp  
aMEKC0Zhz3A/mCpaPjTWCK23GwtCcMjN8ehpWDN3doS8HPi7vRrejX/  
1CbX8qA1sdGZB  
+eRaW14sjGp8YQnk7EpVQQqE7peCFyupjMnYm4yGVGZ7q1EyTZbEEche2uUbLMIL  
5W6t4Zkm22lkm02561c89aQbLNTPPgg3hqSbbbm01r41rhW8NaAaLMzvjITvDUg2WzF  
lyrBWwOCzc6Jkm12QQcL3Vtlmy3opFC9VbbZJR0tNW  
+Vbbahs0b41rhc8FbVZqdEyTb724t5/bYNA3G4e  
+80NYl0gGFkvaR779KKZUWuFKe7nllsT5X//2M5VMZiZB9DQj74xW8ffrwjP90Mb/  
07Vf5CbXYJg0BtO4toKS9vhYHGy1vDZg28FQY6tBZls8tYBehwNLtyt1nhrTDQaDQcW  
Aux2SJWAxpOBpWMWSvm4q0w0Gg4nFQqFTd/m72HIYBYQJV  
+w83bZu9jRaDJYEB4osjJ02aFt8JASUBRq  
+PlarMrWBGl8lQajVanXA5kopUcvBUEGrCAWhSoXs3PZtewKhA/  
MMbTbcpEa7l4KwwURZSHANnVnGz2CVYGmg6oZ1u1XGy2hNWBCA8BogE1m7ZI  
+ShNVMrdW2Wg/v  
+Amr2eRYCcGLBZU2+FgcSBESDfdZxdwGbNvBUGihKgnk1OjPAEwS5gsybeCgNNdT  
QLyAtqtRCwWQNvhYH4ndjtNnlAnlet1uIQsFI9b4WBpgNyaUCEJ45DwGa1vRUGanU6n  
McmB+ZSnlosES3nvm/tUpGm1tFPd5DDAyKFBjGpzRaxSjW5J0o8/MAQ4ZEyKua/b  
+0Np175bIMERDuaECFBZGqzBaxY9iAjIMbDK01U0OVZxcplE6BljLzFRixgQDwflCJaX  
cC+1ToKyOYHFvCOLjPiNmvrTBRI  
+oQoGTk2Z1YQyleiWhIEftWnx8Yf8RcyiMCEkyhic2u4xOWSw9MBBQENTQFI83a  
+iL2rdgpJ1rms45mByYzhbDNwt6qTtTIQC7r6FT/CLRQ02ZLWKc8OmK  
+LzooCykhKpl4q7p+7B/d0SjNggRbqGOzm1gPqL3PX3niZakOQsenf1PDWzWAxr  
+JBtEDQxnnJTISNmvrufBK75t45bORBNGSobcqb9DqBCjdQOI5E370xthbYaDRiljRDx  
KQwJk9a+o2u431gYZERBo/kcBlfvJ/TrSt6K1b+kDUHMkra2V3j5zRlprNbmADILbQ65S/  
z2ggyY82zL0VXsdQnnLdhSOKQzWbLWADIMpDgOrd3q958QiigrG3wusYzmNbXmY4s  
h+tangrVJ2Dgy97X9v0CmlLzLIHcj3ZPTL  
+h6DN7mhYR5nxHI4mtKNbLCaMaX9QDDKFO6C36hDttcdJQFGLeTWRIupocGOj62c  
Bb9WqesLTFwfm000MQgqz9IDLW  
+Hve35HM9Fnqw9HetBkNsF6+Yaet8Jf0+xbka0XbYspSMlg  
+5D8/8psnqdYv3qso1vsS9Hy6SaGQ6AYHP9ngLdqlVpilB8RygRQjGEdOsc4K26RGzk  
6YTjxhbDDdzXcfwC8Fbd8glPnR4Y62gBAM/

a1WybfYVNyyUBiZFPXYCAH70GvFW7nFRHH7Egyl8uAd6qXZ7NAqoilG6ZKuBH184D  
3qpdAQIIWp0p9dE7wFv1q8Y6+njLoPl  
+9P4C4K0GRKSjgTyywvoAeKtBxVWU6YhorovcvA14q0HtouwU0Fw/+jzN8w/cQ/  
zg6ug2/QAAAABJRu5ErkJggg==) 2x);  
}

.ssl .legacy-tls.icon {  
background-image: -webkit-image-set(  
url(  
SR0IArs4c6QAAB21JREFUeAHtXF1IHFcU9ie2bovECqWxeWyLjRH60BYpKZHYpoFC  
U60/  
xKCt5ME3QaSpT6WUPEICEXyTUplojfgTUwshNpBgqZVQ86hGktdgSsFGQqr1t9+nd2  
WZPefO7LjrjYzcJmZc8495zvf3LI3Zu  
+dzcoKt5CBklGQgZCBklFMZSB7r4G3tLS8sLCw8D7ivo1Ssrn5WYL9AZSC7OzsAuyzl  
HuCHcsjyOawZ7lbVFT0W09Pzz843rNtTwhqaGh4ZXV1tQFZfYZSDgKe85MhyFpBvTso  
V/Py8q5g  
+9OPn0TqpJSgurq6CpBxFuUEQO1LBJgH2zUQdgPlwuDg4LgHe18mKSGovr7+2Pr6+j  
kgOulLVEkVJnJzc78eGBi4nXhVe42kEtTY2Fi8vLz8HVrMKXvY1GjRmvrz8/Pb+/  
r65pMVIWkEodV8vLGx8SPI2Z8scH78gKTFnJyc02hN1/3Ud9ZJCkG1tbVfwnEnyMlxBp  
DOkcQybG9ifwv6OezvRyKRv5eWljhyZeG4AMcvweYNnHKkq4TNcezzqXfbYLSBm46h  
oaELbrZu+l0R1Nra+vz8/HwPgH/uFgj6xwA  
+inlNt8Evvb29Tz3U2TFpamp6EbfvR4hVhXISisldpXKAWJeLi4tburu7/1VMXMW  
+Ccll9TKA/oTyni0KQC5B34V9J0abRZutVx1i70fcDti3YR  
+x1UPcSZRPEfsvm52m80WQaTm3beQA1Dr0F9EffANwDzUAu5GDqIPo975FrGbEytV  
8QT  
+JlnTMT0vyRRD6nEsAZLutOlPUDw8P86Eu5VtNTU05goygFGvBQNJI9ElfaHpNrrKuV  
WCHDHLOanoAmUKr  
+QBgzJWbZMtnZ2cflpWV9cPvUZRXFf9vHT58+OnMzMzvil4UJ0QQh3KQ8wM8iS0P5P  
SjVOGWWWhCjpVCIXJ+AgD6EeA2ITAoFbB  
+CyKnp6en7kl6SiYIKhuYhcBYEic85JAethu9bad/Qyq8Ap/  
iwCpyLGEUPeX2Y9PTcwozNE7JGzhQCn0k7MwYAsaBMSXh4gZmLpJNknIqQebe6JT  
mAbB59zru7GanQyW5KvtHJe8ln1TUj3B/  
QiR033t0qvby7eWpB5sUzDgeu0jqE1bshJ85pkgQGU7XBGODVy8lp6EoQrkQFKolv5Wi  
uF/dqKHcC93JObMSO2B4xuSnqbbErQQggDum4Mkt8CLR6D4CSGIIVgqLIFmtrJYi/  
BMIJf+yStq4g3lpOoAZjl1POc+bGHCvdVGYlaGVI5TQMpV8C+eLZGXUS9L3B+lJAuc/  
8FCyotkVS8jvGcFwNInfOoweQj  
+LKJOXfKz53M1pFMdn2xlpno1Hklr0e8XdysYXRp9qCOPsAPd9x4jYQdC1OGHCBBX  
O5yVXMQCWIUzNgPG72AYGW  
+XuO6C3AQmImdidE5mimoZyqrXOVIGg5bxW3weHNRH/  
sinOSBgExE7sSWsyVtjaCSiRnuAraE7VkhiiZBbuYK8GrBIFtsRKC3AtU1gmA0bBrudK1  
bRQ7oMR  
+oMh9i1PxLqaA0bBrueotCAG25smdgTj74JRlyrkFu5gr81JvMTRHsVJ0aiZTSInFqWHX  
crUSFOv4WT5WWxA6rq1JPCc5nNRzyjLIXMOu5cq8VIKGEwnijGemEOLEacEu5sr6Nol  
eOQPwHGxzOjgJNwt2MVcmqRKEjmtOYUF8PIJsgyYWsVty1QICZiJBuAqVQcvaKx4Ldj  
FX+IVbEHR3pcBg

+zgXEki6IMulmdgVjGKutFUJ4oJJOFxxOsRVyOcqC6c86OdmZUjc8hnmyFw1/  
CpBZjWpOLcOkqo0h0GVWzDfsa2cVQkyiV6VEkaw5gRECCrJft0y4iVmBUcYo5RWyt  
BXGoLw7Woccy  
+EAE7Ys4DfWiwFgog10yOgmpbZCWI65Bxj44ptdtwZQ4qusCIDcY2CRByu  
+G21tpKEJ3CyXnJOa5KhlUxJF2QZMRlRBlm5Oa6htGVIMwljMP5hBKg2SxektRplxEbS  
GhWgEyY3BT1ttiVIJpxkbbkBVeG64tGgnirGUwjBmMcfC0np6Hn1RMua264/  
OUorog4xesMmupzkBMBMb  
+ivCPFAIbPa5k8tSAGwbRJOxyLk4UEgsKVZ4HYiMVCDhdQtXsF6rkF0aFZTf8zgovE8s  
qgnElXSzlth+SckggAtg0sZvgkkVX4Ca1R5Nq+0tJSfq  
+lvWpwbeAJrBW8zjWDEshUydjngJgxFA0bR+SvcPEuJYIhoRYUdYz  
+6JIZBizeKIEitD2X9+NqTGp6yluhn8Aw+70ZTSym/lX0zRiMxZiaJ2IIZk1vk/  
tqQQXlC0GnCDZmqQs/ZnFjyOjRJ/n+HArNn1PZDzipF5234uyD  
+YH9dXS6b6Jk5udQsfz9Xz  
+o89VJxxITPeazBR7ADqFF8JuJtGyMTQyJPOe4AfXdSdscm4Xn52AjLh  
+21fWpy4yPep3JYaSrQP+Rys/  
Cx9BqzuPhb9wZO1nnKWIBTnDhHws4GbGcZ9pfU1hSCVUhAyEDIQMhAyEDAWfgP5  
qNU5RLQmxEAAAAAEIFTkSuQmCC) 1x,  
url(  
SR0IArs4c6QAAEp1JREFUeAHtnVuMFkUWx2dgRIBhvUxQSZTsw25wAUPiNQTRgFkv  
8YIbZhBcB8hK2NVkXnxRY0xMDFffJkHsyxskBFRGIJ4iWjioLJqdL3EENFZ35AELxnR  
HZFFBtjff+gePsbv0qe6+vv6+6Y66XR39aIT5/  
zPv6urq6q7m5rCEhAICAQEAglBgYBAQCAgEBAICAQEAglBgYBAQCAgEBAICAQEA  
glBgYBAQCAgEBBoTASaG9Ot8l6tWLFi4sGDB3+P1HStx44d0/  
a85ubmyWwnHz9+fHgbHTdxPEj6IMfD2+j423HjxvWTPryeeeaZX65fv/5/HI  
+pZUwQ6l477vjD0NDQAgiwgOBfynYa23E+l43OY+jcy/Zjtn0tLS19zz///Oc  
+y8ijroYkUEdHxxSCuBDAF7DOZ/  
+CWoAPmb6m3J2sfexv37Jly3e1sCPLMhuGQF1dXRP2799/  
G2TpBLCbWFuyBM5B9xB5XoVIPVOnTn2xu7v7sIOO3GWpewJR21xJG  
+ZukF3MenbuEC5u0A8kb6YNtY5a6YPilvWRWrcEWrx48Xyl8xA1znX1AXVxK6mR3oBl  
qzdV3qxbXd0tdUcgapybIY2IM6fu0C5jMER6j3U1NdlrZcRyd6puCARx5kCabtbLcoeiR4  
Mg0UesXRDpPY9qM1OVewltW7asjT6bJ0DgLGy5t9dTpl6j55/0Ld2/  
YcOGAU86M1GT24BQ0zS3t7evxOvHWNsy8T7/  
SkWeB3t7e9dSK4lUuVtySSBuV9NoID8LWnNzh1htDHqHhvd3Nb21qb40qV67Y0tXUz  
yMzxd3Urt8wk5AnlOwjZXmAibk0n52MtNDbRq1arWgYGBx4HlvmpAwy3hJ8rpJzD98Zg  
W+1+RPjh+/PjB0047bfDQoUMa+2o6/FTTJ//yyy+Tjx49OjxOhsxFJA+PobE/  
PJ5G3kmSrcLyZFtb2wNr1qw5UoWyKhaRCwltWbLklIsaqthCEqypa7CggwqD/  
bbZ9bPsuueSSTx955JFjjupOyYaecbt3756Nbo21acztGraZEQR97zPW1vHcc899dYohN  
TioOYFo78ygvfMavl+Ygf8aQe  
+lhumZMWPGlgKt4YTMF8pp2bNnzzz86oRI7RSo0X3fyz78uoF20R7fii36akqggG/  
nZUA+12J0JVII8zrr08htA+BDleSzPM+t+YwDBw7cjo/LWa/3WRY  
+fs96Sy37jGpGIMhzM1foZgA9wweoAKnb0VbaL6uZRvGpD52+dTctZDbtqlfQuwgy  
+XqA+ZmaaDEkqkkPdk0IRP/  
OnwFwPUCmHjGPiPNMa2vrY5s2bfrCd9Cz0Ld06dKLjxw58iC67/  
JEpCFItBwSqeujqkvVCRTVPC/gpQ/  
yfEgA7tm6deuHVUXNU2GLFi26nAvgKXy43INKkej2atdEvqrRRP6rzRPdtIKRB9APANa



9s2bNuqpeySPAZLt8kC/yKRGlpYVahK0wLi3i/  
0zVaiAcm8GVtos1VYMZoHfQL7O8p6fnW/  
9w1E5jZ2fnefQ7PQ0+N6axAnzUsJ5HTVSVp7OqEEj9PNzz3wWYNI/  
qqqlfZt7MEwCUy3GhNIFXXsjTTG/z/  
dQkj3KYppbeN3HixDkbN27cl9amSvkzv4Wph1mdhBiShjzq85jPVfV4o5JHgZJv8IG  
+cpgm+BcePny4V9hLb5ZL5gTS8ARXVpoe5k8B9AqA/  
VeWQORJt3yVz9jk3B0hzKOhoUxdy/QWpsE/+j1edPWAK/lt1oUA  
+qOrjnrOR7vxLliwnfVaVz/oF7uN2/5Lrvkr5cusBsL5adzL11cyoNR5iLNt0qRJN45V8ggX  
+S4MhEUpnCqIKwaKRSU51/  
OZEIgrphnDn2Xr9MQlwFg7xuKbnqMDKQyEhSuJFIMoFpncbTihUDST0Gk  
+D0C9xVWnyVNHR4M5Vo  
+FhTARNo4Yzl1i4pi9DbvrlzmMPdTpMs0VDWYrx3Lt63SoWpqUpul2kQkml1OrsS5Ae  
ZYT/c9x9p7DRRNgHchjx7Vx3Sbp0TgR5J1YQkjElwe8eOXE0b0+djxWgNxxhWio4h0Ms  
+pVJ6H6eWr2qM64IKlzkmlq48+4jWsA5yvBuedHLQYIR4H57ng7O2Vla81EA22bhwY  
A4tTD9eSPMYg1FxcWAKzB0Oaoxg5ZC2exRuBuCr0xuhlxYspnUrDcleGJ0pLhDPFeli  
GdHYUO1cuTTFsrMrJWM55lxCGaaKUaYE8BzQwytZ0+zAV0qDCwizCzjyK7xKrUjB6l  
RA9zvoGj3kaASA81Gij6qWAZiJd2Aldq27FSjGz5ism74VANOjMTuD4hzNnzvx7MaNC  
WnlEhKGwTJ7jhKRLzlqVkJpA3E  
+vhNGmT6zgsD4Hd4+v12qKOTZW0oShsBSmFp8VM8XOkqeYbGoCYcjKYoorpD1Tzz  
MJk/hW9dMRIs9YC3aM3SnFpCKQPiuHER2naKxwoCtFE  
+AriXTRgSEqUMt1KEYGos6RTwVgfRNQRZPYu3tV7enjgFqZwfRJhuNZp5dhRDY7aT  
4qklhJplJ1UI29N7W8kkg5QVARdsuYPoo6TOizOBalDpU7qmCeBUsa/n9aU/  
ZwRzlFHYCmOjSTcplsY8l+LWsZSRjJBnlQem/  
Dj39liCnO3UcmzLJxTCmNhYXqFuiWK51sUO5xqlwhYYCxxE3nlmnbGssSwujlW1ZbH  
GckR3GgKZejK5MnoZBKzphw5GvG7gHWEsrI0ummJZqNuJQNwz9ZKg6fcBjB73FBYc  
9rNDwlq1Yqn/ibhY5EQgusFNjOWK  
+Enf53ExMOSxlyCshbkp35GY5GPZZ0lhHGmwmD429X6uFPs2FjeCmthbsHAGtNYtx  
OBMO7SWEGSLcb1JZELMv4QsGJujWlsqZIA  
+lkbpneM8K4QKAY8SprrZgrpoqt1TwzgfSnP4xLnA/DftIHLA2GBfl0CAhzYZ9Ui2la/  
cUxaZZhucREKNCqz9palv4wbcMClx/  
ZCHO9XmVZrLFtypxAMNvqhMXhIFsGAQfssycQj/CmQuiTCAQqE  
+QsT1mxt8ZWtpvGspSB+  
+r5MFu7SZe6IFA9vReWfHjkTNgrtgbdw6lutzDTR7Mh21dWo4K8HwQcsDfVla6EMj0  
CX9YbR3Y84Ne0KK7hRV7U2ydCASrTSxlkpPViRB6TwhYsbfG1olAZDIRSH  
+98YRHUGNEwAF7U2xljvkWRrVoKiT  
+ZZLR9yDuAQEr9tbYykQzgTz4FVQ0EAJmAnGfNN2S9LO2BsKrrlyxYm  
+NrcAwE4g8JgLPt391hXoDGeuAvSm2gspMIOujoX4T2UAxqStXrNhY  
+tEIDKZWOrYAfhXUcqsQ7Ym2LrSqDEUwRUAKzWD2rDUgMERnhXpQ1Eld8YsTAN  
vhp1B/HyCFixN/  
8BydwGqsYlb3IMwtmkCFhH162xIR1mApHHOsJrvQqS4hPkKiDALcyKvSm2Kj5zAIHGd  
GbHuZRTAZ5wuhwCEeb5lxBfO/  
8SZh8rZ3zhOdpMk3bv3j27MC3sZ4+AMBf2SutSTBXbpPKxnLIm0M8/  
MGxvrCDJFuMWJJELMv4QsGKumLr83MZMILmIcR9bXmw4QCALYB5krZhbyxb6EQ  
gjDO954Vx13BPNk  
+fjY0MWxsCwlqYW3JZYxrrdiJQS0uLiUAYN2nPnj3z4kLDNIsEhLUwt5RijWms24IAfAnrc  
xj+dawkyZY+iVSfUktSRpA5gYAVa8VSMXXBz4IAUUh6W0zihSuinc/

CnJ44QxB0QkAYC2tjZIMsC3WnlZDpNkahGpX/  
U2HhYT8TBISxdQaENZYjhjsTiGpvO1qGRjQl2OHKWJ5ALlikQACMVxizD0WxNGY7le  
5MID6l9h0qXrWUinPX8yWs0KloAc0gK2zB+I+GLBJ9NYqIMdsJcWcCKTvMNX+2jklO5h  
+zOHk2BjO5YOsSw0JoUxFo6tSpL6Lsh0KFCfYXLV269OIEckHEgECE6SJDfOn  
+EMXQmO2keCoCdXd3H0bV5pPqKu9RxY47cuTlg5Ulg4QFAWEqbC15kN0cxdCY7aS  
4tcCTOaM95pCs+1Vi5YS7+JjB5ZXFGkQSBCIs70oiWyjjGLtCFU7TOU5RQAPsA  
+6jb5ySWOFAVwp5ngrTPCoAleC0MBSW1tpHMVPsEhRRViR1DSTtMNn8AxUcvvyzzz  
77a1nrwsmKCAhDYVIRcJSAS8xGqRg+9EIg/iC8E0a/V6yAcmk4vrqzs/  
O8cjLhXGkEhJ0wLC1R/lxipZgVP2tL9UlgFYIRZkdw/  
hze39bPQZptZgdpYRZhd44VDZdYISrDG4G4n76CYR+VKqhUOkDcyB  
+E7y91PqQXR0CYCbviZ0unKkaKVWkJ2xlvBFKxGNfF5rjNhKYmRo8fZRDwamu  
+sSovrlSZg//Hoxg5ZC2exfutg0fKtRR1d/Hiyqbuo2F3BVeHaZpIWY0NeBLyXAB5/  
o1rFzq4t47/oq10yFcyi9caSKUwMVu3o4GSJZY  
+cSHA7ACgs0qLjO0zwwYYgYILeQai2HgF0TuBNmzYIPK49jRrMHC7yyf3vaKSQ2XCR  
NhgmUtG9lNRbLx65/0WJutwtLm9vX0Xu3NdrOU+vY21g9vZUZf8jZaHmMc8mG5h1Vwfl  
+Wd3t7eeWBqbp9Wksx7DaQCZSjtmTvZfl/  
JgGLnBZQACzVRU1NU8ziTRzGIYuGdPMOxLhZAX2k8at7KFAON2DstOP8W60Jqoh  
+dFNR5JrV5uJC2s17r6gpfar2NTsOXXPNXyje+kkCa83Sz/4e/  
5/0GHXMc9fwW8G6aNWvWC7xpYPqsjGN5uckGefS0pTHGq1IY9SS3ru4U  
+StmzeQWVlhqW1vbA9Qi7xemGfdn67EVQMdMP5F8lc/  
g5NpgVjPifWFvxNosnkkjerQVS5YsuYj5Ku  
+S7vL4Gasb4I7+MNXXE4CTyf08LqhWW2rbZvUwQx51EqZ5EXPfxIkT52zcuHFf1r5Uh  
UBygqtKf3rexXpuGqcgzw6+Prq8p6fH/  
DGkNOVmnVcDo9HYInl4otA28PmedR7txj2F6VntZ9oGKjSaNsx3M2fOFIGWkt5aeM64/  
zv+MLwSXf/  
lav34zTffrOvaSPN5pkyZ8jdq6G1gc4kRi9HiP1NL3wh5Phl9lqvjqVAsQPURDdTRb/  
AcZoqOlandsK9dM9/GCfU01YzCaktNBnMPJ+niJ  
+6xd8OebwNIBYp41dJVSeQLIBEd0Kip9INTSICcAw9z7S2tj62adOmL6Q/74smwEfzWu  
+CPD4eZESe5ZDn2Wr7XhMCycmoJtKE/DN8OB0RaSv9Hqt5z/  
tTHzp969B7W9GrN4s8EUcm6ra1uNo1T4xNzQgkAyDRHIB8mTVVwzp2Jt5CptdZVcNt  
A9hDcXottvio7wGoZ3056/U+bcBHNzhvwUfzbFBfdtSUQHICgGdwO3uN3TSP  
+KXwGATgXq7QHjo0d9FgHSol6DOdclR0iRX86oQ07eie7FN/  
pEvTX26APFV52iplf80JJMPUT8STlcZ70vS6lvJxOB0i/YT  
+t9n2se3Tf9UJtNpPqRc9SembhOhegO4FbK9ha/o+j8UI9L8/  
YcKE9mr081SyKxcEkpGrVq1qHRgYeJzd+yoZ7eM8QdDQSD+B7udK7o/2vyJ9UH/608/  
a4v9t6a83+nEJ7ZfJyE9G5iLkp1PDTGdfX0KdniVh0F  
+4PKke5jVr1hwpTKzVfm4IFAogAVgCs56AeG0XxfrrdQtRNAq  
+IsuBURdsckcgOUG7aBok0iOp03wiFyBynucdyHMn7Z29ebMzlwQSSNRampS2kt3HW  
NuUNgaX4dmdjKivpQbKZY+7j06sTOlqwOhh/gfzeNXGWMeaSwAzcf6Er  
+vkuzDIK3nke25roNGBifqMuqmZLht9rpGOlctHrF217Nux4Fk3Blqdgkg3Q6KHWF0nqc  
WqcrWFNO+xroY4VR3LSgtC3REodpintfk0tEWk6+K0etxcmjdolK/  
29a56tTGoWwLFQFEjXQmJVrJ2kHZ2nJ7z7Q8QZwvrWmqc1J9YqaWvdU+gGLyurq4J  
+/fvv43jZZBJk7JSj/THuj1t9TVUvRS4QZ+VS/  
tlME82pVbTMAQqRIJaaQokWkjaAtb57F9QeL5a  
+xBGr2nvZO1jfu1jb5s21BLQxJodlQglAZs5xNEjVVdynYaW69dGOg8hs69bD9m20e7Z  
ieEqelA52gcsjgeEwQaDZxe1jt48ODvSR8ex4JcGtM6n2ONmk

```
+CANpqzGt4FJ3jQY41sq+txtAGSfsGkgyPoXHcT5/  
Nly7/2yJvWAICAYGAQEAglBAQCAgEBAICAYGAQEAglBAQCAgEBAICAYGAQEAglB  
AQCAgEBAICAYEcIvB/Q079+h6myXwAAAAASUVORK5CYII=) 2x);  
}
```

```
@media (prefers-color-scheme: dark) {  
  .ssl .legacy-tls.icon {  
    filter: invert(1);  
  }  
}
```

```
.ssl-opt-in .checkbox::after {  
  border-color: #696969;  
}
```

```
.ssl-opt-in .checkbox::before {  
  border-color: #696969;  
}
```

```
.ssl-enhanced-protection-message {  
  background-color: rgb(248, 249, 250); /* google-gray-50 */  
  border: 1px solid rgb(218, 220, 224); /* google-gray-300 */  
}
```

```
@media (prefers-color-scheme: dark) {  
  .ssl-enhanced-protection-message {  
    background-color: rgb(32, 33, 36); /* google-gray-900 */  
    border: 1px solid rgb(95, 99, 104); /* google-gray-700 */  
  }  
}
```

```
.ssl-enhanced-protection-message .icon {  
  background-image: url(  
ZHRoPSlyNCIgaGVpZ2h0PSlyNCIgzmlsbD0ilzY5Njk2OSI  
+PHBhdGggZD0iTTkgMjFjMCAuNTUuNDUgMSAxIDFoNGMuNTUgMCAxLS40NSAxL  
TF2LTFlOXYxem0zLTE5QzguMTQgMiA1IDUuMTQgNSA5YzAgMi4zOCxLjE5IDQuN  
DcgMyA1Ljc0VjE3YzAgLjU1LjQ1IDEgMSAxIDAgMS0uNDUgMS0xdjE5LjE5IDQuN  
EuODEtMS4yNyAzLTMuMzYgMy01Ljc0IDAtMy44Ni0zLjE0LTctNy03em0yLjE5IDExLjFs  
LS44NS42VjE2aC00di0yLjNsLS44NS0uNkM3LjggMTluMTYgNyAxMC42MyA3IDljMC0  
yLjc2IDluMjQ1NSA1LTVzNSA5LjE0IDUgNWwIDEuNjMtLjggMy4xNi0yLjE5IDQuMXoiL  
z48L3N2Zz4K);  
}
```

```
.ssl-enhanced-protection-message a:link {  
  color: rgb(26, 115, 232); /* google-blue-600 */
```

```
}
```

```
@media (prefers-color-scheme: dark) {  
  .ssl-enhanced-protection-message a:link {  
    color: rgb(138, 180, 248); /* google-blue-300 */  
  }  
}
```

```
</style>
```

```
<script>// Copyright (c) 2013 The Chromium Authors. All rights reserved.  
// Use of this source code is governed by a BSD-style license that can be  
// found in the LICENSE file.
```

```
/**
```

```
 * @fileoverview Assertion support.
```

```
 */
```

```
/**
```

```
 * Verify !condition is truthy and return !condition if so.
```

```
 * @template T
```

```
 * @param {T} condition A condition to check for truthiness. Note that this
```

```
 *   may be used to test whether a value is defined or not, and we don't want
```

```
 *   to force a cast to Boolean.
```

```
 * @param {string=} opt_message A message to show on failure.
```

```
 * @return {T} A non-null !condition.
```

```
 * @closurePrimitive {asserts.truthy}
```

```
 * @suppress {reportUnknownTypes} because T is not sufficiently constrained.
```

```
 */
```

```
/* #export */ function assert(condition, opt_message) {
```

```
  if (!condition) {
```

```
    let message = 'Assertion failed';
```

```
    if (opt_message) {
```

```
      message = message + ': ' + opt_message;
```

```
    }
```

```
    const error = new Error(message);
```

```
    const global = function() {
```

```
      const thisOrSelf = this || self;
```

```
      /** @type {boolean} */
```

```
      thisOrSelf.traceAssertionsForTesting;
```

```
      return thisOrSelf;
```

```
    }());
```

```
    if (global.traceAssertionsForTesting) {
```

```
      console.warn(error.stack);
```

```
    }
```

```
    throw error;
```

```
  }
```

```
  return condition;
```

```
}
```

```
/**
```

```
 * Call this from places in the code that should never be reached.
```

```
 *
```

```
 * For example, handling all the values of enum with a switch() like this:
```

```
 *
```

```
 * function getValueFromEnum(enum) {
```

```
 *   switch (enum) {
```

```
 *     case ENUM_FIRST_OF_TWO:
```

```
 *       return first
```

```
 *     case ENUM_LAST_OF_TWO:
```

```
 *       return last;
```

```
 *   }
```

```
 *   assertNotReached();
```

```
 *   return document;
```

```
 * }
```

```
 *
```

```
 * This code should only be hit in the case of serious programmer error or
```

```
 * unexpected input.
```

```
 *
```

```
 * @param {string=} opt_message A message to show when this is hit.
```

```
 * @closurePrimitive {asserts.fail}
```

```
 */
```

```
/* #export */ function assertNotReached(opt_message) {
```

```
  assert(false, opt_message || 'Unreachable code hit');
```

```
}
```

```
/**
```

```
 * @param {*} value The value to check.
```

```
 * @param {function(new: T, ...)} type A user-defined constructor.
```

```
 * @param {string=} opt_message A message to show when this is hit.
```

```
 * @return {T}
```

```
 * @template T
```

```
 */
```

```
/* #export */ function assertInstanceOf(value, type, opt_message) {
```

```
  // We don't use assert immediately here so that we avoid constructing an error
```

```
  // message if we don't have to.
```

```
  if (!(value instanceof type)) {
```

```
    assertNotReached(
```

```
      opt_message ||
```

```
      'Value ' + value + ' is not a[n] ' + (type.name || typeof type));
```

```
  }
```

```
  return value;
```

```
}
```

```

/* #ignore */ console.warn('crbug/1173575, non-JS module files deprecated.');
```

</script>
<script>
// Copyright (c) 2012 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.

// #import {assertInstanceOf} from './assert.m.js';
// #import {dispatchSimpleEvent} from './cr.m.js';

/\*\*
 \* Alias for document.getElementById. Found elements must be HTMLElements.
 \* @param {string} id The ID of the element to find.
 \* @return {HTMLElement} The found element or null if not found.
 \*/
/\* #export \*/ function \$(id) {
 // Disable getElementById restriction here, since we are instructing other
 // places to re-use the \$() that is defined here.
 // eslint-disable-next-line no-restricted-properties
 const el = document.getElementById(id);
 return el ? instanceof(el, HTMLElement) : null;
}

// TODO(devlin): This should return SVGElement, but closure compiler is missing
// those externs.
/\*\*
 \* Alias for document.getElementById. Found elements must be SVGElements.
 \* @param {string} id The ID of the element to find.
 \* @return {Element} The found element or null if not found.
 \*/
/\* #export \*/ function getSVGElement(id) {
 // Disable getElementById restriction here, since it is not suitable for SVG
 // elements.
 // eslint-disable-next-line no-restricted-properties
 const el = document.getElementById(id);
 return el ? instanceof(el, Element) : null;
}

/\*\*
 \* @return {?Element} The currently focused element (including elements that are
 \* behind a shadow root), or null if nothing is focused.
 \*/
/\* #export \*/ function getDeepActiveElement() {
 let a = document.activeElement;
 while (a && a.shadowRoot && a.shadowRoot.activeElement) {
 a = a.shadowRoot.activeElement;
 }
}

```

    return a;
}

//

/**
 * @param {Node} el A node to search for ancestors with lclassName.
 * @param {string} className A class to search for.
 * @return {Element} A node with class of lclassName or null if none is found.
 */
/* #export */ function findAncestorByClass(el, className) {
    return /** @type {Element} */ (findAncestor(el, function(el) {
        return el.classList && el.classList.contains(className);
    }));
}

/**
 * Return the first ancestor for which the {@code predicate} returns true.
 * @param {Node} node The node to check.
 * @param {function(Node):boolean} predicate The function that tests the
 *     nodes.
 * @param {boolean=} includeShadowHosts
 * @return {Node} The found ancestor or null if not found.
 */
/* #export */ function findAncestor(node, predicate, includeShadowHosts) {
    while (node !== null) {
        if (predicate(node)) {
            break;
        }
        node = includeShadowHosts && node instanceof ShadowRoot ? node.host :
            node.parentNode;
    }
    return node;
}

/**
 * Disables text selection and dragging, with optional callbacks to specify
 * overrides.
 * @param {function(Event):boolean=} opt_allowSelectStart Unless this function
 *     is defined and returns true, the onselectionstart event will be
 *     suppressed.
 * @param {function(Event):boolean=} opt_allowDragStart Unless this function
 *     is defined and returns true, the ondragstart event will be suppressed.
 */
/* #export */ function disableTextSelectAndDrag(
    opt_allowSelectStart, opt_allowDragStart) {

```

```

// Disable text selection.
document.onselectstart = function(e) {
    if (!(opt_allowSelectStart && opt_allowSelectStart.call(this, e))) {
        e.preventDefault();
    }
};

// Disable dragging.
document.ondragstart = function(e) {
    if (!(opt_allowDragStart && opt_allowDragStart.call(this, e))) {
        e.preventDefault();
    }
};
}

/**
 * Check the directionality of the page.
 * @return {boolean} True if Chrome is running an RTL UI.
 */
/* #export */ function isRTL() {
    return document.documentElement.dir === 'rtl';
}

/**
 * Get an element that's known to exist by its ID. We use this instead of just
 * calling getElementById and not checking the result because this lets us
 * satisfy the JSCompiler type system.
 * @param {string} id The identifier name.
 * @return {!HTMLElement} the Element.
 */
/* #export */ function getRequiredElement(id) {
    return assertInstanceOf(
        $(id), HTMLElement, 'Missing required element: ' + id);
}

/**
 * Query an element that's known to exist by a selector. We use this instead of
 * just calling querySelector and not checking the result because this lets us
 * satisfy the JSCompiler type system.
 * @param {string} selectors CSS selectors to query the element.
 * @param {(!Document!DocumentFragment!Element)=} opt_context An optional
 * context object for querySelector.
 * @return {!HTMLElement} the Element.
 */
/* #export */ function queryRequiredElement(selectors, opt_context) {
    const element = (opt_context || document).querySelector(selectors);

```



```

return assertInstanceOf(
    element, HTMLElement, 'Missing required element: ' + selectors);
}

/**
 * Creates a new URL which is the old URL with a GET param of key=value.
 * @param {string} url The base URL. There is not sanity checking on the URL so
 * it must be passed in a proper format.
 * @param {string} key The key of the param.
 * @param {string} value The value of the param.
 * @return {string} The new URL.
 */
/* #export */ function appendParam(url, key, value) {
    const param = encodeURIComponent(key) + '=' + encodeURIComponent(value);

    if (url.indexOf('?') === -1) {
        return url + '?' + param;
    }
    return url + '&' + param;
}

/**
 * Creates an element of a specified type with a specified class name.
 * @param {string} type The node type.
 * @param {string} className The class name to use.
 * @return {Element} The created element.
 */
/* #export */ function createElementWithClassName(type, className) {
    const elm = document.createElement(type);
    elm.className = className;
    return elm;
}

/**
 * transitionend does not always fire (e.g. when animation is aborted
 * or when no paint happens during the animation). This function sets up
 * a timer and emulate the event if it is not fired when the timer expires.
 * @param {HTMLElement} el The element to watch for transitionend.
 * @param {number=} opt_timeOut The maximum wait time in milliseconds for the
 * transitionend to happen. If not specified, it is fetched from !ell
 * using the transitionDuration style value.
 */
/* #export */ function ensureTransitionEndEvent(el, opt_timeOut) {
    if (opt_timeOut === undefined) {
        const style = getComputedStyle(el);
        opt_timeOut = parseFloat(style.transitionDuration) * 1000;
    }

```

```

    // Give an additional 50ms buffer for the animation to complete.
    opt_timeout += 50;
}

let fired = false;
el.addEventListener('transitionend', function f(e) {
    el.removeEventListener('transitionend', f);
    fired = true;
});
window.setTimeout(function() {
    if (!fired) {
        cr.dispatchSimpleEvent(el, 'transitionend', true);
    }
}, opt_timeout);
}

/**
 * Alias for document.scrollTop getter.
 * @param {!HTMLDocument} doc The document node where information will be
 *   queried from.
 * @return {number} The Y document scroll offset.
 */
/* #export */ function scrollTopForDocument(doc) {
    return doc.documentElement.scrollTop || doc.body.scrollTop;
}

/**
 * Alias for document.scrollTop setter.
 * @param {!HTMLDocument} doc The document node where information will be
 *   queried from.
 * @param {number} value The target Y scroll offset.
 */
/* #export */ function setScrollTopForDocument(doc, value) {
    doc.documentElement.scrollTop = doc.body.scrollTop = value;
}

/**
 * Alias for document.scrollLeft getter.
 * @param {!HTMLDocument} doc The document node where information will be
 *   queried from.
 * @return {number} The X document scroll offset.
 */
/* #export */ function scrollLeftForDocument(doc) {
    return doc.documentElement.scrollLeft || doc.body.scrollLeft;
}

```

```

/**
 * Alias for document.scrollLeft setter.
 * @param {!HTMLDocument} doc The document node where information will be
 *   queried from.
 * @param {number} value The target X scroll offset.
 */
/* #export */ function setScrollLeftForDocument(doc, value) {
  doc.documentElement.scrollLeft = doc.body.scrollLeft = value;
}

```

```

/**
 * Replaces '&', '<', '>', '"', and "'" characters with their HTML encoding.
 * @param {string} original The original string.
 * @return {string} The string with all the characters mentioned above replaced.
 */
/* #export */ function HTMLEscape(original) {
  return original.replace(/&/g, '&amp;')
    .replace(/</g, '&lt;')
    .replace(/>/g, '&gt;')
    .replace(/"/g, '&quot;')
    .replace(/'/g, '&#39;');
}

```

```

/**
 * Shortens the provided string (if necessary) to a string of length at most
 * maxLength.
 * @param {string} original The original string.
 * @param {number} maxLength The maximum length allowed for the string.
 * @return {string} The original string if its length does not exceed
 *   maxLength. Otherwise the first maxLength - 1 characters with '...'
 *   appended.
 */
/* #export */ function elide(original, maxLength) {
  if (original.length <= maxLength) {
    return original;
  }
  return original.substring(0, maxLength - 1) + '\u2026';
}

```

```

/**
 * Quote a string so it can be used in a regular expression.
 * @param {string} str The source string.
 * @return {string} The escaped string.
 */
/* #export */ function quoteString(str) {

```

```

return str.replace(/([\\.\+\\*\\?\\[\\^]\\$\\(\\)\\{\\}\\=\\!\\<\\>\\:\\])/g, '\\$1');
}

/**
 * Calls lcallbackl and stops listening the first time any event in leventNamesl
 * is triggered on ltargetl.
 * @param {!EventTarget} target
 * @param {!Array<string>|string} eventNames Array or space-delimited string of
 *   event names to listen to (e.g. 'click mousedown').
 * @param {function(!Event)} callback Called at most once. The
 *   optional return value is passed on by the listener.
 */
/* #export */ function listenOnce(target, eventNames, callback) {
  if (!Array.isArray(eventNames)) {
    eventNames = eventNames.split(/ +/);
  }

  const removeAllAndCallCallback = function(event) {
    eventNames.forEach(function(eventName) {
      target.removeEventListener(eventName, removeAllAndCallCallback, false);
    });
    return callback(event);
  };

  eventNames.forEach(function(eventName) {
    target.addEventListener(eventName, removeAllAndCallCallback, false);
  });
}

/**
 * @param {!Event} e
 * @return {boolean} Whether a modifier key was down when processing lel.
 */
/* #export */ function hasKeyModifiers(e) {
  return !(e.altKey || e.ctrlKey || e.metaKey || e.shiftKey);
}

/**
 * @param {!Element} el
 * @return {boolean} Whether the element is interactive via text input.
 */
/* #export */ function isTextInputElement(el) {
  return el.tagName === 'INPUT' || el.tagName === 'TEXTAREA';
}

/* #ignore */ console.warn('crbug/1173575, non-JS module files deprecated.');
```

```

</script>
  <script>// Copyright 2014 The Chromium Authors. All rights reserved.
  // Use of this source code is governed by a BSD-style license that can be
  // found in the LICENSE file.

  const CAPTIVEPORTAL_CMD_OPEN_LOGIN_PAGE = 'openLoginPage';
</script>
  <script>// Copyright 2014 The Chromium Authors. All rights reserved.
  // Use of this source code is governed by a BSD-style license that can be
  // found in the LICENSE file.

  function setupSSLDebuggingInfo() {
    if (loadTimeData.getString('type') !== 'SSL') {
      return;
    }

    // The titles are not internationalized because this is debugging information
    // for bug reports, help center posts, etc.
    appendDebuggingField('Subject', loadTimeData.getString('subject'));
    appendDebuggingField('Issuer', loadTimeData.getString('issuer'));
    appendDebuggingField('Expires on', loadTimeData.getString('expirationDate'));
    appendDebuggingField('Current date', loadTimeData.getString('currentDate'));
    appendDebuggingField('PEM encoded chain', loadTimeData.getString('pem'),
                          true);
    const ctInfo = loadTimeData.getString('ct');
    if (ctInfo) {
      appendDebuggingField('Certificate Transparency', ctInfo);
    }

    $('error-code').addEventListener('click', toggleDebuggingInfo);
    $('error-code').setAttribute('role', 'button');
    $('error-code').setAttribute('aria-expanded', false);
  }
</script>
  <script>// Copyright 2015 The Chromium Authors. All rights reserved.
  // Use of this source code is governed by a BSD-style license that can be
  // found in the LICENSE file.

  'use strict';

  // Other constants defined in security_interstitial_page.h.
  const SB_BOX_CHECKED = 'boxchecked';
  const SB_DISPLAY_CHECK_BOX = 'displaycheckbox';

  // This sets up the Extended Safe Browsing Reporting opt-in, either for
  // reporting malware or invalid certificate chains. Does nothing if the

```

```

// interstitial type is not SAFEBROWSING or SSL or CAPTIVE_PORTAL.
function setupExtendedReportingCheckbox() {
  const interstitialType = loadTimeData.getString('type');
  if (interstitialType !== 'SAFEBROWSING' && interstitialType !== 'SSL' &&
    interstitialType !== 'CAPTIVE_PORTAL') {
    return;
  }

  if (!loadTimeData.getBoolean(SB_DISPLAY_CHECK_BOX)) {
    return;
  }

  if ($('#privacy-link')) {
    $('#privacy-link').addEventListener('click', function() {

sendCommand(SecurityInterstitialCommandId.CMD_OPEN_REPORTING_PRIVACY);
      return false;
    });
    $('#privacy-link').addEventListener('mousedown', function() {
      return false;
    });
  }
  $('#opt-in-checkbox').checked = loadTimeData.getBoolean(SB_BOX_CHECKED);
  $('#extended-reporting-opt-in').classList.remove('hidden');

  const billing =
    interstitialType === 'SAFEBROWSING' && loadTimeData.getBoolean('billing');

  let className = 'ssl-opt-in';
  if (interstitialType === 'SAFEBROWSING' && !billing) {
    className = 'safe-browsing-opt-in';
  }

  $('#extended-reporting-opt-in').classList.add(className);

  $('#body').classList.add('extended-reporting-has-checkbox');

  if ($('#whitepaper-link')) {
    $('#whitepaper-link').addEventListener('click', function(event) {
      sendCommand(SecurityInterstitialCommandId.CMD_OPEN_WHITEPAPER);
    });
  }

  $('#opt-in-checkbox').addEventListener('click', function() {
    sendCommand($('#opt-in-checkbox').checked ?
      SecurityInterstitialCommandId.CMD_DO_REPORT :

```

```

        SecurityInterstitialCommandId.CMD_DONT_REPORT);
    });
}
</script>
<script>// Copyright 2020 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.

'use strict';

// Other constants defined in security_interstitial_page.h.
const SB_DISPLAY_ENHANCED_PROTECTION_MESSAGE =
    'displayEnhancedProtectionMessage';

// This sets up the enhanced protection message.
function setupEnhancedProtectionMessage() {
    const interstitialType = loadTimeData.getString('type');
    if (interstitialType !== 'SAFEBROWSING' && interstitialType !== 'SSL' &&
        interstitialType !== 'CAPTIVE_PORTAL') {
        return;
    }

    if (!
loadTimeData.getBoolean(SB_DISPLAY_ENHANCED_PROTECTION_MESSAGE)) {
        return;
    }

    if ($('#enhanced-protection-link')) {
        if (mobileNav) {
            // To make sure the touch area of the link is larger than the
            // minimum touch area for accessibility, make the whole block tappable.
            $('#enhanced-protection-message').addEventListener('click', function() {
                sendCommand(SecurityInterstitialCommandId
                    .CMD_OPEN_ENHANCED_PROTECTION_SETTINGS);
                return false;
            });
        } else {
            $('#enhanced-protection-link').addEventListener('click', function() {
                sendCommand(SecurityInterstitialCommandId
                    .CMD_OPEN_ENHANCED_PROTECTION_SETTINGS);
                return false;
            });
        }
    }
    $('#enhanced-protection-message').classList.remove('hidden');
}

```

```

const billing =
  interstitialType === 'SAFEBROWSING' && loadTimeData.getBoolean('billing');

let className = 'ssl-enhanced-protection-message';
if (interstitialType === 'SAFEBROWSING' && !billing) {
  className = 'safe-browsing-enhanced-protection-message';
}

$('enhanced-protection-message').classList.add(className);
}
</script>
<script>
  // Copyright 2015 The Chromium Authors. All rights reserved.
  // Use of this source code is governed by a BSD-style license that can be
  // found in the LICENSE file.

```

```

let mobileNav = false;

```

```

/**
 * For small screen mobile the navigation buttons are moved
 * below the advanced text.
 */
function onResize() {
  const helpOuterBox = document.querySelector('#details');
  const mainContent = document.querySelector('#main-content');
  const mediaQuery = '(min-width: 240px) and (max-width: 420px) and ' +
    '(min-height: 401px), ' +
    '(max-height: 560px) and (min-height: 240px) and ' +
    '(min-width: 421px)';

  const detailsHidden = helpOuterBox.classList.contains(HIDDEN_CLASS);
  const runnerContainer = document.querySelector('.runner-container');

  // Check for change in nav status.
  if (mobileNav !== window.matchMedia(mediaQuery).matches) {
    mobileNav = !mobileNav;

    // Handle showing the top content / details sections according to state.
    if (mobileNav) {
      mainContent.classList.toggle(HIDDEN_CLASS, !detailsHidden);
      helpOuterBox.classList.toggle(HIDDEN_CLASS, detailsHidden);
      if (runnerContainer) {
        runnerContainer.classList.toggle(HIDDEN_CLASS, !detailsHidden);
      }
    } else if (!detailsHidden) {
      // Non mobile nav with visible details.
      mainContent.classList.remove(HIDDEN_CLASS);
    }
  }
}

```



```

    helpOuterBox.classList.remove(HIDDEN_CLASS);
    if (runnerContainer) {
        runnerContainer.classList.remove(HIDDEN_CLASS);
    }
}
}
}

function setupMobileNav() {
    window.addEventListener('resize', onResize);
    onResize();
}

document.addEventListener('DOMContentLoaded', setupMobileNav);
</script>
<script>// Copyright 2017 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.

// This is the shared code for security interstitials. It is used for both SSL
// interstitials and Safe Browsing interstitials.

/**
 * @typedef {{
 *   dontProceed: function(),
 *   proceed: function(),
 *   showMoreSection: function(),
 *   openHelpCenter: function(),
 *   openDiagnostic: function(),
 *   reload: function(),
 *   openDateSettings: function(),
 *   openLogin: function(),
 *   doReport: function(),
 *   dontReport: function(),
 *   openReportingPrivacy: function(),
 *   openWhitepaper: function(),
 *   reportPhishingError: function(),
 * }}
 */
// eslint-disable-next-line no-var
var certificateErrorPageController;

// Should match security_interstitials::SecurityInterstitialCommand
/** @enum {number} */
const SecurityInterstitialCommandId = {
    CMD_DONT_PROCEED: 0,

```

```

CMD_PROCEED: 1,
// Ways for user to get more information
CMD_SHOW_MORE_SECTION: 2,
CMD_OPEN_HELP_CENTER: 3,
CMD_OPEN_DIAGNOSTIC: 4,
// Primary button actions
CMD_RELOAD: 5,
CMD_OPEN_DATE_SETTINGS: 6,
CMD_OPEN_LOGIN: 7,
// Safe Browsing Extended Reporting
CMD_DO_REPORT: 8,
CMD_DONT_REPORT: 9,
CMD_OPEN_REPORTING_PRIVACY: 10,
CMD_OPEN_WHITEPAPER: 11,
// Report a phishing error.
CMD_REPORT_PHISHING_ERROR: 12,
// Open enhanced protection settings.
CMD_OPEN_ENHANCED_PROTECTION_SETTINGS: 13,
};

```

```

const HIDDEN_CLASS = 'hidden';

```

```

/**
 * A convenience method for sending commands to the parent page.
 * @param {SecurityInterstitialCommandId} cmd The command to send.
 */
function sendCommand(cmd) {
  if (window.certificateErrorPageController) {
    switch (cmd) {
      case SecurityInterstitialCommandId.CMD_DONT_PROCEED:
        certificateErrorPageController.dontProceed();
        break;
      case SecurityInterstitialCommandId.CMD_PROCEED:
        certificateErrorPageController.proceed();
        break;
      case SecurityInterstitialCommandId.CMD_SHOW_MORE_SECTION:
        certificateErrorPageController.showMoreSection();
        break;
      case SecurityInterstitialCommandId.CMD_OPEN_HELP_CENTER:
        certificateErrorPageController.openHelpCenter();
        break;
      case SecurityInterstitialCommandId.CMD_OPEN_DIAGNOSTIC:
        certificateErrorPageController.openDiagnostic();
        break;
      case SecurityInterstitialCommandId.CMD_RELOAD:
        certificateErrorPageController.reload();

```

```

        break;
    case SecurityInterstitialCommandId.CMD_OPEN_DATE_SETTINGS:
        certificateErrorPageController.openDateSettings();
        break;
    case SecurityInterstitialCommandId.CMD_OPEN_LOGIN:
        certificateErrorPageController.openLogin();
        break;
    case SecurityInterstitialCommandId.CMD_DO_REPORT:
        certificateErrorPageController.doReport();
        break;
    case SecurityInterstitialCommandId.CMD_DONT_REPORT:
        certificateErrorPageController.dontReport();
        break;
    case SecurityInterstitialCommandId.CMD_OPEN_REPORTING_PRIVACY:
        certificateErrorPageController.openReportingPrivacy();
        break;
    case SecurityInterstitialCommandId.CMD_OPEN_WHITEPAPER:
        certificateErrorPageController.openWhitepaper();
        break;
    case SecurityInterstitialCommandId.CMD_REPORT_PHISHING_ERROR:
        certificateErrorPageController.reportPhishingError();
        break;
    case
SecurityInterstitialCommandId.CMD_OPEN_ENHANCED_PROTECTION_SETTINGS:
        certificateErrorPageController.openEnhancedProtectionSettings();
        break;
    }
    return;
}
//
window.domAutomationController.send(cmd);
//
//
}

/**
 * Call this to stop clicks on <a href="#"> links from scrolling to the top of
 * the page (and possibly showing a # in the link).
 */
function preventDefaultOnPoundLinkClicks() {
    document.addEventListener('click', function(e) {
        const anchor = findAncestor(/** @type {Node} */ (e.target), function(el) {
            return el.tagName === 'A';
        });
        // Use getAttribute() to prevent URL normalization.
        if (anchor && anchor.getAttribute('href') === '#') {

```

```

    e.preventDefault();
  }
});
}

//

//
</script>
<script>// Copyright 2014 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.

// This is the shared code for the new (Chrome 37) security interstitials. It is
// used for both SSL interstitials and Safe Browsing interstitials.

let expandedDetails = false;
let keyPressState = 0;

/**
 * This allows errors to be skipped by typing a secret phrase into the page.
 * @param {string} e The key that was just pressed.
 */
function handleKeypress(e) {
  // HTTPS errors are serious and should not be ignored. For testing purposes,
  // other approaches are both safer and have fewer side-effects.
  // See https://goo.gl/ZcZixP for more details.
  const BYPASS_SEQUENCE = window.atob('dGhpc2lzdW5zYWZl');
  if (BYPASS_SEQUENCE.charCodeAt(keyPressState) === e.keyCode) {
    keyPressState++;
    if (keyPressState === BYPASS_SEQUENCE.length) {
      sendCommand(SecurityInterstitialCommandId.CMD_PROCEED);
      keyPressState = 0;
    }
  } else {
    keyPressState = 0;
  }
}

/**
 * This appends a piece of debugging information to the end of the warning.
 * When complete, the caller must also make the debugging div
 * (error-debugging-info) visible.
 * @param {string} title The name of this debugging field.
 * @param {string} value The value of the debugging field.
 * @param {boolean=} fixedWidth If true, the value field is displayed fixed

```

```

*           width.
*/
function appendDebuggingField(title, value, fixedWidth) {
  // The values input here are not trusted. Never use innerHTML on these
  // values!
  const spanTitle = document.createElement('span');
  spanTitle.classList.add('debugging-title');
  spanTitle.innerText = title + ': ';

  const spanValue = document.createElement('span');
  spanValue.classList.add('debugging-content');
  if (fixedWidth) {
    spanValue.classList.add('debugging-content-fixed-width');
  }
  spanValue.innerText = value;

  const pElem = document.createElement('p');
  pElem.classList.add('debugging-content');
  pElem.appendChild(spanTitle);
  pElem.appendChild(spanValue);
  $('error-debugging-info').appendChild(pElem);
}

function toggleDebuggingInfo() {
  const hiddenDebug = $('error-debugging-info').classList.toggle(HIDDEN_CLASS);
  $('error-code').setAttribute('aria-expanded', !hiddenDebug);
}

function setupEvents() {
  const overridable = loadTimeData.getBoolean('overridable');
  const interstitialType = loadTimeData.getString('type');
  const ssl = interstitialType === 'SSL';
  const captivePortal = interstitialType === 'CAPTIVE_PORTAL';
  const badClock = ssl && loadTimeData.getBoolean('bad_clock');
  const lookalike = interstitialType === 'LOOKALIKE';
  const billing =
    interstitialType === 'SAFEBROWSING' && loadTimeData.getBoolean('billing');
  const originPolicy = interstitialType === 'ORIGIN_POLICY';
  const blockedInterception = interstitialType === 'BLOCKED_INTERCEPTION';
  const legacyTls = interstitialType === 'LEGACY_TLS';
  const insecureForm = interstitialType === 'INSECURE_FORM';
  const httpsOnly = interstitialType === 'HTTPS_ONLY';
  const hidePrimaryButton = loadTimeData.getBoolean('hide_primary_button');
  const showRecurrentErrorParagraph = loadTimeData.getBoolean(
    'show_recurrent_error_paragraph');

```

```

if (ssl || originPolicy || blockedInterception || legacyTls) {
  $('body').classList.add(badClock ? 'bad-clock' : 'ssl');
  if (loadTimeData.valueExists('errorCode')) {
    $('error-code').textContent = loadTimeData.getString('errorCode');
    $('error-code').classList.remove(HIDDEN_CLASS);
  }
} else if (captivePortal) {
  $('body').classList.add('captive-portal');
} else if (billing) {
  $('body').classList.add('safe-browsing-billing');
} else if (lookalike) {
  $('body').classList.add('lookalike-url');
} else if (insecureForm) {
  $('body').classList.add('insecure-form');
} else if (httpsOnly) {
  $('body').classList.add('https-only');
} else {
  $('body').classList.add('safe-browsing');
  // Override the default theme color.
  document.querySelector('meta[name=theme-color]').setAttribute('content',
    'rgb(217, 48, 37)');
}

$('icon').classList.add('icon');

if (legacyTls) {
  $('icon').classList.add('legacy-tls');
}

if (hidePrimaryButton) {
  $('primary-button').classList.add(HIDDEN_CLASS);
} else {
  $('primary-button').addEventListener('click', function() {
    switch (interstitialType) {
      case 'CAPTIVE_PORTAL':
        sendCommand(SecurityInterstitialCommandId.CMD_OPEN_LOGIN);
        break;

      case 'SSL':
      case 'LEGACY_TLS':
        if (badClock) {
          sendCommand(SecurityInterstitialCommandId.CMD_OPEN_DATE_SETTINGS);
        } else if (overridable) {
          sendCommand(SecurityInterstitialCommandId.CMD_DONT_PROCEED);
        } else {

```

```

        sendCommand(SecurityInterstitialCommandId.CMD_RELOAD);
    }
    break;

    case 'SAFEBROWSING':
    case 'ORIGIN_POLICY':
        sendCommand(SecurityInterstitialCommandId.CMD_DONT_PROCEED);
        break;
    case 'HTTPS_ONLY':
    case 'INSECURE_FORM':
    case 'LOOKALIKE':
        sendCommand(SecurityInterstitialCommandId.CMD_DONT_PROCEED);
        break;

    default:
        throw 'Invalid interstitial type';
    }
});
}

if (lookalike || insecureForm || httpsOnly) {
    const proceedButton = 'proceed-button';
    $(proceedButton).classList.remove(HIDDEN_CLASS);
    $(proceedButton).textContent = loadTimeData.getString('proceedButtonText');
    $(proceedButton).addEventListener('click', function(event) {
        sendCommand(SecurityInterstitialCommandId.CMD_PROCEED);
    });
}

if (lookalike) {
    // Lookalike interstitials with a suggested URL have a link in the title:
    // "Did you mean <link>example.com</link>?". Handle those clicks. Lookalike
    // interstitials without a suggested URL don't have this link.
    const dontProceedLink = 'dont-proceed-link';
    if ($(dontProceedLink)) {
        $(dontProceedLink).addEventListener('click', function(event) {
            sendCommand(SecurityInterstitialCommandId.CMD_DONT_PROCEED);
        });
    }
}

if (overridable) {
    const overrideElement = billing ? 'proceed-button' : 'proceed-link';
    // Captive portal page isn't overridable.
    $(overrideElement).addEventListener('click', function(event) {
        sendCommand(SecurityInterstitialCommandId.CMD_PROCEED);
    });
}

```

```

if (ssl) {
    $(overrideElement).classList.add('small-link');
} else if (billing) {
    $(overrideElement).classList.remove(HIDDEN_CLASS);
    $(overrideElement).textContent =
        loadTimeData.getString('proceedButtonText');
}
} else if (!ssl) {
    $('final-paragraph').classList.add(HIDDEN_CLASS);
}

if (!ssl || !showRecurrentErrorParagraph) {
    $('recurrent-error-message').classList.add(HIDDEN_CLASS);
} else {
    $('body').classList.add('showing-recurrent-error-message');
}

if ($('#diagnostic-link')) {
    $('#diagnostic-link').addEventListener('click', function(event) {
        sendCommand(SecurityInterstitialCommandId.CMD_OPEN_DIAGNOSTIC);
    });
}

if ($('#learn-more-link')) {
    $('#learn-more-link').addEventListener('click', function(event) {
        sendCommand(SecurityInterstitialCommandId.CMD_OPEN_HELP_CENTER);
    });
}

if (captivePortal || billing || lookalike || insecureForm || httpsOnly) {
    // Captive portal, billing, lookalike pages, insecure form, and
    // HTTPS only mode interstitials don't have details buttons.
    $('details-button').classList.add('hidden');
} else {
    $('details-button')
        .setAttribute(
            'aria-expanded', !$('#details').classList.contains(HIDDEN_CLASS));
    $('details-button').addEventListener('click', function(event) {
        const hiddenDetails = $('#details').classList.toggle(HIDDEN_CLASS);
        $('details-button').setAttribute('aria-expanded', !hiddenDetails);
    });

    if (mobileNav) {
        // Details appear over the main content on small screens.
        $('main-content').classList.toggle(HIDDEN_CLASS, !hiddenDetails);
    }
}

```



```

    } else {
        $('main-content').classList.remove(HIDDEN_CLASS);
    }

    $('details-button').innerText = hiddenDetails ?
        loadTimeData.getString('openDetails') :
        loadTimeData.getString('closeDetails');
    if (!expandedDetails) {
        // Record a histogram entry only the first time that details is opened.
        sendCommand(SecurityInterstitialCommandId.CMD_SHOW_MORE_SECTION);
        expandedDetails = true;
    }
    });
}

if ($('report-error-link')) {
    $('report-error-link').addEventListener('click', function(event) {

sendCommand(SecurityInterstitialCommandId.CMD_REPORT_PHISHING_ERROR);
    });
}

if (lookalike) {
    console.log(
        'Chrome has determined that ' +
        loadTimeData.getString('lookalikeRequestHostname') +
        ' could be fake or fraudulent.\n\n' +
        'If you believe this is shown in error please visit ' +
        'https://g.co/chrome/lookalike-warnings');
}

preventDefaultOnPoundLinkClicks();
setupExtendedReportingCheckbox();
setupEnhancedProtectionMessage();
setupSSLDebuggingInfo();
document.addEventListener('keypress', handleKeypress);
}

document.addEventListener('DOMContentLoaded', setupEvents);
</script>
</head>
<body id="body" class="ssl">
    <div class="interstitial-wrapper">
        <div id="main-content" class="hidden">
            <div class="icon" id="icon"></div>
            <div id="main-message">

```

# Your connection is not private

Attackers might be trying to steal your information from **accounts.shopify.com** (for example, passwords, messages, or credit cards). [Learn more](#)

net::ERR\_CERT\_DATE\_INVALID

Subject:accounts.shopify.com

Issuer:R3

Expires on:Nov 13, 2021

Current date:Oct 5, 2021

PEM encoded chain:-----BEGIN CERTIFICATE-----  
MIIF2DCCBMCGAwIBAgISA3IP  
+wuRtzmomjRFbuWpyrsCMA0GCSqGSIb3DQEBCwUA  
MDIxGzAJBgNVBAYTAIV  
TMRYwFAYDVQQKEw1MZXRQncyBFbmNyeXB0MQswCQYDVQQD  
EwJSMzAeFw  
0yMTA4MTUxNjE3MDNaFw0yMTEyMTMxNjE3MDFaMmB8xHTAbBgNVBAMT  
FGFj  
Y291bnRzLnNob3BpZnkuY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB  
CgK  
CAQEAmNO6kENQqqUM0WgozXecZGcPzrlyEbrU/  
N7mdStXBNjXN9WAEbLSTGrw  
gkyShj1dbDXVJudJ7YdQXJr/  
KJbm87u8a1HEZI4Dmj1FxBIAd+Gs1RsFdQd0nEw  
5Zgi0Sojj6d87SEqpS2i0Vz6V  
+hTrbuy+XXP5+J8piNiRp2lkrZQkWKaifZaLyvH  
w2S2OfEtEjxH3SSoKfsA/  
ST4ZPRDFIppsXgmpFf7sU5u1E3XckNjxl220Q29/  
vtls  
ErLP9noTy1wdKzXJYbu4IT6nVOSs  
+TGLQBcBCfNQyaG19jMrL299G4QyNy2h6uWX  
MwPm/Opq3qk3Bnl5uZgyV  
+mx7EMe1QIDAQABo4IC+TCCAvUwDgYDVR0PAQH/  
BAQD  
AgWgMB0GA1UdJQQWMBQGCCsGAQUFBwMBBggrBgEFBQcDAjAMBg  
NVHRMBAf8EAjAA  
MB0GA1UdDgQWBBR/TrZ5x8Ma/  
wcElldjg8fyl3L3zAfBgNVHSMEGDAWgBQULrMX  
t1hWy65QCUDmH6+dixTCxjBVB  
ggrBgEFBQcBAQRJMEcwIQYIKwYBBQUHMAGGFWH0  
dHA6Ly9yMy5vLmxlbmN  
yLm9yZzAiBggrBgEFBQcwAoYWAHR0cDovL3lzLmkubGVu  
Y3lub3JnLzCBYQYDV  
R0RBIHBMIG  
+gh5hY2NvdW50cy1kZW1vLnNob3BpZnlibG91  
ZC5jb22CFGFjY291bnRzLnNob3B  
pZnkuY29tgh9jdXN0b21lcnMtZGVtby5zaG9w  
aWZ5Y2xvdWQuY29tgiZpZGVudGl0  
eS1tYWluLWNsdXN0ZXluc2hvcGlmeWNsb3V  
kLmNvbYlIaWRIbnRpdHktbWVyY2h  
hbnQuc2hvcGlmeWNsb3VkLmNvbYlIaWRIbnR  
pZnkuY29tgh9jdXN0b21lcnMtZGVtby5zaG9w  
aWZ5Y2xvdWQuY29tgiZpZGVudGl0eS1tYWlu  
LWNsdXN0ZXluc2hvcGlmeWNsb3VkLmNvbYlIa  
WRIbnRpdHktbWVyY2hbnQuc2hvcGlmeWNsb3V  
kLmNvbYlIaWRIbnRpdHktbWVyY2hbnQuc2hvc  
GlmeWNsb3VkLmNvbYlIaWRIbnRpdHktbWVyY2  
hbnQuc2hvcGlmeWNsb3VkLmNvbYlIaWRIbnR  
pdHktbWVyY2hbnQuc2hvcGlmeWNsb3VkLmNvb  
YlIaWRIbnRpdHktbWVyY2hbnQuc2hvcGlmeW  
Nsb3VkLmNvbYlIaWRIbnRpdHktbWVyY2hbnQ  
uc2hvcGlmeWNsb3VkLmNvbYlIaWRIbnRpdHk  
tbWVyY2hbnQuc2hvcGlmeWNsb3VkLmNvbYlI  
aWRIbnRpdHktbWVyY2hbnQuc2hvcGlmeWNsb3  
VkLmNvbYlIaWRIbnRpdHktbWVyY2hbnQuc2h  
vcGlmeWNsb3VkLmNvbYlIaWRIbnRpdHktbWV  
yY2hbnQuc2hvcGlmeWNsb3VkLmNvbYlIaWRI  
bnRpdHktbWVyY2hbnQuc2hvcGlmeWNsb3VkLm  
NvbYlIaWRIbnRpdHktbWVyY2hbnQuc2hvcGl  
meWNsb3VkLmNvbYlIaWRIbnRpdHktbWVyY2h  
bnQuc2hvcGlmeWNsb3VkLmNvbYlIaWRIbnR  
pdHktbWVyY2hbnQuc2hvcGlmeWNsb3VkLmNv  
bYlIaWRIbnRpdHktbWVyY2hbnQuc2hvcGlme  
WNsb3VkLmNvbYlIaWRIbnRpdHktbWVyY2hbn  
Quc2hvcGlmeWNsb3VkLmNvbYlIaWRIbnRpdH  
ktbWVyY2hbnQuc2hvcGlmeWNsb3VkLmNvbYlI  
aWRIbnRpdHktbWVyY2hbnQuc2hvcGlmeWNsb3  
VkLmNvbYlIaWRIbnRpdHktbWVyY2hbnQuc2h  
vcGlmeWNsb3VkLmNvbYlIaWRIbnRpdHktbWV  
yY2hbnQuc2hvcGlmeWNsb3VkLmNvbYlIaWRI  
bnRpdHktbWVyY2hbnQuc2hvcGlmeWNsb3VkLm  
NvbYlIaWRIbnRpdHktbWVyY2hbnQuc2hvcGl  
meWNsb3VkLmNvbYlIaWRIbnRpdHktbWVyY2h  
bnQuc2hvcGlmeWNsb3VkLmNvbYlIaWRIbnR  
pdHktbWVyY2hbnQuc2hvcGlmeWNsb3VkLmNv  
bYlIaWRIbnRpdHktbWVyY2hbnQuc2hvcGlme  
WNsb3VkLmNvbYlIaWRIbnRpdHktbWVyY2hbn  
Quc2hvcGlmeWNsb3VkLmNvbYlIaWRIbnRpdH  
ktbWVyY2hbnQuc2hvcGlmeWNsb3VkLmNvbYlI  
aWRIbnRpdHktbWVyY2hbnQuc2hvcGlmeWNsb3  
VkLmNvbYlIaWRIbnRpdHktbWVyY2hbnQuc2h  
vcGlmeWNsb3VkLmNvbYlIaWRIbnRpdHktbWV  
yY2hbnQuc2hvcGlmeWNsb3VkLmNvbYlIaWRI  
bnRpdHktbWVyY2hbnQuc2hvcGlmeWNsb3VkLm  
NvbYlIaWRIbnRpdHktbWVyY2hbnQuc2hvcGl  
meWNsb3VkLmNvbYlIaWRIbnRpdHktbWVyY2h  
bnQuc2hvcGlmeWNsb3VkLmNvbYlIaWRIbnR  
pdHktbWVyY2hbnQuc2hvcGlmeWNsb3VkLmNv  
bYlIaWRIbnRpdHktbWVyY2hbnQuc2hvcGlme  
WNsb3VkLmNvbYlIaWRIbnRpdHktbWVyY2hbn  
Quc2hvcGlmeWNsb3VkLmNvbYlIaWRIbnRpdH  
ktbWVyY2hbnQuc2hvcGlmeWNsb3VkLmNvbYlI  
aWRIbnRpdHktbWVyY2hbnQuc2hvcGlmeWNsb3  
VkLmNvbYlIaWRIbnRpdHktbWVyY2hbnQuc2h  
vcGlmeWNsb3VkLmNvbYlIaWRIbnRpdHktbWV  
yY2hbnQuc2hvcGlmeWNsb3VkLmNvbYlIaWRI  
bnRpdHktbWVyY2hbnQuc2hvcGlmeWNsb3VkLm  
NvbYlIaWRIbnRpdHktbWVyY2hbnQuc2hvcGl  
meWNsb3VkLmNvbYlIaWRIbnRpdHktbWVyY2h  
bnQuc2hvcGlmeWNsb3VkLmNvbYlIaWRIbnR  
pdHktbWVyY2hbnQuc2hvcGlmeWNsb3VkLmNv  
bYlIaWRIbnRpdHktbWVyY2hbnQuc2hvcGlme  
WNsb3VkLmNvbYlIaWRIbnRpdHktbWVyY2hbn  
Quc2hvcGlmeWNsb3VkLmNvbYlIaWRIbnRpdH  
ktbWVyY2hbnQuc2hvcGlmeWNsb3VkLmNvbYlI  
aWRIbnRpdHktbWVyY2hbnQuc2hvcGlmeWNsb3  
VkLmNvbYlIaWRIbnRpdHktbWVyY2hbnQuc2h  
vcGlmeWNsb3VkLmNvbYlIaWRIbnRpdHktbWV  
yY2hbnQuc2hvcGlmeWNsb3VkLmNvbYlIaWRI  
bnRpdHktbWVyY2hbnQuc2hvcGlmeWNsb3VkLm  
NvbYlIaWRIbnRpdHktbWVyY2hbnQuc2hvcGl  
meWNsb3VkLmNvbYlIaWRIbnRpdHktbWVyY2h  
bnQuc2hvcGlmeWNsb3VkLmNvbYlIaWRIbnR  
pdHktbWVyY2hbnQuc2hvcGlmeWNsb3VkLmNv  
bYlIaWRIbnRpdHktbWVyY2hbnQuc2hvcGlme  
WNsb3VkLmNvbYlIaWRIbnRpdHktbWVyY2hbn  
Quc2hvcGlmeWNsb3VkLmNvbYlIaWRIbnRpdH  
ktbWVyY2hbnQuc2hvcGlmeWNsb3VkLmNvbYlI  
aWRIbnRpdHktbWVyY2hbnQuc2hvcGlmeWNsb3  
VkLmNvbYlIaWRIbnRpdHktbWVyY2hbnQuc2h  
vcGlmeWNsb3VkLmNvbYlIaWRIbnRpdHktbWV  
yY2hbnQuc2hvcGlmeWNsb3VkLmNvbYlIaWRI  
bnRpdHktbWVyY2hbnQuc2hvcGlmeWNsb3VkLm  
NvbYlIaWRIbnRpdHktbWVyY2hbnQuc2hvcGl  
meWNsb3VkLmNvbYlIaWRIbnRpdHktbWVyY2h  
bnQuc2hvcGlmeWNsb3VkLmNvbYlIaWRIbnR  
pdHktbWVyY2hbnQuc2hvcGlmeWNsb3VkLmNv  
bYlIaWRIbnRpdHktbWVyY2hbnQuc2hvcGlme  
WNsb3VkLmNvbYlIaWRIbnRpdHktbWVyY2hbn  
Quc2hvcGlmeWNsb3VkLmNvbYlIaWRIbnRpdH  
ktbWVyY2hbnQuc2hvcGlmeWNsb3VkLmNvbYlI  
aWRIbnRpdHktbWVyY2hbnQuc2hvcGlmeWNsb3  
VkLmNvbYlIaWRIbnRpdHktbWVyY2hbnQuc2h  
vcGlmeWNsb3VkLmNvbYlIaWRIbnRpdHktbWV  
yY2hbnQuc2hvcGlmeWNsb3VkLmNvbYlIaWRI  
bnRpdHktbWVyY2hbnQuc2hvcGlmeWNsb3VkLm  
NvbYlIaWRIbnRpdHktbWVyY2hbnQuc2hvcGl  
meWNsb3VkLmNvbYlIaWRIbnRpdHktbWVyY2h  
bnQuc2hvcGlmeWNsb3VkLmNvbYlIaWRIbnR  
pdHktbWVyY2hbnQuc2hvcGlmeWNsb3VkLmNv  
bYlIaWRIbnRpdHktbWVyY2hbnQuc2hvcGlme  
WNsb3VkLmNvbYlIaWRIbnRpdHktbWVyY2hbn  
Quc2hvcGlmeWNsb3VkLmNvbYlIaWRIbnRpd

iFVoJMLAyp5SiXkrxQ54CX8uapdomX4i8NcAAAF7StEgegAABAMA<br>RzBFaiBQFP/  
Jq9s0RX2lddzPRWorKmK2vHb0gBjdeR4DafmJfglhAOGXW8pNxh  
+q<br>AgkEhILco7khwJUOAcDqcKCI9HS9FOJ8MA0GCSqGSib3DQEBCwUAA4IBAQ  
BcKanD<br>DWicj0QRpqqbMyHi1uwU0vJoOuCXAB5TMevD1OGhI2HGfV8blgxWlfOU  
LxGoEd/k<br>SA0t9X94AFhu+FhBQnrssUH/D  
+xQFKSbaZcQIB5j1nEFNdtpokxW5JyhjS5MXiYO<br>dWfgQcEhx3HlnYVT  
+Z1CpNslgCSOaoOawx+b1y1/Kj5ZIEld2rZBIOxh+/ifPP9h<br>Z8Y/kug  
+O2Vu7xfPWn4RtVeG39FpK3FcbOxkTA09K1CHtPQf+KZ/  
Ou5BcLn8BJi5<br>MVXdjQTeKuzilyV7QM+dAxHn3/  
reMXmlGengm7nj9K2P3I7Z70dX1sEwUXQR8Wr<br>/+IfVfwGqqjAuskK<br>-----END  
CERTIFICATE-----<br>-----BEGIN CERTIFICATE-----  
<br>MIIFfJCCAv6gAwIBAgIRAJErCErPDBinU/  
bWLiWnX1owDQYJKoZIhvcNAQELBQAw<br>TzELMAkGA1UEBhMCVVMxKTAnBgN  
VBAoTIEludGVybmV0IFNIY3VyaXR5IFJlc2Vh<br>cmNoIEdyb3VwMRUwEwYDVQQD  
EwxJU1JHIFJvb3QgWDEwHhcNMjAwOTA0MDAwMDAw<br>WhcNMjUwOTE1MTYw  
MDAwWjAyMQswCQYDVQQGEwJVUzEWMBQGA1UEChMNTGV0J3Mg<br>RW5jcnl  
wdDELMAkGA1UEAxMCUjMwggEiMA0GCSqGSib3DQEBAQUAA4IBDwAwggEK<br>  
AoIBAQC7AhUozPaglNMPEuyNVZLD+ILxmaZ6QoinXSagtSu5xUyxr45r  
+XXlo9cP<br>R5QUVTVXjJ6oojkZ9YI8QqlObvU7wy7bjcCwXPNZOOftz2nwWgsbvsCU  
JCWH+jdx<br>sxPnHKzhm+/  
b5DtFUKWWqcFTzjTIUu61ru2P3mBw4qVUq7ZtDpelQDRrK9O8Zutm<br>NHZ6a4uPVy  
mZ+DAXXbpyb/uBxa3Shlg9F8fnCbxvK/eG3MHacV3URuPMrSXBiLxg<br>Z3Vms/  
EY96Jc5IP/Ooi2R6X/ExjqmAl3P51T+c8B5fWmcBcUr2Ok/5mzk53cU6cG<br>/  
kiFHaFpriV1uxPMUgP17VGhi9sVAgMBAAGjggEIMIIBBDAOBgNVHQ8BAf8EBAMC<br>  
>AYYwHQYDVR0IBBYwFAYIKwYBBQUHAWIGCCsGAQUFBwMBMBIGA1UdEwEB/  
wQIMAYB<br>Af8CAQAwHQYDVR0OBBYEFBQusxe3WFbLrIAJQOYfr52LFMLGMB8  
GA1UdIwQYMBaA<br>FHm0WeZ7tuXkAXOACijlGij26ZtuMDIGCCsGAQUFBwEBBCY  
wJDAiBggrBgEFBQcw<br>AoYWaHR0cDovL3gxLmkubGVuY3lub3JnLzAnBgNVHR8EI  
DAeMBygGqAYhhZodHRw<br>Oi8veDEuYy5sZW5jci5vcmcvMCIGA1UdIAQbMBkwCA  
YGZ4EMAQIBMA0GCysGAQQB<br>gt8TAQEBA0GCSqGSib3DQEBCwUAA4ICAQ  
CFyk5HPqP3hUSFvNVneLKYY611TR6W<br>PTNlclQtdgaDqw+34IL9fzLdwALduO/  
ZelN7klJ+m74uyA+eitRY8kc607TkC53wl<br>ikfmZW4/RvTZ8M6UK  
+5UzhK8jCdLuMGYL6KvzXGRSgi3yLgjewQtCPkIVz6D2QQz<br>CkcheAmCJ8MqyJu  
5zlzyZMjAvnnAT45tRAxekrsu94sQ4egdRCnbWSDtY7kh  
+Blm<br>IJNXoB1IBMEKlq4QDUOXoRgffuDghje1WrG9ML+Hbisq/  
yFOGwXD9RiX8F6sw6W4<br>avAuvDsue5L3sz85K+EC4Y/wFVDNvZo4TYXao6Z0f  
+lQKc0t8DQYzk1OXVu8rp2<br>yJMC6alLbFODALZvYH7n7do1AZIs4I9d1P4jnkDrQox  
B3UqQ9hVl3LEKQ73xF1O<br>yK5GhDDX8oVfGKF5u  
+declsH4YaTw7mP3GFxJSqv3+0IUfJoi5Lc5da149p90lds<br>hCExroL1+7mrylkXPeF  
M5TgO9r0rvZaBFOvV2z0gp35Z0+L4WPIbuEjN/lxPFin+<br>HIUjr8gRsl3qfJOQFy/  
9rKIJR0Y/8Omwt/  
8oTWgy1mdeHmnmjk7j1nYsvC9JSQ6Zv<br>MldITTKB3zhThV1+XWYp6rjd5JW1zbVW  
EkLNxE7GJThEUG3szgBVGP7pSWTUTsqX<br>nLRbwHOoq7hHwg==<br>-----END  
CERTIFICATE-----<br>-----BEGIN CERTIFICATE-----  
<br>MIIFYDCCBEigAwIBAgIQQAF3ITfU6UK47naqPGQKtzANBgkqhkiG9w0BAQsFAD  
A/

<br>MSQwlgYDVQQKEExtEaWdpdGFsIFNpZ25hdHVyZSBUcnVzdCBDby4xFzAVBgNV  
BAMT<br>DkRtVCBSb290IENBIFgzMB4XDTIxMDEyMDE5MTQwM1oXDTI0MDkzMD  
E4MTQwM1ow<br>TzELMAkGA1UEBhMCVVMxKTAnBgNVBAoTIEludGVybmV0IFNIY  
3VyaXR5IFJlc2Vh<br>cmNoIEdyb3VwMRUwEwYDVQQDEwxJU1JHIFJvb3QgWDEwg  
gliMA0GCSqGSIb3DQEB<br>AQUAA4ICDwAwggIKAoICAQCt6CRz9BQ385ueK1coHI  
e+3LffOJCMbjzmV6B493XC<br>ov71am72AE8o295ohmxEk7axY/0UEmu/  
H9LqMZshftEzPLPl9d1537O4/xLxIZpL<br>wYqGcWIKZmZsj348cL  
+tKSIG8+TA5oCu4kuPt5l+IAOf00eXfJIII1PoOK5PCm  
+D<br>LtFJV4yAdLbaL9A4jXsDcCEbdfwPPqPrt3aY6vrFk/CjhFLfs8L6P  
+1dy70sntK<br>4EwSJQxwjQMpoOFTJOwT2e4ZvxCzSow/  
iaNhUd6shweU9GNx7C7ib1uYgeGJXDR5<br>bHbvO5BieebbpJovJsXQEOEO3tkQjhb  
7t/eo98flAgeYjzYllefIn5YNNnWe+w5y<br>sR2bvAP5SQXYgd0FtCrWQemsAXaVCg/  
Y39W9Eh81LygXbNKYwagJZHduRze6zqxZ<br>Xmidf3LWicUGQSk  
+WT7dJvUkyRGnWqNMQB9GoZm1pzpRboY7nn1ypxIFeFntPIF4<br>FQsDj43QLwWy  
PntKHetzBRL8xurgUBN8Q5N0s8p0544fAQjQMNRbcTa0B7rBMDBC<br>SLeCO5imfW  
CKoqMpgsy6vYMEG6KDA0Gh1gXxG8K28Kh8hjtGqEgqiNx2mna/  
H2ql<br>PRmP6zjzZN7IKw0KKP/32+IVQtQi0Cdd4Xn+GOdwik1O5tmLOsbdJ1Fu/  
7xk9TND<br>TwIDAQABo4IBRjCCAUIwDwYDVR0TAQH/BAUwAwEB/  
zAOBgNVHQ8BAf8EBAMCAQYw<br>SwYIKwYBBQUHAQEEPzA9MDsGCCsGAQUF  
BzAChi9odHRwOi8vYXBwcy5pZGVudHJ1<br>c3QuY29tL3Jvb3RzL2RzdHJvb3RjYXgz  
LnA3YzAfBgNVHSMEGDAWgBTEp7Gkeyxx<br>  
+tvhS5B1/8QVYIWJEDBUBgNVHSAETTLMAgGBmeBDAECATA/  
BgserBgEEAYLfEwEB<br>ATAwMC4GCCsGAQUFBwIBFiJodHRwOi8vY3BzLnJvb3Qte  
DEubGV0c2VuY3J5cHQw<br>b3JnMDwGA1UdHwQ1MDMwMaAvoC2GK2h0dHA6Ly9  
jcmwuaWRlbnRydXN0LmNvbS9E<br>U1RST09UQ0FYM0NSTC5jcmwwHQYDVR0OB  
BYEFHm0WeZ7tuXkAXOACIjGJl26Ztu<br>MA0GCSqGSIb3DQEBChwUAA4IBAQAkCw  
Bslm7/DILQrt2M51oGrS+o44+/  
yQoDFVDC<br>5WxCu2+b9LRPwkSICHXM6webFGJueN7sJ7o5XPWioW5WIHAQU7  
G75K/  
QosMrAdSW<br>9MUgNTP52GE24HGNtLi1qoJFicDyqSMo59ahy2cl2qBDLKobkx/  
J3vWraV0T9VuG<br>WCLKTVXkcGdtwlfFRjIBz4pYg1htmf5X6DYO8A4jqv2Il9DjXA6U  
SbW1FzXSLr9O<br>he8Y4IWS6wY7bCkjCWDcRQJMEhg76fsO3txE  
+FiYruq9RUWhiF1myv4Q6W  
+CyBFC<br>Dfvp7OOGAN6dEOM4+qR9sdjoSYKEBpsr6GtPAQw4dy753ec5<br>-----  
END CERTIFICATE-----<br>-----BEGIN CERTIFICATE-----  
<br>MIIDSjCCAjKgAwIBAgIQRK  
+wgNajJ7qJMDmGLvhAazANBgkqhkiG9w0BAQUFADA/  
<br>MSQwlgYDVQQKEExtEaWdpdGFsIFNpZ25hdHVyZSBUcnVzdCBDby4xFzAVBgNV  
BAMT<br>DkRtVCBSb290IENBIFgzMB4XDTAwMDkzMDIxMTIxOVVoXDTIxMDkzMDE  
0MDExNVow<br>PzEkMCIGA1UEChMbRGlnaXRhbCBTaWduYXR1cmUgVHJ1c3QgQ  
28uMRcwFQYDVQQD<br>Ew5EU1QgUm9vdCBDQSBYmzCCASlwdQYJKoZIhvcNA  
QEBBQADggEPADCCAQoCggEB<br>AN  
+v6ZdQCINXtMxiZfaQguzH0yxrMMpb7NnDfcdAwRgUi+DoM3ZJKuM/  
IUmTrE4O<br>rz5ly2Xu/  
NMhD2XSKtkyj4zl93ewEnu1lcCJo6m67XMuegwGMOifooUMM0RoOEQ<br>OLI5CjH9  
UL2AZd

+3UWODyOKIYepLYYHsUmu5ouJLGiifSKOeDNoJjj4XLh7dIN9b<br>xiqKqy69cK3FCx  
olkHRyxXtqqzTWMIn/5WgTe1QLyNau7Fqckh49ZLOMxt+/  
yUFw<br>7BZy1SbsOFU5Q9D8/  
RhcQPGX69Wam40dutolucbY38EVAjqr2m7xPi71XAicPNaD<br>aeQQmxkqtilX4+U9m  
5/wAl0CAwEAAaNCMEAwDwYDVR0TAQH/BAUwAwEB/  
zAOBgNV<br>HQ8BAf8EBAMCAQYwHQYDVR0OBBYEFMSnsaR7LHH62+FLkHX/  
xBVghYkQMA0GCSqG<br>Slb3DQEBBQUAA4IBAQCjGiybFwBcqR7uKGY3Or  
+Dxz9LwwmglSBd49IZRNI+DT69<br>ikugdB/  
OEIKcdBodfpga3csTS7MgROSR6cz8faXbauX  
+5v3gTt23ADq1cEmv8uXr<br>AvHRAosZy5Q6XkjEGB5YGV8eAlrwDPGxrancWYaLbu  
mR9YbK  
+rlmM6pZW87ipxZz<br>R8srzJmwN0jP41ZL9c8PDHlyh8bwRLtTcm1D9SZlmlJnt1ir/  
md2cXjbDaJWFBM5<br>JDGFoqgCWjBH4d1QB7wCCZAA62RjYJsWvljJEubSfZGL  
+T0yjWW06XyxV3bqxbYo<br>Ob8VZRzI9neWagqNdwvYkQsEjgfbKbYK7p2CNTUQ<b  
r>-----END CERTIFICATE-----<br>/span>/p><p class="debugging-content"><span  
class="debugging-title">Certificate Transparency: </span><span class="debugging-  
content"><br><br>SCT Cloudflare 'Nimbus2021' Log (Embedded in certificate,  
Verified)<br><br>SCT Google 'Xenon2021' log (Embedded in certificate, Verified)</  
span></p></div>

</div>

</div>

<div id="recurrent-error-message" class="hidden">

Warnings may be common while websites update their security. This should  
improve soon.

</div>

<div id="extended-reporting-opt-in" class="hidden">

<label>

<div class="checkboxes">

<input type="checkbox" id="opt-in-checkbox">

<span class="checkbox"></span>

</div>

<span id="opt-in-label">Help improve security on the web for everyone by  
sending <a href="#" id="whitepaper-link">URLs of some pages you visit, limited system  
information, and some page content</a> to Google. <a id="privacy-link"  
href="#">Privacy policy</a></span>

</label>

</div>

<div id="enhanced-protection-message" class="hidden">

<label>

<div class="icon"></div>

<div>To get Chrome's highest level of security, <a href="#" id="enhanced-  
protection-link">turn on enhanced protection</a></div>

</label>

</div>

</div>

<div class="nav-wrapper">

```

    <button id="primary-button">Reload</button>
    <button id="proceed-button" class="secondary-button small-link hidden"></button>
    <button id="details-button" class="secondary-button small-link" aria-
expanded="true">Hide advanced</button>
  </div>
  <div id="details" class="">
    <p>accounts.shopify.com normally uses encryption to protect your information.
When Chrome tried to connect to accounts.shopify.com this time, the website sent back
unusual and incorrect credentials. This may happen when an attacker is trying to
pretend to be accounts.shopify.com, or a Wi-Fi sign-in screen has interrupted the
connection. Your information is still secure because Chrome stopped the connection
before any data was exchanged.</p>
    <p id="final-paragraph">You cannot visit accounts.shopify.com right now because
the website uses HSTS. Network errors and attacks are usually temporary, so this page
will probably work later.</p>
  </div>
</div>

```

```

<style>/* Copyright 2014 The Chromium Authors. All rights reserved.

```

```

* Use of this source code is governed by a BSD-style license that can be
* found in the LICENSE file. */

```

```

/* This file is dynamically processed by a C++ data source handler to fill in
* some per-platform/locale styles that dramatically alter the page. This is
* done to reduce flicker, as JS may not run before the page is rendered.
*

```

```

* There are two ways to include this stylesheet:
* 1. via its chrome://resources/ URL in HTML, i.e.:
*

```

```

* <link rel="stylesheet" href="chrome://resources/css/text_defaults.css">
*

```

```

* 2. via the webui::AppendWebUICSSTextDefaults() method to directly append it
* to an HTML string.
* Otherwise its placeholders won't be expanded. */

```

```

body {
  font-family: system-ui, sans-serif;
  font-size: 75%;
}

```

```

button {
  font-family: system-ui, sans-serif;
}

```

```

</style><script>// Copyright (c) 2012 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be

```

```
// found in the LICENSE file.
```

```
/**
 * @fileoverview This file defines a singleton which provides access to all data
 * that is available as soon as the page's resources are loaded (before DOM
 * content has finished loading). This data includes both localized strings and
 * any data that is important to have ready from a very early stage (e.g. things
 * that must be displayed right away).
 *
 * Note that loadTimeData is not guaranteed to be consistent between page
 * refreshes (https://crbug.com/740629) and should not contain values that might
 * change if the page is re-opened later.
 */
```

```
/** @type {!LoadTimeData} */
// eslint-disable-next-line no-var
var loadTimeData;
```

```
class LoadTimeData {
  constructor() {
    /** @type {?Object} */
    this.data_ = null;
  }
}
```

```
/**
 * Sets the backing object.
 *
 * Note that there is no getter for |data_| to discourage abuse of the form:
 *
 *   var value = loadTimeData.data()['key'];
 *
 * @param {Object} value The de-serialized page data.
 */
```

```
set data(value) {
  expect(!this.data_, 'Re-setting data.');
```

```
  this.data_ = value;
}

/**
 * @param {string} id An ID of a value that might exist.
 * @return {boolean} True if |idl| is a key in the dictionary.
 */
```

```
valueExists(id) {
  return id in this.data_;
}
```

```

/**
 * Fetches a value, expecting that it exists.
 * @param {string} id The key that identifies the desired value.
 * @return {*} The corresponding value.
 */
getValue(id) {
  expect(this.data_, 'No data. Did you remember to include strings.js?');
  const value = this.data_[id];
  expect(typeof value !== 'undefined', 'Could not find value for ' + id);
  return value;
}

/**
 * As above, but also makes sure that the value is a string.
 * @param {string} id The key that identifies the desired string.
 * @return {string} The corresponding string value.
 */
getString(id) {
  const value = this.getValue(id);
  expectIsType(id, value, 'string');
  return /** @type {string} */ (value);
}

/**
 * Returns a formatted localized string where $1 to $9 are replaced by the
 * second to the tenth argument.
 * @param {string} id The ID of the string we want.
 * @param {...(string|number)} var_args The extra values to include in the
 *   formatted output.
 * @return {string} The formatted string.
 */
getStringF(id, var_args) {
  const value = this.getString(id);
  if (!value) {
    return '';
  }

  const args = Array.prototype.slice.call(arguments);
  args[0] = value;
  return this.substituteString.apply(this, args);
}

/**
 * Returns a formatted localized string where $1 to $9 are replaced by the
 * second to the tenth argument. Any standalone $ signs must be escaped as
 * $$$.

```



```

* @param {string} label The label to substitute through.
*   This is not an resource ID.
* @param {...(string|number)} var_args The extra values to include in the
*   formatted output.
* @return {string} The formatted string.
*/
substituteString(label, var_args) {
  const varArgs = arguments;
  return label.replace(/\$(.|\n)/g, function(m) {
    expect(m.match(/\$[1-9]/), 'Unescaped $ found in localized string. ');
    return m === '$$' ? '$' : varArgs[m[1]];
  });
}

/**
* Returns a formatted string where $1 to $9 are replaced by the second to
* tenth argument, split apart into a list of pieces describing how the
* substitution was performed. Any standalone $ signs must be escaped as $$
* @param {string} label A localized string to substitute through.
*   This is not an resource ID.
* @param {...(string|number)} var_args The extra values to include in the
*   formatted output.
* @return {Array<!{value: string, arg: (null|string)}>} The formatted
*   string pieces.
*/
getSubstitutedStringPieces(label, var_args) {
  const varArgs = arguments;
  // Split the string by separately matching all occurrences of $1-9 and of
  // non $1-9 pieces.
  const pieces = (label.match(/(\$(1-9)|((\^$\)|\$(1-9)|$))+/g) ||
    []).map(function(p) {
    // Pieces that are not $1-9 should be returned after replacing $$
    // with $.
    if (!p.match(/^\$(1-9)$/)) {
      expect(
        (p.match(/\$/g) || []).length % 2 === 0,
        'Unescaped $ found in localized string. ');
      return {value: p.replace(/\$\$/g, '$'), arg: null};
    }

    // Otherwise, return the substitution value.
    return {value: varArgs[p[1]], arg: p};
  });

  return pieces;
}

```

```

/**
 * As above, but also makes sure that the value is a boolean.
 * @param {string} id The key that identifies the desired boolean.
 * @return {boolean} The corresponding boolean value.
 */
getBoolean(id) {
  const value = this.getValue(id);
  expectIsType(id, value, 'boolean');
  return /** @type {boolean} */ (value);
}

/**
 * As above, but also makes sure that the value is an integer.
 * @param {string} id The key that identifies the desired number.
 * @return {number} The corresponding number value.
 */
getInteger(id) {
  const value = this.getValue(id);
  expectIsType(id, value, 'number');
  expect(value === Math.floor(value), 'Number isn\'t integer: ' + value);
  return /** @type {number} */ (value);
}

/**
 * Override values in loadTimeData with the values found in replacements.
 * @param {Object} replacements The dictionary object of keys to replace.
 */
overrideValues(replacements) {
  expect(
    typeof replacements === 'object',
    'Replacements must be a dictionary object.');
  for (const key in replacements) {
    this.data_[key] = replacements[key];
  }
}

/**
 * Reset loadTimeData's data. Should only be used in tests.
 * @param {?Object} newData The data to restore to, when null restores to
 *   unset state.
 */
resetForTesting(newData = null) {
  this.data_ = newData;
}

```

```

/**
 * @return {boolean} Whether loadTimeData.data has been set.
 */
isInitialized() {
  return this.data_ !== null;
}
}

/**
 * Checks condition, throws error message if expectation fails.
 * @param {*} condition The condition to check for truthiness.
 * @param {string} message The message to display if the check fails.
 */
function expect(condition, message) {
  if (!condition) {
    throw new Error(
      'Unexpected condition on ' + document.location.href + ': ' + message);
  }
}

/**
 * Checks that the given value has the given type.
 * @param {string} id The id of the value (only used for error message).
 * @param {*} value The value to check the type on.
 * @param {string} type The type we expect lvalue to be.
 */
function expectIsType(id, value, type) {
  expect(
    typeof value === type, '[' + value + '] (' + id + ') is not a ' + type);
}

expect(!loadTimeData, 'should only include this file once');
loadTimeData = new LoadTimeData;

// Expose lloadTimeData directly on lwindowl, since within a JS module the
// scope is local and not all files have been updated to import the exported
// lloadTimeData explicitly.
window.loadTimeData = loadTimeData;

console.warn('crbug/1173575, non-JS module files deprecated.');
```

</script><script>loadTimeData.data = {"bad\_clock":false,"closeDetails":"Hide advanced","ct":"\n\nSCT Cloudflare 'Nimbus2021' Log (Embedded in certificate, Verified)\n\nSCT Google 'Xenon2021' log (Embedded in certificate, Verified)","currentDate":"Oct 5, 2021","displayEnhancedProtectionMessage":false,"displaycheckbox":false,"enhancedProtectionMessage":"To get Chrome's highest level of security, \u003Ca href=\"\#\" id=

\>turn on enhanced protection\u003C/a>","errorCode":"net::ERR\_CERT\_DATE\_INVALID","errorType":1,"expirationDate":"Nov 13, 2021","explanationParagraph":"accounts.shopify.com normally uses encryption to protect your information. When Chrome tried to connect to accounts.shopify.com this time, the website sent back unusual and incorrect credentials. This may happen when an attacker is trying to pretend to be accounts.shopify.com, or a Wi-Fi sign-in screen has interrupted the connection. Your information is still secure because Chrome stopped the connection before any data was exchanged.", "finalParagraph":"You cannot visit accounts.shopify.com right now because the website uses HSTS. Network errors and attacks are usually temporary, so this page will probably work later.", "fontfamily":"system-ui, sans-serif", "fontsize":"75%", "heading":"Your connection is not private", "hide\_primary\_button":false, "issuer":"R3", "language":"en", "openDetails":"Advanced", "optInLink":"Help improve security on the web for everyone by sending \u003Ca href=\"\" id=\"whitepaper-link\">URLs of some pages you visit, limited system information, and some page content\u003C/a> to Google. \u003Ca id=\"privacy-link\" href=\"\">Privacy policy\u003C/a>","overridable":false, "pem":"-----BEGIN CERTIFICATE-----\nMIIF2DCCBMCGAwIBAgISA3IP+wuRtzmomjRFbuWpyrsCMA0GCSqGSIb3DQEBCwUA\nMDIxGzAJBgNVBAYTAiVTMRYwFAYDVQQKEw1MZXQncyBFbmNyeXB0MQswCQYDVQQD\nEwJSMzAeFw0yMTA4MTUxNjE3MDNaFw0yMTE3MTUxNjE3MDFaMB8xHTAbBgNVBAMT\nFGFjY291bnRzLnNob3BpZnkuY29tMIIIBjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB\nCgKCAQEA\nmNO6kENQqqUM0WgozXecZGcPzrlyEbrU/N7mdStXBNjXN9WAEbLSTGrw\ngkyShj1dbDXVJudJ7YdQXJr/KJbm87u8a1HEZI4Dmj1FxBIA\n+Gs1RsFdQd0nEw\n5Zgi0Sojj6d87SEqpS2i0Vz6V+hTrbuy+XXP5+J8piNiRp2lkrZQkWKaifZaLyvH\nnw2S2OFtEJxH3SSoKfsA/ST4ZPRDFlppsXgmpFf7sU5u1E3XckNjxl220Q29/vtIs\nErLP9noTy1wdKzXJYbu4IT6nVOSs+TGLQBcBCfNQyaG19jMrL299G4QyNy2h6uWX\nMwPm/Opq3qk3Bnl5uZgyV+mx7EMe1QIDAQAB\no4IC+TCCAvUwDgYDVR0PAQH/BAQD\nAgWgMB0GA1UdJQQWMBQGCCsGAQUFBwMBBggrBgEFBQcDAjAMBgNVHRMBAf8EAjAA\nnMB0GA1UdDgQWBBR/TRz5xF8Ma/wcElldjg8fyl3L3zAfBgNVHSMEGDAWgBQULrMX\nnt1hWy65QCUDmH6+dixTCxjBVBggrBgEFBQcBAQRJMEcwIQYIKwYBBQUHMAGGF\nWh0ndHA6Ly9yMy5vLmxlbmNyLm9yZzAiBggrBgEFBQcwAoYWaHR0cDovL3lzLmkub\nGVu\nY3lub3JnLzCBYQYDVR0RBIHBMIG+gh5hY2NvdW50cy1kZW1vLnNob3BpZnlibG91\nnZC5jb22CFGFjY291bnRzLnNob3BpZnkuY29tgh9jdXN0b21lcnMtZGVtby5zaG9w\nnaWZ5Y2xvdWQuY29tgiZpZGVudGl0eS1tYWluLWNsdXN0ZXIuc2hvcGlmeWNsb3V\nkLnNmNvbYliaWRlbnRpdHktbWVyY2hhbnQuc2hvcGlmeWNsb3V\nkLnNmNvbYIZaWRlbnRpdHkuc2hvcGlmeWNsb3V\nkLnNmNvbTBMBgNVHSAERTBDMAgGBmeBDAECATA3BggrBgEE\nAYLfEwEBATAoMCYGCCsGAQUFBwIBFhpodHRwOi8vY3BzLmxldHNIbmNyeXB0Lm9y\nnZzCCAQMGCisGAQQB1nkCBAIEgfQEgEA7wB1AESUZS6w7s6vxEAH2Kj

+KMDa5oK+\n2MsxtT/TM5a1toGoAAABe0rRIFkAAAQDAEYwRAIgcU9G4Kf  
+YE3BwY4IHwrPfCyx\nHBfmiWZVYY3syP6PMtQCICbrsrVPy0eYoa  
+way7bJOxuRAbwT4Sfd4Nd14ITkrE\nAHYAft7y+l//  
iFVoJMLAyp5SiXkrxQ54CX8uapdomX4i8NcAAAF7StEgegAABAMA\nRzBFAiBQFP/  
Jq9s0RX2lddzPRWorKmK2vHb0gBjdeR4DafmJfglhAOGXW8pNxx+q  
\nAgkEhILco7khwJUOAcDqcKCI9HS9FOJ8MA0GCSqGSib3DQEBCwUAA4IBAQBcKa  
nD\nDWicj0QRpqqbMyHi1uwU0vJoOuCXAB5TMevD1OGhl2HGfV8blgxWlfOULxGoEd/  
k\nSA0t9X94AFhu+FhBQnrssUH/D+xQFKSbaZcQIB5j1nEFNdtpokxW5JyhjS5MXiYO  
\ndWfgQcEhx3HlnYVT+Z1CpNslgCSOaoOawx+b1y1/Kj5ZIEld2rZBIOxh+/ifPP9h\nZ8Y/  
kug+O2Vu7xfPWn4RtVeG39FpK3FcbOxkTA09K1CHtPQf+KZ/  
Ou5BcLn8BJi5\nMVXdjQTeKuzilyV7QM+dAxHn3/  
reMXmlGengm7nj9K2P3I7Z70dX1sEwUXQR8Wr\n/+lfVfwGqqjAuskK\n-----END  
CERTIFICATE-----\n-----BEGIN CERTIFICATE-----  
\nMIIFFjCCAv6gAwIBAgIRAJErCERPDBinU/bWLiWnX1owDQYJKoZIhvcNAQELBQAw  
\nTzELMAkGA1UEBhMCVVMxKTAnBgNVBAoTIEludGVybmV0IFNIY3VyaXR5IFJlc2Vh  
\nncmNoIEdyb3VwMRUwEwYDVQQDEwxJU1JHIFJvb3QgWDEwHhcNMjAwOTA0MDAwMDAw  
wMDAw  
\nWhcNMjUwOTE1MTYwMDAwWjAQMzswCQYDVQQGEwJVUzEWMBQGA1UEChMNTGVOJ3Mg  
\nRW5jcnlwdDELMAkGA1UEAxMCUjMwggEiMA0GCSqGSib3DQEBAQUAA4IBDwAwggEK\n\nAoIBAQC7AhUozPaglNMPEuyNVZLD+ILxmaZ6QoinXSaqtSu5xUyxr45r+XXlo9cP  
\nR5QUVTvXjJ6oojkZ9Yl8QqlObvU7wy7bjcCwXPNZOOftz2nwWgsbvsCUJCWH+jdx  
\nsxPnHKzhm+/b5DtFukWWqcFTzjTIUu61ru2P3mBw4qVUq7ZtDpelQDRrK9O8Zutm  
\nNHZ6a4uPVymZ+DAXXbpyb/uBxa3Shlg9F8fnCbvxK/eG3MHacV3URuPMrSXBilXg  
\nZ3Vms/EY96Jc5IP/Ooi2R6X/ExjqmAl3P51T+c8B5fWmcBcUr2Ok/5mzk53cU6cG\n\n/kiFHaFpriV1uxPMUgP17VGhi9sVAgMBAAGjggEIMIIBBDAOBgNVHQ8BAf8EBAMC  
\nAYYwHQYDVR0IBBYwFAYIKwYBBQUHAWIGCCsGAQUFBwMBMBIGA1UdEwEB/wQIMAYB  
\nAf8CAQAwHQYDVR0OBBYEFBQusxe3WFbLrIAJQOYfr52LFMLGMB8GA1UdIwQYMBAA  
\nFHm0WeZ7tuXkAXOACljlGlj26ZtuMDIGCCsGAQUFBwEBBCYwJDAlBggrBgEFBQc  
w  
\nAoYWaHR0cDovL3gxLmkubGVuY3lub3JnLzAnBgNVHR8EIDAeMBygGqAYhhZodHRw  
\nOi8veDEuYy5sZW5jci5vcmcvMCIGA1UdIAQbMBkwCAYGZ4EMAQIBMA0GCysGAQ  
QB  
\ngt8TAQEBMA0GCSqGSib3DQEBCwUAA4ICAQCfYk5HPqP3hUSFvNVneLKYy611T  
R6W\n\nPTNlclQtdq+34IL9fzLdwALduO/ZelN7klJ+m74uyA+eitRY8kc607TkC53wl  
\nikfmZW4/RvTZ8M6UK+5UzhK8jCdLuMGYL6KvzXGRSgi3yLgjewQtCPkIVz6D2QQz  
\nCkcheAmCJ8MqyJu5zlzyZMjAvnnAT45tRAXekrsu94sQ4egdRCnbWSDtY7kh+Blm  
\nIJNXoB1IBMEKlq4QDUOXoRgffuDghje1WrG9ML+Hbisq/  
yFOGwXD9RiX8F6sw6W4\n\nnavAuvDsue5L3sz85K+EC4Y/wFVDNvZo4TYXao6Z0f  
+IQKc0t8DQYzk1OXVu8rp2\n\nnyJMC6alLbBfODALZvYH7n7do1AZls4l9d1P4jnkDrQoxB3  
UqQ9hVI3LEKQ73xF1O\n\nnyK5GhDDX8oVfGKF5u  
+declsH4YaTw7mP3GFxJSqv3+0IUFJoi5Lc5da149p90lds

\nhCExroL1+7mrylkXPeFM5TgO9r0rvZaBFOvV2z0gp35Z0+L4WPIbuEjN/lxPFin+  
\nHIUjr8gRsl3qfJOQFy/9rKIJR0Y/8OmwT/8oTWgy1mdeHm mj7j1nYsvC9JSQ6Zv  
\nMldITTKB3zhThV1+XWYp6rjd5JW1zbVWEkLNxE7GJThEUG3szgBVGP7pSWTUTsq  
X\n\nLRbwHOoq7hHwg==\n-----END CERTIFICATE-----\n-----BEGIN CERTIFICATE-----  
\nMIIFYDCCBEigAwIBAgIQQAF3ITfU6UK47naqPGQKtzANBgkqhkiG9w0BAQsFADA/  
\nMSQWlgYDVQQKEExtEaWdpdGFsIFNpZ25hdHVyZSBUcnVzdCBDby4xFzAVBgNVBA  
MT  
\nDkRtVCBSb290IENBIFgzMB4XDTIxMDEyMDE5MTQwM1oXDTI0MDkzMDE4MTQw  
M1ow  
\nTzELMAkGA1UEBhMCVVMxKTAnBgNVBAoTIEludGVybmV0IFNIY3VyaXR5IFJlc2Vh  
\ncmNoIEdyb3VwMRUwEwYDVQQDEwxFU1JHIFJvb3QgWDEwggliMA0GCSqGSIb3D  
QEB\n\nAQUAA4ICDwAwggIKAoICAQCt6CRz9BQ385ueK1coHle  
+3LffOJCMbjzmV6B493XC\n\nnov71am72AE8o295ohmxEk7axY/0UEmu/  
H9LqMZshftEzPLpl9d1537O4/xLxIZpL\n\nnwYqGcWIKZmZsj348cL  
+tKSIG8+TA5oCu4kuPt5l+IAOf00eXfJlIl1PoOK5PCm+D  
\nLtFJV4yAdLbaL9A4jXsDcCEbdfwPPqPrt3aY6vrFk/CjhFLfs8L6P+1dy70sntK  
\n4EwSJQxwjqMpoOFTJOwT2e4ZvxCzSow/  
iaNhUd6shweU9GNx7C7ib1uYgeGJXDR5\n\nnbHbvO5BieebbpJovJsXQEOEO3tkQjhb7t/  
eo98flAgeYjzYllefI5YNNnWe+w5y\n\nnsR2bvAP5SQXYgd0FtCrWQemsAXaVCg/  
Y39W9Eh81LygXbNKYwagJZHduRze6zqxZ\n\nXmidf3LWicUGQSk  
+WT7dJvUkyRGnWqNMQB9GoZm1pzpRboY7nn1ypxIFeFntPIF4\n\nnFQsDj43QLwWyPn  
tKHEtzBRL8xurgUBN8Q5N0s8p0544fAQjQMNRbcTa0B7rBMDbC  
\nSLeCO5imfWCKoqMpgsy6vYMEG6KDA0Gh1gXxG8K28Kh8htGqEgqiNx2mna/H2ql  
\nPRmP6zjzZN7IKw0KKP/32+IVQtQi0Cdd4Xn+GOdwiK1O5tmLOsbdJ1Fu/7xk9TND  
\nTwIDAQABo4IBRjCCAUIlwDwYDVR0TAQH/BAUwAwEB/  
zAOBgNVHQ8BAf8EBAMCAQYw  
\nSwYIKwYBBQUHAQEEPzA9MDsGCCsGAQUFBzACChi9odHRwOi8vYXBwcy5pZGVu  
dHJ1\n\nc3QuY29tL3Jvb3RzL2RzdHJvb3RjYXgzLnA3YzAfBgNVHSMEGDAWgBTEp7G  
keyxx\n\ntvhS5B1/8QVYIWJEDBUBgNVHSAETTLMAgGBmeBDAECATA/  
BgSrBgEEAYLfEwEB  
\nATAwMC4GCCsGAQUFBwIBFiJodHRwOi8vY3BzLnJvb3QteDEubGV0c2VuY3J5cHQ  
u  
\nb3JnMDwGA1UdHwQ1MDMwMaAvoC2GK2h0dHA6Ly9jcmwuaWRlbnRydXN0LmNv  
bS9E  
\nU1RST09UQ0FYM0NSTC5jcmwwHQYDVR0OBBYEFHm0WeZ7tuXkAXOAClJlGj26  
Ztu\n\nMA0GCSqGSIb3DQEBCwUAA4IBAQAkCwBslm7/DILQrt2M51oGrS+o44+/  
yQoDFVDC  
\n5WxCu2+b9LRPwkSiCHXM6webFGJueN7sJ7o5XPWioW5WIHAQU7G75K/  
QosMrAdSW\n\nn9MUgNTP52GE24HGNtLi1qoJfIcDyqSMo59ahy2cl2qBDLKobkx/  
J3vWraV0T9VuG  
\nWCLKTVXkcGdtwlfFRjlBz4pYg1htmf5X6DYO8A4jqv2lI9DjXA6USbW1FzXSLr9O  
\nhe8Y4IWS6wY7bCkjCWDcRQJMEhg76fsO3txE+FiYruq9RUWhiF1myv4Q6W  
+CyBFC\n\nnDfvp7OOGAN6dEOM4+qR9sdjoSYKEBpsr6GtPAQw4dy753ec5\n\n-----END  
CERTIFICATE-----\n-----BEGIN CERTIFICATE-----\n\nMIIDSjCCAjKgAwIBAgIQRK  
+wgNajJ7qJMDmGLvhAazANBgkqhkiG9w0BAQUFADA/  
\nMSQWlgYDVQQKEExtEaWdpdGFsIFNpZ25hdHVyZSBUcnVzdCBDby4xFzAVBgNVBA

MT  
\\nDkRTVCBSb290IENBIFgzMB4XDTAwMDkzMDIxMTIxOVVoXDTIxMDkzMDE0MDExN  
Vow  
\\nPzEkMCIGA1UEChMbRGlnaXRhbCBTaWduYXR1cmUgVHJ1c3QgQ28uMRcwFQYD  
VQQD  
\\nEw5EU1QgUm9vdCBDQSBYmzCCASlwDQYJKoZIhvcNAQEBBQADggEPADCCAAQ  
oCggEB\\nAN+v6ZdQCINXtMxiZfaQguzH0yxrMMpb7NnDfcdAwRgUi+DoM3ZJKuM/  
lUmTrE4O\\nrz5ly2Xu/  
NMhD2XSKtkyj4zl93ewEnu1lcCJo6m67XMuegwGMOifooUMM0RoOEQ  
\\nOLI5CjH9UL2AZd+3UWODyOKIYepLYYHsUmu5ouJLGiifSKOeDNoJjj4XLh7dIN9b  
\\nxiqKqy69cK3FCxolkHRyxXtqqzTWMI\\n/5WgTe1QLyNau7Fqckh49ZLOMxt+/yUFW  
\\n7BZy1SbsOFU5Q9D8/RhcQPGX69Wam40dutolucbY38EVAjqr2m7xPi71XAicPNaD  
\\naeQQmxkqtiIX4+U9m5/wAl0CAwEAAANCMEEAwDwYDVR0TAQH/BAUwAwEB/  
zAOBgNV\\nHQ8BAf8EBAMCAQYwHQYDVR0OBBYEFMSnsaR7LHH62+FLkHX/  
xBVghYkQMA0GCSqG\\nSIlb3DQEBBQUAA4IBAQCjGiybFwBcqR7uKGY3Or  
+Dxz9LwwmgISBd49IZRNI+DT69\\nikugdB/  
OEIKcdBodfpga3csTS7MgROSR6cz8faXbauX+5v3gTt23ADq1cEmv8uXr  
\\nAvHRAosZy5Q6XkjEGB5YGV8eAlrwDPGxrancWYaLbumR9YbK+rlmM6pZW87ipxZz  
\\nR8srzJmwN0jP41ZL9c8PDHlyh8bwRLtTcm1D9SZlmlJnt1ir/  
md2cXjbDaJWFBM5\\nJDGFoqgCWjBH4d1QB7wCCZAA62RjYJsWvljJEubSfZGL  
+T0yjWW06XyxV3bqxbYo\\nOb8VZRzI9neWagqNdwvYkQsEjgfbKbYK7p2CNTUQ\\n-----  
END CERTIFICATE-----\\n","primaryButtonText":"Reload","primaryParagraph":"Attackers  
might be trying to steal your information from \\u003Cstrong>accounts.shopify.com  
\\u003C/strong> (for example, passwords, messages, or credit cards). \\u003Ca href=\\\"#  
\\\" id=\\\"learn-more-link\\\">Learn more\\u003C/a>","recurrentErrorParagraph":"Warnings  
may be common while websites update their security. This should improve  
soon.","show\_recurrent\_error\_paragraph":false,"subject":"accounts.shopify.com","tabTitl  
e":"Privacy error","textdirection":"ltr","type":"SSL"};</script></body></html>

-----

```
<head>
  <meta charset="utf-8">
  <meta name="color-scheme" content="light dark">
  <meta name="theme-color" content="#fff">
  <meta name="viewport" content="initial-scale=1, minimum-scale=1, width=device-
width">
  <meta http-equiv="Content-Security-Policy" content="require-trusted-types-for 'script';
trusted-types;">
  <title>Privacy error</title>
  <style>/* Copyright 2017 The Chromium Authors. All rights reserved.
* Use of this source code is governed by a BSD-style license that can be
* found in the LICENSE file. */
```

```
a {
  color: var(--link-color);
}
```

```
body {
  --background-color: #fff;
  --error-code-color: var(--google-gray-700);
  --google-blue-100: rgb(210, 227, 252);
  --google-blue-300: rgb(138, 180, 248);
  --google-blue-600: rgb(26, 115, 232);
  --google-blue-700: rgb(25, 103, 210);
  --google-gray-100: rgb(241, 243, 244);
  --google-gray-300: rgb(218, 220, 224);
  --google-gray-500: rgb(154, 160, 166);
  --google-gray-50: rgb(248, 249, 250);
  --google-gray-600: rgb(128, 134, 139);
  --google-gray-700: rgb(95, 99, 104);
  --google-gray-800: rgb(60, 64, 67);
  --google-gray-900: rgb(32, 33, 36);
  --heading-color: var(--google-gray-900);
  --link-color: rgb(88, 88, 88);
  --popup-container-background-color: rgba(0,0,0,.65);
  --primary-button-fill-color-active: var(--google-blue-700);
  --primary-button-fill-color: var(--google-blue-600);
  --primary-button-text-color: #fff;
  --quiet-background-color: rgb(247, 247, 247);
  --secondary-button-border-color: var(--google-gray-500);
  --secondary-button-fill-color: #fff;
  --secondary-button-hover-border-color: var(--google-gray-600);
  --secondary-button-hover-fill-color: var(--google-gray-50);
  --secondary-button-text-color: var(--google-gray-700);
  --small-link-color: var(--google-gray-700);
  --text-color: var(--google-gray-700);
  background: var(--background-color);
  color: var(--text-color);
  word-wrap: break-word;
}
```

```
.nav-wrapper .secondary-button {
  background: var(--secondary-button-fill-color);
  border: 1px solid var(--secondary-button-border-color);
  color: var(--secondary-button-text-color);
  float: none;
  margin: 0;
  padding: 8px 16px;
}
```



```
.hidden {  
  display: none;  
}
```

```
html {  
  -webkit-text-size-adjust: 100%;  
  font-size: 125%;  
}
```

```
.icon {  
  background-repeat: no-repeat;  
  background-size: 100%;  
}
```

```
@media (prefers-color-scheme: dark) {  
  body {  
    --background-color: var(--google-gray-900);  
    --error-code-color: var(--google-gray-500);  
    --heading-color: var(--google-gray-500);  
    --link-color: var(--google-blue-300);  
    --primary-button-fill-color-active: rgb(129, 162, 208);  
    --primary-button-fill-color: var(--google-blue-300);  
    --primary-button-text-color: var(--google-gray-900);  
    --quiet-background-color: var(--background-color);  
    --secondary-button-border-color: var(--google-gray-700);  
    --secondary-button-fill-color: var(--google-gray-900);  
    --secondary-button-hover-fill-color: rgb(48, 51, 57);  
    --secondary-button-text-color: var(--google-blue-300);  
    --small-link-color: var(--google-blue-300);  
    --text-color: var(--google-gray-500);  
  }  
}
```

```
</style>
```

```
<style>/* Copyright 2014 The Chromium Authors. All rights reserved.
```

```
  Use of this source code is governed by a BSD-style license that can be  
  found in the LICENSE file. */
```

```
button {  
  border: 0;  
  border-radius: 4px;  
  box-sizing: border-box;  
  color: var(--primary-button-text-color);  
  cursor: pointer;  
  float: right;  
  font-size: .875em;
```

```
margin: 0;
padding: 8px 16px;
transition: box-shadow 150ms cubic-bezier(0.4, 0, 0.2, 1);
user-select: none;
}
```

```
[dir='rtl'] button {
  float: left;
}
```

```
.bad-clock button,
.captive-portal button,
.https-only button,
.insecure-form button,
.lookalike-url button,
.main-frame-blocked button,
.neterror button,
.pdf button,
.ssl button,
.safe-browsing-billing button {
  background: var(--primary-button-fill-color);
}
```

```
button:active {
  background: var(--primary-button-fill-color-active);
  outline: 0;
}
```

```
#debugging {
  display: inline;
  overflow: auto;
}
```

```
.debugging-content {
  line-height: 1em;
  margin-bottom: 0;
  margin-top: 1em;
}
```

```
.debugging-content-fixed-width {
  display: block;
  font-family: monospace;
  font-size: 1.2em;
  margin-top: 0.5em;
}
```

```
.debugging-title {  
  font-weight: bold;  
}
```

```
#details {  
  margin: 0 0 50px;  
}
```

```
#details p:not(:first-of-type) {  
  margin-top: 20px;  
}
```

```
.secondary-button:active {  
  border-color: white;  
  box-shadow: 0 1px 2px 0 rgba(60, 64, 67, .3),  
    0 2px 6px 2px rgba(60, 64, 67, .15);  
}
```

```
.secondary-button:hover {  
  background: var(--secondary-button-hover-fill-color);  
  border-color: var(--secondary-button-hover-border-color);  
  text-decoration: none;  
}
```

```
.error-code {  
  color: var(--error-code-color);  
  font-size: .8em;  
  margin-top: 12px;  
  text-transform: uppercase;  
}
```

```
#error-debugging-info {  
  font-size: 0.8em;  
}
```

```
h1 {  
  color: var(--heading-color);  
  font-size: 1.6em;  
  font-weight: normal;  
  line-height: 1.25em;  
  margin-bottom: 16px;  
}
```

```
h2 {  
  font-size: 1.2em;  
  font-weight: normal;
```

```
}
```

```
.icon {  
  height: 72px;  
  margin: 0 0 40px;  
  width: 72px;  
}
```

```
input[type=checkbox] {  
  opacity: 0;  
}
```

```
input[type=checkbox]:focus ~ .checkbox::after {  
  outline: -webkit-focus-ring-color auto 5px;  
}
```

```
.interstitial-wrapper {  
  box-sizing: border-box;  
  font-size: 1em;  
  line-height: 1.6em;  
  margin: 14vh auto 0;  
  max-width: 600px;  
  width: 100%;  
}
```

```
#main-message > p {  
  display: inline;  
}
```

```
#extended-reporting-opt-in {  
  font-size: .875em;  
  margin-top: 32px;  
}
```

```
#extended-reporting-opt-in label {  
  display: grid;  
  grid-template-columns: 1.8em 1fr;  
  position: relative;  
}
```

```
#enhanced-protection-message {  
  border-radius: 4px;  
  font-size: 1em;  
  margin-top: 32px;  
  padding: 10px 5px;  
}
```

```
#enhanced-protection-message label {  
  display: grid;  
  grid-template-columns: 2.5em 1fr;  
  position: relative;  
}
```

```
#enhanced-protection-message div {  
  margin: 0.5em;  
}
```

```
#enhanced-protection-message .icon {  
  height: 1.5em;  
  vertical-align: middle;  
  width: 1.5em;  
}
```

```
.nav-wrapper {  
  margin-top: 51px;  
}
```

```
.nav-wrapper::after {  
  clear: both;  
  content: " ";  
  display: table;  
  width: 100%;  
}
```

```
.small-link {  
  color: var(--small-link-color);  
  font-size: .875em;  
}
```

```
.checkboxes {  
  flex: 0 0 24px;  
}
```

```
.checkbox {  
  --padding: .9em;  
  background: transparent;  
  display: block;  
  height: 1em;  
  left: -1em;  
  padding-inline-start: var(--padding);  
  position: absolute;  
  right: 0;
```

```
top: -.5em;  
width: 1em;  
}
```

```
.checkbox::after {  
border: 1px solid white;  
border-radius: 2px;  
content: "";  
height: 1em;  
left: var(--padding);  
position: absolute;  
top: var(--padding);  
width: 1em;  
}
```

```
.checkbox::before {  
background: transparent;  
border: 2px solid white;  
border-inline-end-width: 0;  
border-top-width: 0;  
content: "";  
height: .2em;  
left: calc(.3em + var(--padding));  
opacity: 0;  
position: absolute;  
top: calc(.3em + var(--padding));  
transform: rotate(-45deg);  
width: .5em;  
}
```

```
input[type=checkbox]:checked ~ .checkbox::before {  
opacity: 1;  
}
```

```
#recurrent-error-message {  
background: #ededed;  
border-radius: 4px;  
margin-bottom: 16px;  
margin-top: 12px;  
padding: 12px 16px;  
}
```

```
.showing-recurrent-error-message #extended-reporting-opt-in {  
margin-top: 16px;  
}
```

```
.showing-recurrent-error-message #enhanced-protection-message {  
  margin-top: 16px;  
}
```

```
@media (max-width: 700px) {  
  .interstitial-wrapper {  
    padding: 0 10%;  
  }  
}
```

```
#error-debugging-info {  
  overflow: auto;  
}  
}
```

```
@media (max-width: 420px) {  
  button,  
  [dir='rtl'] button,  
  .small-link {  
    float: none;  
    font-size: .825em;  
    font-weight: 500;  
    margin: 0;  
    width: 100%;  
  }  
}
```

```
button {  
  padding: 16px 24px;  
}
```

```
#details {  
  margin: 20px 0 20px 0;  
}
```

```
#details p:not(:first-of-type) {  
  margin-top: 10px;  
}
```

```
.secondary-button:not(.hidden) {  
  display: block;  
  margin-top: 20px;  
  text-align: center;  
  width: 100%;  
}
```

```
.interstitial-wrapper {  
  padding: 0 5%;  
}
```

```

}

#extended-reporting-opt-in {
  margin-top: 24px;
}

#enhanced-protection-message {
  margin-top: 24px;
}

.nav-wrapper {
  margin-top: 30px;
}
}

/**
 * Mobile specific styling.
 * Navigation buttons are anchored to the bottom of the screen.
 * Details message replaces the top content in its own scrollable area.
 */

@media (max-width: 420px) {
  .nav-wrapper .secondary-button {
    border: 0;
    margin: 16px 0 0;
    margin-inline-end: 0;
    padding-bottom: 16px;
    padding-top: 16px;
  }
}

/* Fixed nav. */
@media (min-width: 240px) and (max-width: 420px) and
  (min-height: 401px),
  (min-width: 421px) and (min-height: 240px) and
  (max-height: 560px) {
  body .nav-wrapper {
    background: var(--background-color);
    bottom: 0;
    box-shadow: 0 -12px 24px var(--background-color);
    left: 0;
    margin: 0 auto;
    max-width: 736px;
    padding-inline-end: 24px;
    padding-inline-start: 24px;
    position: fixed;
  }
}

```



```
    right: 0;
    width: 100%;
    z-index: 2;
}

.interstitial-wrapper {
    max-width: 736px;
}

#details,
#main-content {
    padding-bottom: 40px;
}

#details {
    padding-top: 5.5vh;
}

button.small-link {
    color: var(--google-blue-600);
}
}

@media (max-width: 420px) and (orientation: portrait),
    (max-height: 560px) {
    body {
        margin: 0 auto;
    }

    button,
    [dir='rtl'] button,
    button.small-link,
    .nav-wrapper .secondary-button {
        font-family: Roboto-Regular,Helvetica;
        font-size: .933em;
        margin: 6px 0;
        transform: translatez(0);
    }

    .nav-wrapper {
        box-sizing: border-box;
        padding-bottom: 8px;
        width: 100%;
    }

    #details {
```

```
    box-sizing: border-box;
    height: auto;
    margin: 0;
    opacity: 1;
    transition: opacity 250ms cubic-bezier(0.4, 0, 0.2, 1);
}
```

```
#details.hidden,
#main-content.hidden {
    height: 0;
    opacity: 0;
    overflow: hidden;
    padding-bottom: 0;
    transition: none;
}
```

```
h1 {
    font-size: 1.5em;
    margin-bottom: 8px;
}
```

```
.icon {
    margin-bottom: 5.69vh;
}
```

```
.interstitial-wrapper {
    box-sizing: border-box;
    margin: 7vh auto 12px;
    padding: 0 24px;
    position: relative;
}
```

```
.interstitial-wrapper p {
    font-size: .95em;
    line-height: 1.61em;
    margin-top: 8px;
}
```

```
#main-content {
    margin: 0;
    transition: opacity 100ms cubic-bezier(0.4, 0, 0.2, 1);
}
```

```
.small-link {
    border: 0;
}
```

```

.suggested-left > #control-buttons,
.suggested-right > #control-buttons {
  float: none;
  margin: 0;
}
}

@media (min-width: 421px) and (min-height: 500px) and (max-height: 560px) {
  .interstitial-wrapper {
    margin-top: 10vh;
  }
}

@media (min-height: 400px) and (orientation:portrait) {
  .interstitial-wrapper {
    margin-bottom: 145px;
  }
}

@media (min-height: 299px) {
  .nav-wrapper {
    padding-bottom: 16px;
  }
}

@media (max-height: 560px) and (min-height: 240px) and (orientation:landscape) {
  .extended-reporting-has-checkbox #details {
    padding-bottom: 80px;
  }
}

@media (min-height: 500px) and (max-height: 650px) and (max-width: 414px) and
  (orientation: portrait) {
  .interstitial-wrapper {
    margin-top: 7vh;
  }
}

@media (min-height: 650px) and (max-width: 414px) and (orientation: portrait) {
  .interstitial-wrapper {
    margin-top: 10vh;
  }
}

/* Small mobile screens. No fixed nav. */

```

```
@media (max-height: 400px) and (orientation: portrait),
  (max-height: 239px) and (orientation: landscape),
  (max-width: 419px) and (max-height: 399px) {
  .interstitial-wrapper {
    display: flex;
    flex-direction: column;
    margin-bottom: 0;
  }

  #details {
    flex: 1 1 auto;
    order: 0;
  }

  #main-content {
    flex: 1 1 auto;
    order: 0;
  }

  .nav-wrapper {
    flex: 0 1 auto;
    margin-top: 8px;
    order: 1;
    padding-inline-end: 0;
    padding-inline-start: 0;
    position: relative;
    width: 100%;
  }

  button,
  .nav-wrapper .secondary-button {
    padding: 16px 24px;
  }

  button.small-link {
    color: var(--google-blue-600);
  }
}

@media (max-width: 239px) and (orientation: portrait) {
  .nav-wrapper {
    padding-inline-end: 0;
    padding-inline-start: 0;
  }
}
</style>
```

<style>/\* Copyright 2017 The Chromium Authors. All rights reserved.  
\* Use of this source code is governed by a BSD-style license that can be  
\* found in the LICENSE file. \*/

```
.bad-clock .icon {  
  background-image: -webkit-image-set(  
    url(  
QVR4Xu3cS1OTVxwG8Ha6dsZNt/0S7ozX  
+wUSGKN7yclvkJ2OiNcdvbwMfABXLS1VvLXFSMWUgFAh1oJICCEGq8UFp8+fPu87J  
4S3vrncvlfOceaZMKOSnN/8z/  
+c95Yv3B8XFxcXFxcXFzNRZ89+rZLJJNKLDCAZdfp0Hini5zWJ/lzk5e+QAf7bpPzf/  
yvKLkD0l1IArCOqwazL70D65Xdub5RUagcA0kDJCYCJ8Hen5b22D8y5czvVmTN9gCk  
RwHzwXvKe8t72wij1JWB6AFMIBDAPVZDPIJ/  
Fth7zDWAeESD64LPIZ7ICBwAxZBFRIUmURiUWLk0ymuCwrG8NtQyqKKfUVYG4RwP7  
gs8pnbhsOAO4gapvITluQzFaO+Uoy3nNMAayurlalcOqUKaSUudUKTa9dQN/  
t3m0EiWOltX6fw6W8nUCDyFJHh5ktAMbUuh0yN4HtBPpWgGlxNYQUTSBhTC3Zccv  
W3QNoawUR6AfK7p49ZpAwtuYPPHIsFukFEehHAA0jy61GwthkjM1UTx8Blq2gleDcQ  
+7v3atKnZ2trqK+xs/n8JRFVBUEJL+CBGgYQD/  
t369WWomEMTZ0PgkAaQ8g6ik2RjyH+/  
apxwD6BSm3tpLS9QPXTGDUFTQkTZrT6wGAplJGkKcHD6p38XjLzkzWfw6ZADb0oHv  
AEaBHAPpZKohAv7YSCWOup3r6CRBpBX3v9R  
+pHuSxAB04oDKl4EjGDh1qDRLGXA9Q1iagYQFigxagJwR6BpwxptIsEsYc  
+roVANajBhpkcXag+3qDBs5TRMd5jowfPtws0rqMPUz1JAKqBQ8i0F1vBWP/  
GQHOqEwvgWGywPmNWW0GCWMPA9RrA9Cg339YPQR6AhwBGtuEM47kjhRE0jD  
SBh7GKABC4CqGzSBZHplNCDA6PGBXhw9qt4nEo0ADYQBytgCJLvnB97yLs2ZQM  
+Q59WV4+NM8nWqESSMPcwOOm8BEA8v2KC5QcxlgxYgTq0sgSY8HFAP4Ewjvx87  
Vi9SPgxQ0QYgHqD6q9cIK2jU7z2sHIY4NUAzyF/hkYphptiaDUBjFy6oYa//  
aLtnArHn1FaOjvMSkdfZsEgY+7YBqpTLKnxonrl/jPiTS/  
ijBPHAxKYFxrONDLD5l8f33j9AKQQQHZOsaVr1wSlBmkcSN7hxagGIPtM5Xg4swR6xX  
wGqWhnk2aWr1/fEmni0iUfiNPLX84nQ+L8wbw+ceK/  
kPKGIInnzSFNA8nbPOQBNalOKOEw1DiMwfuaQj11dDS7z3ChGmVIA0nRvr8oCaKPvb  
MJ5ycxqOLObKudP4syfPCmpRcLYDRxqmMIKANLM5cvSf/  
zqmUl4rQik9ZwAnDnkDbKA/  
K0jYeyhD1ZtRnoFJKxSWuVwWjHEERjyCsrR8N5iyu3i4iPhLEbON1hNuUApNdXrgAoo  
HJqgHQchDiSApE  
+dXXxdleBE2am8+7GjS2R5oBU1ZCDp5WfBR2HweVtueaWNXDKNXqkeSARRlutiMP  
404pZJJCHU0RwKanf8EI786kEIL0BEisnsOfOJIClZjLSLmjY5fFI33CZ/  
XmzSCkwMqZD5hWgkOgnIELh/  
Yhvb16Vc0RqXa1Ympx1EoikTZw6TnavA9AWujpERgdh0B+5bAp  
+72npBKJHQZuXrAPqXD7dIDIEldVw  
+BGCKmePgO3v9iHBjzaymF0nCUNB9f3C6q7e6eBG6jsyofz57V9DrOpIRc0GMkKUun  
s7DFwC56dwYGnpvTX9hzGwynF47wFz8BNnLYGx1ScUn64WIXjIOPxxY8Yk6HbgO1H  
0nsOo+OsVRKJmNEbyW3PGpAAhGg9hwFOyj2KQCR/  
WjG4TeaWe5hiE5LAIBG8bvUwi3sc6IN390blGMBxD9S5RzINxD3U6x4Ld18s4L6awt64  
LzdxX4/zDxj9/IEueAvhAAAAAEIFTkSuQmCC) 1x,
```

url(  
BMVEUAAAD/gID/gID/VVX/VVX/Tk7/YmL/YGD/VVXzUVH/XV32UIL/W1v2T0//  
WFj3UIL/UIL3UFD/WFjwTk7/U1P/U1PxTU3/V1fyT0//VFTzTk7/UILwTU3/VVX0UFD/  
VFT/VFT1Tk7/VVX/VFT/U1PyT0//VFT/U1PxTEz/UILuS0v/U1P/UIL/VFT0T0//U1P0Tk7/  
VFT/U1PuTU3/UILzTU3/U1P/U1PwTEz/UIL/U1PvTU3/U1P/U1PxTU3/U1PzTk70Tk7/  
U1PyTk7/U1PzTk7/U1P/U1P6UFD/UILzTk7/U1P/U1PyTk7/U1PtTEz/UILyTU3/U1P/UIL/  
UILxTk7/UILvTEz/U1PvTU3/U1P/U1P/UILxTEzXTU3zTU3/UiK7Ozu8Ozu8PDy9PDy  
+PDy+PT2/PDy/PT3APDzAPT3BPT3BPj7CPT3CPj7DPT3DPj7EPj7EPz/FPj7FPz/  
GPj7GPz/HPz/HQEDIPz/IQEDJPz/  
JQEDKQEDKQUHLQEDLQUHMQEDMQUHNQUHNQkLOQUHOQkLOZWXPQUHPQk  
LPZWXQQkLRQkLRQ0PSQkLSQ0PSZmbTQ0PTZmbUQ0PURETVQ0PVRETVaGjWR  
ETWRUXXRET XRUXaGjYRUXZRUXZaGjaRUXaRkbaaWnbRUXbRkbbaWncRkdbRk  
bdaWneRkbeR0ffRkbfR0ffa2vgR0fga2vhR0fhSEjha2viR0fiSEjia2vjSEjibGzkSEjkSUnkb  
GzlSEjlSUnlbGzmSUnmbGznSUnnSkroSkrobW3pSkrqSkrqS0vqi4vrS0vriYnri4vsS0vsi  
YntS0vtTEzuTEzvTEzwTEzwTU3w6OjxTU3x6OjyTU3y6Ojy6eny8vLz8/  
P0Tk71Tk72Tk72cnL3T0/3cnL4T0/4cnL5T0/5c3P6T0/7UFD8UFD9UFD/  
UIJWZWgAAAAYXRSTIMAAgQGDA0NEBUWFhwcHR0fHyAgNDQ3ODg9PT4+QkJD  
Q0ILS15fdHR1fHyEhIWGiliJiYuVlaioqaurR+vuLm5u7u7wsLExMXGxsZM0tTU2dna2t/  
p7Ozt7fPz+fv+v7+jD+tjQAACYhJREFUeAHs1cFqwJAcx/  
G1FR0ilqKIFFEUHKJQKIIRFKGUilSKVCmiHrKpCDuPHcbA99xtjA1+b7HLjmMkaelu  
+TzBl18C/xtFURRFURSFQ6bc6g0ns8Uq3u3i1Wl2GfZa5cz/tKQafW  
+NX629fiN11Rij1p3v8af9vFszrpRTGWxAZTOoyK8pdpZgsOwUpeaUxgSMYlgkLac6BZ  
dpVUpO3QM3ry48JzdClqOc0By9vUVC27YurscMIUBoiprHlhCCWEJGyvsQxhfwk5oxBl  
qbCXM0m0AoYmuJrqgD4RyDvyftQgl3zduTDSBFcMvXU4ggSVTg2ieCNFGW4/8EkCh  
g/keGC6lcg61Hc8Dg7cc76DkaU5ANnqAnliKb6V4QrqB7liLcCExyMfiCHp4/QC3O0/  
boPniDDi8MRb5OGWSBN+hwOL4yFFi0PSYB/  
0LH0+UTtIhJ9WAhEgSdTufLF2iFNI92ByRa6PzIUPTNih21uFG1cQC/  
Kewn6GWh0BdeSu1NKRRKS2+EUuiF2CKI7UWR3WabNHFmk84k2WSTndmsGyfZMc  
42ziYmJhozbo0hXSVsMYRYFhEEFWt7Uau1V3q+hf+Ts2ZXMSmMnHk+wY//ec55zjn/d/  
Heuc4DmqUg/4pr0XXnO+1Zwp2QP5B1LTrr  
+D4IHKC3aQ8BFHxn3bXI6U37OuFPKBaUI4U/3b6yHf4TCIqrh3w  
+vwCQVHQrmv4TcYFwJ3THHxIkSYqZLkUXpv7/  
EC4QTmrf3J1QSAwDtFh2KZr2f3SOGzQXCnWNCblKx5LpqjvRuSn/dbclbw/  
5fYQGQIN6T5EQ6rX7qSnRr8q/fcYLi2vaz876Allr34oIESIEzzb+lizo+EfQmJwie  
+WAQLXQvGk  
+mIEwma7kRvTHxv5dwJ3R7PiAloixHk8mUmnErmvR7fJoXNHsbp3RQkMLRaHw5BY6  
maV+4EJ2e8PK5xg2apaClijZOJjFvDarZXNtZ9E1+zfREclNoi0UXJBjMbpiq/  
BouZzWcRYdsQWd4AfNYY4tLOAQiidVdVWjnpuyf  
+UoOmELusQP8gfQ0wuyneBCqWdIdHh0w1F0yc5z6CY3aM6POSZIsryYSCtolayXXsg  
bht51EN08ZAM6SrhBPI9IFDDpFxfT6QxaGvkUUEbR6DuljtqAtvGD/  
L67oihF5MTSUIrNaDoKHN0oFs2vydQ6ZXs141+yu4IQFumeX1LX1vKaRj2GUQSo9M1  
/vqZd9SAhP0BhKYEouJrWr6gF3Tq2TBNs/SYTKmr454Zwg+ax6AXo/  
F4Mqkqa3I4DJYPOGalskum1MwY6LAHIAQkShIGq6KghfI6Wy8EVCqVKpXyt2RyHR4  
D/c8DUCAgRiJR3Dxw9UBLG6yfTXjMUqVarX1HJtaxMdBjD0BBQYhEpEQqqWBs5KjI  
+CeeKkDTRcfHQGc8ANGAZDmeSuGYph7WP0UmqtUatcb3ZEKd4X9v2CUUBAhjQ1I

ZwdgolB54hhzkU6k1GvXmD67fHpc8AAhSCQmJxQEIf1f3yiAM/  
SUwUFA9Uaz2fzR7TS77AVIACiWximEMVYoDNeL5VODh3KshmUvujwGuulBKBSOyA  
Ap6iruHRs4DunuMoftzECWtbVI/URs6soY6C0PQO+KmPRoIUx6BASQud8/  
aCDLagLUav1s9wM6BrrhAejJe1EJPa2q2bymb2zQfEbx1Fk  
+8LTaNqIbXoJePt8XxZMJVVWYGKvY88P1QjtXWf8AZG21HrTbrV  
+cQTxL9ur5vuj9ZAoLItHzBlasshcQPA14miyfdrvTeeawZJxN/  
fuTA6LIIJrVADKKFRQ81RryoR29hQKo1elsbz9zbOrLxCPRB  
+oqNplxv1iulPfyozRhBoMG+WwD1P3VadtJF6JPqRzo3Afoio7EJt1ixaLp418HnZR/  
xZdHAOdJ96JNGx6c7NC46Ee1Kh/Oggl+XS3d3ZekAN13nm4cog  
+Wsce29xkGx7FNrw1BIHzcHunu7PTe/Ri6nA9SbwTPf14vWCW2X5nG77J8kE/  
dxAPDajX6/V/I6N6bQx0jHgp  
+sQslsusf5qsSilctoUNOT0+oN90THuK6yD6DNcERusfxoW2+5DD2sfgPr9/mDw  
+OXkK+wM8Vb0eam6Ny8sxnkAUBugIQee3mAw6I9EM/  
bPIG7R0wOien00MFp0YAw5XfRzr9vrPRr0wdnd3X016RmEh6IH9ccB0ZeUM1qvv7s53  
9YmgiCMx5S  
+KP4RFUREBKFaFSqiSLWCVVSwBdEKIr5QP6IgtYrYiNe7unpudleWpHlrxdRv4cwOm  
yYVmSudg6XzCX7MPnN5sjsz+P0BoCXIz+DAsizPikB0T  
+avNEP0mvwP8dD3Bw5smWgQBzJUFCFHV5jLBiGiN+R/IPD3Ar/  
PpGcssCz7nCNPdQGXTYw1zEiRG/j95n0PBB00DPyKOA9QZdx/  
A2XyBH5H9IPwCEcg4FFs9LIWtaqwfMIZ4g0TvSc8BBQbeiniE/GeSnhNDmotylJ09E  
+UEBYX0BzUfAQQGhfABIK21OyV0L80QfyP+8h1hGnhbVF  
+AUOfJobZ6NSV6c80Tkf0jP+H2m+qIEKeAxVyWeFpj4uUXUXQk/YLG  
+SM5ZEWhK5DHHG/+Jx4JEv4ZytEL+B3FIPwWEWqPzMuaR8PMUT9T9tBR/  
UGO5F0E+CnisPS/zgMcTdbeIVsn/oH5Az3mQDyWobV8cEX/i5InWV1sD/  
WR4XsRjMEE3+UfgWojl/  
2SxvEJ9tYHHHZN7JueJ1iNQ7yv5n3BeOeGUxgLPPrRoacXii3rfof0J  
+FMonCNq4E7KtFjwR8Qz5nyhnAzTO3ZVvRuGJel9G/  
A8dmGnjgbnTDSZuyBNhfkB8T6mNBhxnbn8u39DEE33/x/9oAwFE/vn+Bhvn/  
kjH73y7/9EWcaz3Z+tpiuOiX2zzP8biiXn/sCnfNliRaMT/AJBzwPPypHxjZVUiNex/  
qL6cvyTfelqdqBzyP6Sfzv1mXc25IYj0wP+0rYPwTw/W1r5cjchE/wM8oOhXZ  
+pr8K5OpCk9cF7+co0t8JWJgn5MILq2L4EhgT7yBBx/eyyJMYq+Czw/OgvjiQya9G3Iz  
+JEMqM4/Y71ncXDCQ0rbXb8wkRS41ybd8bTH3hLfYQw/aHJusZKDyU/  
eLsHRpMpSVO7H96easqOt8/ujmf2wF5fAJDgioQEIOjQmo0deYAnF46mv4gk/  
VUtKS2zkV/3Mzk9Mzcf1/3Mz81MTwqt+/kLc5W5R5JoGz0AAAAASUVORK5CYII=) 2x);  
}

</style>

<style>/\* Copyright 2017 The Chromium Authors. All rights reserved.

\* Use of this source code is governed by a BSD-style license that can be

\* found in the LICENSE file. \*/

.captive-portal .icon {

background-image: -webkit-image-set(

url(

5HvMAAAEYIEQVR4Xu3afWhVZRwH8LvdmjVSjKyQbKazia1k1sL6YzXoZdy28/

J8f5XRC7hwEVk0hZQCXY2SipCgYgappXMq5Ya9ELZqoEM3VHqxBkapm21jOBWmb

[illegible]



w2kk0UysLz3PaVIOdJMkHsfo2pmbHXPfZ1z2A4RPTDb/BG/qLX6r/  
AK9FRRBzqnk1hFZkAReL/qd8E0SC8iZJwJhGewgXP+v/T9dSEM1GCN2kQaZOBdTo/  
LfOhhhqOCt3Pwx413FD4iS1CgAWCsNvEN2A36RO9W/wxlggq/uYt0yjH  
+DW1Ri1Us69eC9Ua/JWPCS1mX  
+1quAgLrRFEFMvnNnG7cbyDeRh9i2ntPLyTMEi0YSrpgjGo5057BEkw74ab/  
Ut3hieSjZn8MvbyEW7idm7il9jLL2NmJpTa9ujV/  
0lVPzHw9Ag9VgiScGl3Zsd5gCh3iPMEasRTIp+vGvWjcKbzAPZ16cF6A2OP  
+HFZKYgyeLtaQhmUwc/ypyl/5SfDWdiMW+nDO03lxiaLRMk4RA  
+SF07PuAQaZMhXuyWC8K0XqrfTrpgHXfaLUgWfHo1GBzSLv  
+gCHHbBUImjMRHqenBRzNGaoekluayWJAsAN0Ql7VjL1bgaWcWnsYLeJ/br  
+u5oF3uCWekiN1qQRIn3DXHQTN  
+XXQPCQqH8fJrxag2J0y68FohwHJBEjUblWo4USwfr  
+Aot3ALjqKsOxUWHcFvq9kGsoWIWyPoEnmgS0SiaFRGR5jKF9bZ8/  
zwKfIA3u9JNvaSCXipRXo6sYM8oCb0kJK/  
pCYYKeRwiw1idINJPH6T6sfJBnhmkx6cLb6DPKJeFKJfNFUI7bTpUj8l7wzmPdcF7THU  
DcTVNunBjtliko1PuvTgk0i2oRYEm/Twn3KHkCbOl2hFq/MlmYGrbJGDj/EUGUHNv/  
ON9fdYoKYF9diCH4dvl/  
tAhQWhg704YxEPBCUBJckFBILqA0FJwOTkAgJBZYGgpMUdPh8lSoKallxAlGilbM7Hv  
kCQADVcWKLcYXwsEHTDgiW5l5d5+Nq08V/pJ8h5WK4kSnGxlRMR/  
6y5u1fPyeVL6SalBslanxOhVOBSIWjbDQuNOqwWpJ8vLfWQJFML6Dr4eboJUguEoGp  
KBdnIjzP/WGFaDy70q6AJsjsxpaw2cuBKzOLxa4YFvU/  
9SO4QOWyksHw4licE1ffwke8aFNSG79oUIMfyyA01V9zA2z02AB82lucc/  
s4h6mdkSVrNITewWjxB66kHoqPwcboEElgv7ne1u9FyYXT+Lfv/  
LqSHIDVf3EF5bysZhcn6btJBEBcKQVW9jMOSLSHznka8DZ6gKWJlqXE3WitG9YcoCfj  
VwBcUe0jCQS25gQYRnYzVzTzaX70RT1ADuSHDN3Wva0/7WwNbkLo3ITR3Qw69oSx  
yYcrtfHAgCwplyYmrUGyZ4JPpJkg/Z+YJDqem9LvJ6Y/SMsl7os0G6T1X/MSFeMr/  
xeveT4gBOVTyuAXPhelzmEr5xgXIC8EHSA3sEuGGpb147eob/kZamCXTrDqziB  
+w29FqPQ3WNVNd7i/  
NPf4rKjJ13THKnejc1wSZq4pW78XLvibMPOQcvWwsuyMrYLcU64aSfvUud/  
BxYEhKHcIX5ZJey9lHw+oH3C7HYL0yz7uhUNPqLIWCNlVHLqXnm1rA/Wz9KzRvOAF/  
NkCQfrNC+7tLxpLgXdalEi3/  
cW9gYo8lBb3WYqIF8mEvdcWvH2kQXQEGmwVJFsL1UrvTZw5Woq  
+zedtFMQ5Urya5r0NeDlpob7HX1ooaLkQdD6U5b2R/  
ARpgpl8xTZBMpOOMq2lCLF80gS/  
tEtQLF9qx2TqHXlfcWwmbewShM1Cz3HNeTA6iselk6DicXKuhxL9BXUb00kQ/  
ij0xJ2x2ksy0RodlS6CoqPQKgRVmFnUuyFdBPEGqTw6yciycHyOMekgCGMSipxV5jYW  
els08G3h+HF+w5mV+jQvsR8lVmBwawoU2SNIXjjJTCmAogTB1WY3NznlfYm//  
6sYsd5tJ83wbXxKCnKmkx6Jx0BgBXkE//  
BbkHsdDy9lPdhqfIMldHg4KUCkcn2+OFnDjrk  
+IXRpl9alq3z6cJh5AEnty8E4aQoVMltAU9LmWqJ0U3e9OvjeLcvFDmzqEdQmSCzLpx  
JAoPbBGKxtyoUWv0XhL9RD2Cx1IM4HiWB2Y0m272N/  
mp2Hwg63uPbOLGQuZyExrcqxUVMJA/gZ9zudw8RJYCJiaVw7Hf9eelvdovG6HiP  
+4J86K+ixBZTNEo9al7dT+7ob5eMkx43GR7kRPhVHMFvsKX/  
RU0Y2Ti1BBxl3jA6lbbtUKRJSGl0FobqA/ryAWjW7bjZHS8rYKi43HyJj1bXMIR85v  
+oxETbRSEiWi8Sc9ut6jfl2MjcNGZbpsgZ3oPTVyHoneTBjoHj7TzIpsE8aKbJxOo1zr1Sf/  
oGIQWDrND0PWgQujRylcaO/zotJqilcjP65DG02Py+Cx0YEX4NtsEYbfO2GP

```
+ALZTKLJJELbov7nMH+H3FsbYIAhxrNOY9/h5CCQ+5w3RUf0rCM26QYXPx4iilTcWj
+svQdhvliT1/SBadGBzLL+vBSHOa00mNPw/yvgEL
+ecPhNUJ7OF9jxHS5MfMYF9vMh5mAb5KqiFI4YzyVYwGttSGDq3qQVqQu4Q84KwDa
PJDvQP5EcH6vltrDemp1qEzHYTK+jjAwOqYgU00lhOQoX/
274jjgrR3zPgdsYvQb1vcupR4oylgQ8mo8xwb8d5lGEypROhLDVNrUSN3nan6ECNWq
mmhbloXQnf5US4lKvR2CsxjVzNpU5ELFhKd5yhsTw1F6u5nKtQw7VoQDO3cRua0cC1
qOEqLsdqNTEw5wwla/gvpXzJeo7GTncAAAAASUVORK5CYII=) 2x);
}
```

```
@media (prefers-color-scheme: dark) {
  .captive-portal .icon {
    filter: invert(1);
  }
}
```

</style>

```
<style>/* Copyright 2021 The Chromium Authors. All rights reserved.
* Use of this source code is governed by a BSD-style license that can be
* found in the LICENSE file. */
```

```
.https-only #main-content a {
  color: var(--google-blue-700);
  text-decoration: none;
}
```

```
.https-only .icon {
  background-image: -webkit-image-set(
    url(
VEUAAADcRTfcRDfdRET/gIDcRjr/
ZmbjVTncRDfcRTfcRDfdRDzgSTncRDjeSDvcRTjbRDfbRDjeRzvcRjfbRjjcRTjcRTjcRTf
dRTfcRDjdRTjcRTjbRDjbRTjbRTjbRTfcRjjdRDrcRjfbRTjZQzfcRDjZRDfZRzbWQzXXRD
XXQzbXQzbWQjXYSDvWQjbbRDfOQDPSQTTUQjXCPDDNPzPJPjLGPTHVQjXMPzP
RQTTWQjXLPzPDPDHYQzbAOzDTQTXHPTLIPjK8Oi++Oy/
FPTHEPTHPQDTQQDTUQTXBPDDKPjK/OzC9Oi/////
PQDPRQDS3OS66OS7TQTTEPDHXQjbMPjMBhLaWAAAAL3RSTIMA4tgPAhYFCcL9
8B4x9ie1+s49WICbqXNKZY3pjuqcgVdLZnL2qKg9zmXpjfontV8LANsAAAJrSURBVHhe
7ZTndulwFAY3ARlgBAg9vW1v173ROylby/u/yso2Fx3MNaxs9h/zAHM+Sfa8+M/s2LFjx
+3tdjwH+/
sHWxHVAerb8KSyANnUFkRXwLiK78llgJHJxRalwSMd11OGOeV4nsM9FO0dxhJdw4Lr
OJ6jYy46PoohqgEHatE9JViiFNWTPiEITpIRRxcQ4C6aJ3EJAS4TkUQXsMJFFE+
+CCsU8xFEBSAoiHsaQNIQ7yuQCFe3DiHUhftKllzdKoRSFe0r8sXDAkSoumkligYalOkIf
eWi56EESFm8r1w0fFI4epWgBA9qOMpmirCfeWijtoa9WSx6taAELFBRI/
vilS3BJRlBk9/VFTsLrifUXRuNfXLU0y/7m6p0CKxqN+v6lJU/
k3eJxu7Os5LWKDHi1tYstKG1zON1X3DGiRMR80Mx3fdCbc1+bQe3o2SJrYXcV0fFMx
L9xXiz0987BBtux65qaCeF8IHCR3FabBTQ3xvk4M1yN5B/
Mw2+urew8hTP1BM38Qnu5evK8gMw+7lcfH9E3ZIEBfMSO//Kf35+Cm6ua
```

+rhbSYDeEa9CUyW3qK1Hlj5DBz8dWd0bWCd6Ult/uMPeR+BmbV/JHrVG/  
a9MsEyV5fsK50R3frmBFXtCtVXmt73H4PhQ4t9k9rkJ55tYXwZrO4rCEUfPHfUEcuaZC  
/umw97TfaVpslu2tCb2lRWnBlKFtf+huwrjaa6Pxx7RfgW7nubJPtKI/X0puQO4k/Pfe/  
ovtLY7KbxVwve0/sE3VeaLoslkbEDvt8Hoq/hKGwQYvoq5OMnoq/hLAbgc/  
FVn33PX7pAfE5QHR6fAAAAAEIFTkSuQmCC) 1x,  
url(  
BMVEUAAADcRDf/ZmbcRjrjVTn/  
glDdRETdRDzZQzbXQzXWQzbXQjbWQzXZRDbbRDnWQjXWQzXYSDvbRTjcRTjbRT  
fcRjfcRTjcRTjdRjncRTfdRTndRTfdRDrbRTjcRDnbRDfbRDjbRjfcRjfbRTjcRTjdRTjbRjjcR  
TjcRDjcRjncRTncRTndRDnbRTjcRDfZQzbcRTfgSTncRDfcRjjZQzjcRTfVRDbcRDjcRDj  
WQzXeRzvbRDjXRDXXQzbXQzbbRDfeSDvWQjbVQjXIPjLOQDPXQjbCPDDNPzPUQ  
TXRQTS5OS7QQDTUQjW3OS7SQTTTPQDTFPDHJPjK2OC26OS7HPjHOPzPLPjLMPj  
PRQDTGPDHTQTTEPTHLPzPGPTG7Oi/HPTLKPjLTQTXyQza9Oi/  
MPzPFPTHDPDHBPDC/OzC+Oy+8Oi/AOzDWQjX////  
bRDd3undHAAAAQnRSTIMA2AUWCQIPHj39wvbO8DH64ifqqYFmtrVMc1IKS5x0nY6P  
WKqbjYDpZXWCZ1py8Jv9McJXV  
+KA9qioPc5l6Y36J7VmcHe8AAAFWUIEQVR4XuzWS4rCQBSG0euz56lSgiaEjHwgGh  
AhDnRF3/6HDY1la5WPjP4a3LOKY28555xzzjnnnHPOuSyzpPR7vb6lZAUrS8hgB7uBp  
aMEKC0Zhz3A/mCpaPjTWCK23GwtCcMjN8ehpWDN3doS8HPi7vRjejX/  
1CbX8qA1sdGZB  
+eRaW14sjGp8YQnk7EpVQqQe7peCFyupjMnYm4yGVGZ7q1EyTZbEEche2uUbLMIL  
5W6t4Zkm22lkm02561c89aQbLNTpPgq3hqSbbbmo1r41rhW8NaAaLMzvjITvDUg2Wzf  
lyrBWwOCzc6Jkm12QQcL3Vtlmy3opFC9VbbZJR0tNW  
+Vbbahs0b41rhc8FbVZqdEyTb724t5/bYNA3G4e  
+80NYl0gGFkvaR779KKZUWuFKe7nllsT5X//2M5VMZiZB9DQj74xW8ffrwjP90Mb/  
07Vf5CbXYJg0BtO4toKS9vhYHGY1vDZg28FQY6tBZls8tYBehwNLtyt1nhrTDQaDQcW  
Aux2SjWAXpOBpWMWSvm4q0w0Gg4nFQqFTd/m72HIYBYQJV  
+w83bZu9jRaDJYEB4osjJ02aFt8JASUBRq  
+PlarMrWBGI8lQajVanXA5kopUcvBUEGrCAWhSoXs3PZtewKhA/  
MMbTbcpEa7l4KwwURZSHANnVnGz2CVYGmg6oZ1u1XGy2hNWBCA8BogE1m7ZI  
+ShNVMrdW2Wg/v  
+Amr2eRYCcGLBZU2+FgcSBESDfdZxdwGbNvBUGihKgnk1OjPAEwS5gsybeCgNNdT  
QLyAtqtRCwWQNvhYH4ndjtNnlAnlet1ulQsFI9b4WBpgNyaUCEJ45DwGa1vRUGanU6n  
McmB+ZSnlosES3nvm/tUpGm1tFPd5DDAyKFBjGpzRaxSjW5J0o8/MAQ4ZEyKua/b  
+0Np175blMERDuaECFBZGqzBaxY9iAjIMbDK01U0OVZxcplE6BljLzFRixgQDwflCJaX  
cC+1ToKyOYHFvCOljPiNmvrTBRI  
+oQoGTk2Z1YQyleiWhIEftWnx8Yf8RcyiMCEkyhic2u4xOWSw9MBBQENTQFI83a  
+iL2rdgpJ1rms45mByYzhdNwt6qTtTIQC7r6FT/CLRQ02ZLWKc8OmK  
+LzooCykhKpl4q7p+7B/d0SjNggRbqGOzm1gPqL3PX3niZakOQsenf1PDWzWAXr  
+JBtEDQxnnJTISNmvrufBK75t45bORBNGSobcqb9DqBCjdQOI5E370xthbYaDRiljRDx  
KQwJk9a+o2u431gYZERBo/kcBlfvJ/TrSt6K1b+kDUHMkra2V3j5zRlprNbmADILbQ65S/  
z2ggyY82zL0VXsdQnnLdhSOKQzWbLWADIMpDgOrd3q958QiigrG3wusYzmNbXmY4s  
h+tangrVJ2Dgy97X9v0CmlLzzIHcj3ZPTL  
+h6DN7mhYR5nxHI4mtKNbLCaMaX9QDDKFO6C36hDttcdJQFGLeTWRIupocGOj62c  
Bb9WqesLTFwfm000MQgqz9IDLW

```
+Hve35HM9Fnqw9HetBkNsF6+Yaet8Jf0+xbka0XbYspSMlg
+5D8/8psnqdYv3qso1vsS9Hy6SaGQ6AYHP9ngLdqllVpilB8RygRQjGEdOsc4K26RGzk
6YTxbhDDdzXcfwC8Fbd8glPnR4Y62gBAM/
a1WybfYVNyyUBiZFPXYCAH70GvFW7nFRHH7Egyl8uAd6qXZ7NAqoilG6ZKuBH184D
3qpdAQIIWp0p9dE7wFv1q8Y6+njLoPI
+9P4C4K0GRKSjgTyywvoAeKtBxVWU6YhorovcvA14q0HtouwU0Fw/+jzN8w/cQ/
zg6ug2/QAAAABJRU5ErkJggg==) 2x);
  filter: grayscale(1);
}
```

```
@media (prefers-color-scheme: dark) {
  .https-only .icon {
    filter: invert(1) grayscale(1);
  }
}
```

</style>

```
<style>/* Copyright 2020 The Chromium Authors. All rights reserved.
* Use of this source code is governed by a BSD-style license that can be
* found in the LICENSE file. */
```

```
.insecure-form #main-content a {
  color: var(--google-blue-700);
  text-decoration: none;
}
```

```
.insecure-form .icon {
  background-image: -webkit-image-set(
    url(
SR0IArs4c6QAAB21JREFUeAHtXF1IHFcU9ie2bovECqWxeWyLjRH60BYpKZHYpoFC
U60/
xKct5ME3QaSpT6WUPEICEXyTUplojfgTUwshNpBgqZVQ86hGktdgSsFGQqr1t9+nd2
WZPefO7LjrjYzcJmZc8495zvf3LI3Zu
+dzcoKt5CBkIGQgZCBkIFMZSB7r4G3tLS8sLCw8D7ivo1Ssrn5WYL9AZSC7OzsAuyzl
HuCHcsjyOawZ7lbVFT0W09Pzz843rNtTwhqaGh4ZXV1tQFZfYZSDgKe85MhyFpBvTso
V/Py8q5g
+9OPn0TqpJSgurq6CpBxFuUEQO1LBJgH2zUQdgPlwuDg4LgHe18mKSGovr7+2Pr6+j
kgOulLveKVJnJzc78eGBi4nXhVe42kEtTY2Fi8vLz8HVRMKXvY1GjRmvrz8/Pb+/
r65pMVIWkEodV8vLGx8SPI2Z8sch78gKTFnJyc02hN1/3Ud9ZJCKg1tbVfwnEnyMlxBp
DOkcQybG9ifwv6OezvRyKRv5eWljhyZeG4AMcvweYNnHKkq4TNcezzqXfbYLSBm46h
oaELbrZu+l0R1Nra+vz8/HwPgH/uFgj6xwA
+inINt8Evvb29Tz3U2TFpamp6EbfvR4hVhXISisldpXKAWJeLi4tburu7/1VMXMW
+Ccll9TKA/oTyni0KQC5B34V9J0abRZutVx1i70fcDti3YR
+x1UPcSZRPEfsvm52m80WQaTm3beQA1Dr0F9EffANwDzUAu5GDqIPo975FrGbEytV
8QT
+JlnTMT0vyRRD6nEsAZLutOlpuDw8P86Eu5VtNTU05goygFGvBQNJI9ElfaHpNrrKuV
```

WCHDHLOanoAmUKr  
+QBgZjWbZMtnZ2cflpWV9cPvUZRXF9vHT58+OnMzMzvil4UJ0QQh3KQ8wM8iS0P5P  
SjVOGWWWhCjpVCIXJ+AgD6EeA2ITAoFbB  
+CyKnp6en7kl6SiYIKhuYhcBYEic85JAethu9bad/Qyq8Ap/  
iwCpyLGEUPeX2Y9PTcwozNE7JGzhQCn0k7MwYAsaBMSXh4gZmLpJNknIqQebe6JT  
mAbB59zru7GanQyW5KvtHJe8In1TUj3B/  
QiR033t0qvby7eWpB5sUzDgeu0jqE1bshJ85pkgQGU7XBG0dVy8lp6EoQrkQFKolv5Wi  
uF/dqKHcC93JObMS02B4xuSnqbbErQQggDum4Mkt8CLR6D4CSGIIvgqLIFmtrJYi/  
BMIJf+yStq4g3lpOoAZjI1POc+bGHCVdVGyIaGVI5TQMpV8C+eLZGXUS9L3B+lJauc/  
8FCyotkVS8jvGcFwNInfOoweQj  
+LKJOXFKz53M1pFMdn2xlpno1Hklr0e8XdysYXRp9qCOPsAPd9x4jYQdC1OGHCBBX  
O5yVXMQCWIUzNgPG72AYGW  
+XuO6C3AQmImdidE5mimoZyqrXOVIG5bxW3weHNRH/  
sinOSBgExE7sSWsyVtjaCSiRnuAraE7VkhiiZBbuYK8GrBIFtsRKC3AtU1gmA0bBrudK1  
bRQ7oMR  
+oMh9i1PxLqaA0bBrueotCAG25smdgTj74JRlyrkFu5gr81JvMTRHsVJ0aiZTSInFqWHX  
crUSFOv4WT5WWxA6rq1JPCc5nNRzyLIXMOu5cq8VIKgeWnijGemEOLEacEu5sr6Nol  
eOQPwHGxzOjgjNwt2MVcmqRKEjmtOYUF8PIJsgyYWsVty1QICZiJBuAqVQcvaKx4Ldj  
FX+IVbEHR3pcBg  
+zgXEki6IMulmdgVjGKutFUJ4oJJOFxxOsRVyOcqC6c86OdmZUjc8hnmyFw1/  
CpBZjWpOLcOkqo0h0GVWzDfsa2cVQkyiV6VEkawK5gRECCrJft0y4iVmBUcYo5RWyt  
BXGoLw7Woccy  
+EAE7Ys4DfWiwFgog10yOgmpbZCWI65Bxj44ptdtwZQ4qusCIDcY2CRByu  
+G21tpKEJ3CyXnJOa5KhluxJF2QZMRlRlIm5Oa6htGVIMwljMP5hBKg2SxektRplxEbS  
GhWgEyY3BT1ttiVIJpxkbbkBVeG64tGgnirGUwjBmMcfC0np6Hn1RMua264/  
OUorog4xesMmupzkBMBMb  
+ivCPFAIbPa5k8tSAGwbRJOxyLk4UEgsKVZ4HYiMVCDhdQtXsF6rkF0aFZTf8zgove8s  
qgnEIXSzlth+SckggAtg0sZvgkkVX4Ca1R5Nq+0tJSfq  
+lvWpwbeAJrBW8zjWDEshUydjngJgxFA0bR+SvcPEuJYIhoRYUdYz  
+6JIZBizeKIEitD2X9+NqTGp6yluhn8Aw+70ZTSym/lX0zRiMxZiaJ2IIZk1vk/  
tqQXQlCOGnCDZmqQs/ZnFjyOjRJ/n+HArNn1PZDzipF5234uyD  
+YH9dXS6b6Jk5udQsfz9Xz  
+o89VJxxITPeazBR7ADqFF8JuJtGyMTQyJPOe4AfXdSdscm4Xn52AjLh  
+21fWpy4yPep3JYaSrQP+Rys/  
Cx9BqzuPhb9wZO1nnKWIBTnDhHws4GbGcZ9pfU1hSCVUhAyEDIQMhAyEDAWfgP5  
qNU5RLQmxEAAAAAEIFTkSuQmCC) 1x,  
url(  
SR0IArs4c6QAAEp1JREFUeAHtnVuMFkUWx2dgRlBhvUxQSZTsw25wAUPiNQTRgFkv  
8YIbZhBcB8hK2NVkXnxRY0xMDFffJkHsyxskBFRGIJ4iWjioLJqdL3EENFZ35AELxnR  
HZFFBtjff+gePsbv0qe6+vv6+6Y66XR39aIT5/  
zPv6urq6q7m5rCEhAICAQEAgIBgYBAQCAgEBAICAQEAgIBgYBAQCAgEBAICAQEAg  
IBgYBAQCAgEBBoTASaG9Ot8l6tWLFi4sGDB3+P1HStx44d0/  
a85ubmyWwnHz9+fHgbHTdxPEj6IMfD2+j423HjxvWTPryeeeaZX65fv/5/HI  
+pZUwQ6l477vjD0NDQAgiwgOBfynYa23E+l43OY+jcy/Zjtn0tLS19zz///Oc  
+y8ijroYkUEdHxxSCuBDAF7DOZ/

+CWoAPmb6m3J2sfexv37Jly3e1sCPLMhuGQF1dXRP2799/  
G2TpBLCbWFuyBM5B9xB5XoVIPVOnTn2xu7v7sIOO3GWpewJR21xJG  
+ZukF3MenbuEC5u0A8kb6YNtY5a6YPilvWRWrcEWrx48Xyl8xA1znX1AXVxK6mR3oBl  
qzdV3qxbXd0tdUcgapybly2IM6fu0C5jMER6j3U1NdlrZcRyd6puCARx5kCabtbLcoeiR4  
Mg0UesXRDpPY9qM1OVewltW7asjT6bJ0DgLGy5t9dTpl6j55/0Ld2/  
YcOGAU86M1GT24BQ0zS3t7evxOvHWNsy8T7/  
SkWeB3t7e9dSK4IUuVtySSBuV9NoID8LWnNzh1htDHqHhvd3Nb21qb40qV67Y0tXUz  
yMzxd3Urt8wk5AnlOwjZXmAibk0n52MtNDbRq1arWgYGBx4HlvmpAwy3hJ8rpJzD98Zg  
W+1+RPjh+/PjB0047bfDQoUMa+2o6/FTTJ//yyy+Tjx49OjxOhsxFJA+PobE/  
PJ5G3kmSrcLyZFtb2wNr1qw5UoWyKhaRCwltWbLklsaqthCEqypa7CggwqD/  
bbZ9bPsuueSSTx955JFjjupOyYaecbt3756Nbo21acztGraZEQR97zPW1vHcc899dYohN  
TioOYFo78ygvfMavl+Ygf8aQe  
+IhumZMWPGLGkT4YTMF8pp2bNnzzz86oRI7RS0oX3fyz78uoF20R7fii36akqqgG/  
nZUA+12J0JVII8zrr08htA+BDleSzPM+YwDBw7cjo/LWa/3WRY  
+fs96Sy37jGpGIMhzm1foZgA9wweoAKnb0VbaL6uZRVGpD52+dTCtZDbtqlfQuwgy  
+XqA+ZmaaDEkqkkPdk0IRP/  
OnwFwPUCmHjGPIPNMa2vrY5s2bfrCd9Cz0Ld06dKLjxw58iC67/  
JEpCFItBwSqeujqkvVCRTVPC/gpQ/  
yfEgA7tm6deuHVUXNU2GLFi26nAvgKXy43INKkej2atdEvqrRRP6rzRPdtIKRB9APANa  
9s2bNuqpeySPAZL8kC/yKRGlpYVahK0wLi3i/  
0zVaiAcm8GVtos1VYMZoHfQL7O8p6fnW/  
9w1E5jZ2fnefQ7PQ0+N6axAnzUsJ5HTVSVp7OqEEj9PNzz3wWYNI/  
qqqlfZt7MEwCUy3GhNIFXXsjTTG/z/  
dQkj3KYppbeN3HixDkbN27cl9amSvkzv4Wph1mdhBiShjqz85jPVfV4o5JHgZJv8IG  
+cpgm+BcePny4V9hLb5ZL5gTS8ARXVpoe5k8B9AqA/  
VeWQORJt3yVz9jk3B0hzKOhoUxdy/QWpsE/+j1edPWAK/lt1oUA  
+qOrjnrOR7vxLliwnfVaVz/oF7uN2/5Lrvkr5cusBsL5adzL11cyoNR5iLNt0qRJN45V8ggX  
+S4MhEUpnCqIKwaKRSU51/  
OZEIgrphnDn2Xr9MQlwFg7xuKbnqMDKQyEhSuJFIMoFpncbTihUDST0Gk  
+D0C9xVWnyVNHR4M5Vo  
+FhTARNo4Yzl1i4pi9dDbvrlzmMPdTpMs0VDWYrx3Lt63SoWpqUpul2kQkml1OrsS5Ae  
ZYT/c9x9p7DRRNghChjx7Vx3Sbp0TgR5J1YQkjElwe8eOXE0b0+djxWgNxxhWio4h0Ms  
+pVJ6H6eWr2qM64IKlzkmlq48+4jWsA5yvBuedHLQYIR4H57ng7O2Vla81EA22bhwy  
A4tTD9eSPMYg1FxcWAKzB0Oaoxg5ZC2exRuBuCr0xuhlxYspnUrDcleGJ0pLhDPFEli  
GdHYUO1cuTTFSrMrJWM55lxCGaaKUaYE8BzQwyZ0+zAV0qDCwizCzyK7xKrUjB6l  
RA9zvoGj3kaASA81Gij6qWaziJd2AIDq27FSjGz5ism74VANOjMTuD4hzNnzvx7MaNC  
WnlEhKGWtJ7jhKRLzIqVkJZpA3E  
+vhNGmT6zgsD4Hd4+v12qKOTZW0oShsBSmFp8VM8XOkqeYbGoCYcjKYoorpD1Tzz  
MJk/hW9dMRIs9YC3aM3SnFpCKQPiuHER2naKxwoCtFE  
+AriIXTRgSEqUMt1KEYGos6RTwVgfrNRqRZPyu3tV7enjgFqZwfRjhuNZp5dhRDY7aT  
4qklhJplJ1UI29N7W8kkg5QVARdsuYPoo6TOizOBalDpU7qmCeBUsa/n9aU/  
ZwRzIFHYCmOjStcplsY8I+LWsZSRjJBnlQem/  
Dj39liCnO3UcmzLJxTCmNhYXqFuiWK51sUO5xqlwhYYCxxE3nlmnbGssSwujlW1ZbH  
GckR3GgKZejK5MnoZBKzphw5GvG7gHWEsrI0ummmJZqNuJQNwz9ZKg6fcBjB73FBYc  
9rNDwlq1Yqn/ibhY5EQgusFNjOWK  
+Enf53ExMOSxlyCshbkp35GY5GPZZ0lhHGmwmD429X6uFPs2FjeCmthbsHAGtNYtx

OBM07SWEGSLcb1JZELMv4QsGJujWlsqZIA  
+lkbpneM8K4QKAY8SptrZgrpoqt1TwzgfSnP4xLnA/DftIHLa2GBfI0CAhzYZ9Ui2la/  
cUxaZZhucREKNCqz9palv4wbcMClx/  
ZCHO9XmVZrLFtypxAMNvqhMXhIFsGAQfssycQj/CmQuiTCAQqE  
+QsT1mxt8ZWtpvGspSB+  
+r5MFu7SZe6IFA9vReWFHjkTNgrtgbdw6lutzDTR7Mh21dWo4K8HwQcsDfFVla6EMj0  
CX9YbR3Y84Ne0KK7hRV7U2ydCASrTSxlkpPViRB6TwhYsbfG1oIAZDIRSH  
+98YRHUGNEwAF7U2xljvkWRrVoKiT  
+ZZLR9yDuAQEr9tbYykQzgTz4FVQ0EAJmAnGfNN2S9LO2BsKrrlyxYm  
+NrcAwE4g8JgLPt391hXoDGeuAvSm2gspMIoujX4T2UAxqStXrNhbY  
+tEIDKZWOrYAfhXUcqsQ7Ym2LrSqDEUwRUAKzWD2rDUgMErNhXpQ1Eld8YsTAN  
vhp1B/HyCFixN/  
8BydwGqsYlb3IMwtmkCFhH162xIR1mApHHOsJrvQqS4hPkKiDALcyKvSm2Kj5zAIHGd  
GbHuZRTAZ5wuhwCEeb5lxBfO/  
8SZh8rZ3zhOdpMk3bv3j27MC3sZ4+AMBf2SUtSTBXbpPKxnLlm0M8/  
MGxvrCDJFuMWJJEMLMv4QsGKumLr83MZMILmIcR9bXmw4QCALYB5krZhbYxqb6EQ  
gjDO954Vx13BPNk  
+fjY0MWxsCwlqYW3JZYxrrdiJQS0uLiUAYN2nPnj3z4kLDNIsEhLUwt5RijWms24IAfAnrc  
xj+dawkyZY+iVSfUktSRpA5gYAVa8VSMXXBz4IAUUH6W0zihSuinc/  
CnJ44QxB0QkAYC2tjZIMsC3WnlZDpNkahGpX/  
U2HhYT8TBISxdQaENZyjhjsTiGpvO1qGRjQl2OHKWJ5ALlikQACMVxizD0WxNGY7le  
5MID6l9h0qXrWUinPX8yWs0KloAc0gK2zB+I+GLBJ9NYqIMdsJcWcCKTvMNX+2jklO5h  
+zOHk2BjO5YOsSw0JoUxFo6tSpL6Lsh0KFCfYXLV269OIEckHEgECE6SJDFon  
+EMXQmO2keCoCdXd3H0bV5pPqKu9RxY47cuTlg5Ulg4QFAWEqbC15kN0cxdCY7aS  
4tcCTOaM95pCs+1Vi5YS7+JjB5ZXFGkQSBCIs70oiWyjjGLtCFU7TOU5RQAPsA  
+6jb5ySWOFavwp5ngrTPCoAleC0MBSW1tpHMVPsEhRRViR1DSTtMNN8AxUcvvyzzz  
77a1nrwsmKCAhDYVIRcJSAS8xGqRg+9Elg/iC8E0a/V6yAcmk4vrqzs/  
O8cjLhXGkEhJ0wLC1R/lxipZgVP2tL9UlgFYIRZkdw/  
hze39bPQZptZgdpYRZhd44VDZdYISrDG4G4n76CYR+VKqhUOkDcyB  
+E7y91PqQXR0CYCbviZ0unKkaKVWkJ2xlvBFKxGNfF5rjNhKYmRo8fZRDwamu  
+sSovrlSZg//Hoxg5ZC2exfutg0fKtRR1d/Hiyqbuo2F3BVeHaZpIWY0NeBLyXAB5/  
o1rFzq4t47/oq10yFcyi9caSKUwMVu3o4GSJZY  
+cSHA7ACgs0qLjO0zkwYYgYILeQai2HgF0TuBNmzYIPK49jRrMHC7yyf3vaKSQ2XCR  
Nhgmutg9lNRbLx65/0WJutwtLm9vX0Xu3NdrOU+vY21g9vZUZf8jZaHmmc8mG5h1Vwfl  
+Wd3t7eeWBqbp9Wksx7DaQCZSjtmTvZfl/  
JgGLnBZQACzVRU1NU8ziTRzGIYuGdPMOxLhZAX2k8at7KFAON2DstOP8W60Jqoh  
+dFNR5JrV5uJC2s17r6gpfar2NTsOXXPNXyje+kkCa83Sz/4e/  
5/0GHXMc9fwW8G6aNWvWC7xpYPqsjGN5uckGefS0pTHGq1IY9SS3ru4U  
+StmzeQWVlhqW1vbA9Qi7xemGfdn67EVQMdMP5F8lc/  
g5NpgVjPifWFvxNosnkkjerQVS5YsuYj5Ku  
+S7vL4Gasb4l7+MNXXE4CTyf08LqhWW2rbZvUwQx51EqZ5EXPfxIkT52zcuHFf1r5Uh  
UBygqtKf3rexXpuGqcgzw6+Prq8p6fH/  
DGkNOVmnVcDo9HYInl4otA28PmedR7txj2F6VntZ9oGKjSaNsx3M2fOFIGWkt5aeM64/  
zv+MLwSXf/  
lav34zTffrOvaSPN5pkyZ8jdq6G1gc4kRi9HiP1NL3wh5Phl9lqvjqTVAsQPURDdTRb/  
AcZooQlandsK9dM9/GCfU01YzCaktNBnMPJ+niJ

+6xd8OebwNIBYp41dJVSeQLIBEd0Kip9INTSICcAw9z7S2tj62adOmL6Q/74smwEfzww  
+CPD4eZESe5ZDn2Wr7XhMCycmoJtKE/DN8OB0RaSv9Hqt5z/  
tTHzp969B7W9GrN4s8EUcm6ra1uNo1T4xNzQgkAyDRHIB8mTVVwzp2Jt5CptdZVcNt  
A9hDcXottvio7wGoZ3056/U+bcBHNZhvwUfzbFBfdtSUQHICgGdwO3uN3TSP  
+KXwGATgXq7QHjo0d9FgHSol6DOdcl0iRX86oQ07eie7FN/  
pEvTX26APFV52iplf80JJMPUT8STlcZ70vS6lvJxOB0i/YT  
+t9n2se3Tf9UJtNpPqRc9SembhOhegO4FbK9ha/o+j8UI9L8/  
YcKE9mr081SyKxcEkpGrVq1qHRgYeJzd+yoZ7eM8QdDQSD+B7udK7o/2vyJ9UH/608/  
a4v9t6a83+nEJ7ZfJyE9G5iLkp1PDTGdfX0KdniVh0F  
+4PKke5jVr1hwpTKzVfm4IFAOGAVgCs56AeG0XxfrdQtRNAq  
+IsuBURdsckcgOUG7aBok0iOp03wiFyBynucdyHMn7Z29ebMzlwQSSNRampS2kt3HW  
NuUNgaX4dmdjKivpQbKZY+7j06sTOlqwOhh/gfzeNXGWMeaSwAzcf6Er  
+vkuzDIK3nke25roNGBifqMuqmZLht9rpGOlctHrF217Nux4Fk3Blqdgkg3Q6KHWF0nqc  
WqcrWFNO+xroY4VR3LSgtC3REodpintfk0tEWk6+K0etxCmjdoIK/  
29a56tTGoWwLFQFEjXQmJVrJ2kHZ2nJ7z7Q8QZwvrWmqc1J9YqaWvdU+gGLyurq4J  
+/fvv43jZZBJk7JSj/THuj1t9TVUvRS4QZ+VS/  
tIME82pVbTMAQqRIJaaQokWkjaAtb57F9QeL5a  
+xBGr2nvZO1jfzu1jb5s21BLQxJodIQglAZs5xNEjVVdynYaW69dGOg8hs69bD9m20e7Z  
ieEqelA52gcsjgeEwQaDZxe1jt48ODvSR8ex4JcGtM6n2ONmk  
+CANpqzGt4FJ3jQY41sq+txtAGSfsGkgyPoXHcT5/  
Nly7/2yJvWAICAYGAQEAgIBAQCAGBAICAYGAQEAgIBAQCAGBAICAYGAQEAgIB  
AQCAgEBAICAYEclvB/Q079+h6myXwAAAAASUVORK5CYII=) 2x);  
}

@media (prefers-color-scheme: dark) {

.insecure-form .icon {  
filter: invert(1);  
}

}

</style>

<style>/\* Copyright 2019 The Chromium Authors. All rights reserved.

\* Use of this source code is governed by a BSD-style license that can be

\* found in the LICENSE file. \*/

.lookalike-url #main-content a {  
color: var(--google-blue-700);  
text-decoration: none;  
}

.lookalike-url .icon {  
background-image: -webkit-image-set(  
url(  
SR0IArs4c6QAAB21JREFUeAHtXF1IHFcU9ie2bovECqWxeWyLjRH60BYpKZHYpoFC  
U60/  
xKCt5ME3QaSpT6WUPEICEXyTUplojfgTUwshNpBgqZVQ86hGktdgSsFGQqr1t9+nd2



WZPefO7LjrjYzcJmZc8495zvf3LI3Zu  
+dzcoKt5CBklGQgZCBklFMZSB7r4G3tLS8sLCw8D7ivo1Ssrn5WYL9AZSC7OzsAuyzl  
HuCHcsjyOawZ7lbVFT0W09Pzz843rNtTwhqaGh4ZXV1tQFZfYZSDgKe85MhyFpBvTso  
V/Py8q5g  
+9OPn0TqpJSgurq6CpBxFuUEQO1LBJgH2zUQdgPlwuDg4LgHe18mKSGovr7+2Pr6+j  
kgOulLveKVJnJzc78eGBi4nXhVe42kEtTY2Fi8vLz8HvRmKXvY1GjRmvrz8/Pb+/  
r65pMVIWkEodV8vLGx8SPI2Z8scH78gKTFnJyc02hN1/3Ud9ZJCkG1tbVfwnEnyMlxBp  
DOkcQybG9ifwv6OezvRyKRv5eWljhyZeG4AMcvweYNnHKkq4TNcezzqXfbYLSBm46h  
oaELbrZu+l0R1Nra+vz8/HwPgH/uFgj6xwA  
+inlNt8Evvb29Tz3U2TFpamp6EbfvR4hVhXISisldpXKAWJeLi4tburu7/1VMXMW  
+Ccll9TKA/oTyni0KQC5B34V9J0abRZutVx1i70fcDti3YR  
+x1UPcSZRPEfsvm52m80WQaTm3beQA1Dr0F9EffANwDzUAu5GDqIPo975FrGbEytV  
8QT  
+JlnTMT0vyRRD6nEsAZLutOlPUDw8P86Eu5VtNTU05goygFGvBQNJI9ElfaHpNrrKuV  
WCHDHLOanoAmUKr  
+QBgzJWbZMtnZ2cflpWV9cPvUZRXFf9vHT58+OnMzMzvil4UJ0QQh3KQ8wM8iS0P5P  
SjVOGWWhCjpVCIXJ+AgD6EeA2lTAoFbB  
+CyKnp6en7kl6SiYIKhuYhcBYEic85JAethu9bad/Qyq8Ap/  
iwCpyLGEUPeX2Y9PTcwozNE7JGzhQCn0k7MwYAsaBMSXh4gZmLpJNknIqQebe6JT  
mAbB59zru7GanQyW5KvtHJe8ln1TUj3B/  
QiR033t0qvby7eWpB5sUzDgeu0jqE1bshJ85pkgQGU7XBG0dVy8lp6EoQrkQFKolv5Wi  
uF/dqKHcC93JObMSO2B4xuSnqbbErQQggDum4Mkt8CLR6D4CSGIIVgqLIFmtrJYi/  
BmIJf+yStq4g3lpOoAZjl1POc+bGHCvdVGyLaGVI5TQMpV8C+eLZGXUS9L3B+lJauc/  
8FCyotkVS8jvGcFwNlnfOoweQj  
+LKJOXfKz53M1pFMdn2xlpno1Hklr0e8XdysYXRp9qCOPsAPd9x4jYQdC1OGHCBBX  
O5yVXMQCWIUzNgPG72AYGW  
+XuO6C3AQmImdidE5mimoZyqrXOVIGg5bxW3weHNRH/  
sinOSBgExE7sSWsyVtjaCSiRnuAraE7VkhiiZBbuYK8GrBIFtsRKC3AtU1gmA0bBrudK1  
bRQ7oMR  
+oMh9i1PxLqaA0bBrueotCAG25smdgTj74JRlyrkFu5gr81JvMTRHsVJ0aiZTSInFqWHX  
crUSFOv4WT5WWxA6rq1JPCc5nNRzyjLIXMOu5cq8VIKGEwnijGemEOLEacEu5sr6Nol  
eOQPwHGxzOjgJNwt2MVcmqRKEjmtOYUF8PIJsgyYWsVty1QICZiJBuAqVQcvaKx4Ldj  
FX+IVbEHR3pcBg  
+zgXEki6IMulmdgVjGKutFUJ4oJJOFxxOsRVyOcqC6c86OdmZUjc8hnmyFw1/  
CpBZjWpOLcOkqo0h0GVWzDfsa2cVQkyiV6VEkaw5gRECCrJft0y4iVmBUcYo5RWyt  
BXGoLw7Woccy  
+EAE7Ys4DfWiwFgog10yOgmpbZCWI65Bxj44ptdtwZQ4qusCIDcY2CRByu  
+G21tpKEJ3CyXnJOa5KhluxJF2QZMRlrBlm5Oa6htGVIMwljMP5hBKg2SxektRplxEbS  
GhWgEyY3BT1ttiVIJpxkbbkBVeG64tGgnirGUwjBmMcfC0np6Hn1RMua264/  
OUorog4xesMmupzkBMBMb  
+ivCPFAIbPa5k8tSAGwbRJOxyLk4UEgSKVZ4HYiMVCDhdQtXsF6rkF0aFZTf8zgoveE8s  
qgnEIXSzlth+SckggAtg0sZvgkkVX4Ca1R5Nq+0tJSfq  
+lvWpwbeAJrBW8zjWDEshUydjngJgxFA0bR+SvcPEuJYIhoRYUdYz  
+6JIZBizeKIEitD2X9+NqTGp6yluhn8Aw+70ZTSym/lX0zRiMxZiaJ2IIZk1vk/  
tqQQXlC0GnCDZmqQs/ZnFjyOjRJ/n+HArNn1PZDzipF5234uyD  
+YH9dXS6b6Jk5udQsfz9Xz

+o89VJxxITPeazBR7ADqFF8JuJtGyMTQyJPOe4AfXdSdscm4Xn52AjLh  
+21fWpy4yPep3JYaSrQP+Rys/  
Cx9BqzuPhb9wZO1nnKWIBTnDhHws4GbGcZ9pfU1hSCVUhAyEDIQMhAyEDAWfgP5  
qNU5RLQmxEAAAAAEIFTkSuQmCC) 1x,  
url(  
SR0IArs4c6QAAEp1JREFUeAHtnVuMFkUWx2dgRIBhvUxQSZTsw25wAUPiNQTRgFkv  
8YIbZhBcB8hK2NVkXnxRY0xMDFffJkHsyxskBFRGIJ4iWjioLJqdL3EENFZ35AELxnR  
HZFFBtjff+gePsbv0qe6+vv6+6Y66XR39aIT5/  
zPv6urq6q7m5rCEhAlCAQEAglBgYBAQCAgEBAICAQEAglBgYBAQCAgEBAICAQEA  
glBgYBAQCAgEBBoTASaG9Ot8l6tWLFi4sGDB3+P1HStx44d0/  
a85ubmyWwnHz9+fHgbHTdxPEj6IMfD2+j423HjxvWTPryeeeaZX65fv/5/HI  
+pZUwQ6l477vjD0NDQAgiwgOBfynYa23E+l43OY+jcy/Zjtn0tLS19zz///Oc  
+y8ijroYkUEdHxxSCuBDAF7DOZ/  
+CWoAPmb6m3J2sfexv37Jly3e1sCPLMhuGQF1dXRP2799/  
G2TpBLCbWFuyBM5B9xB5XoVIPVOnTn2xu7v7sIOO3GWpewJR21xJG  
+ZukF3MenbuEC5u0A8kb6YNtY5a6YPilvWRWrcEWrx48Xyl8xA1znX1AXVxK6mR3oBl  
qzdv3qxbXd0tdUcgapybIY2IM6fu0C5jMER6j3U1NdlrZcRyd6puCARx5kCabtbLcoeiR4  
Mg0UesXRDpPY9qM1OVewltW7asjT6bJ0DgLG6y5t9dTpl6j55/0Ld2/  
YcOGAU86M1GT24BQ0zS3t7evxOvHWNsy8T7/  
SkWeB3t7e9dSK4IUuVtySSBuV9NoID8LWnNzh1htDHqHhvd3Nb21qb40qV67Y0tXUz  
yMzxd3Urt8wk5AnlOwjZXmAibk0n52MtNDbRq1arWgYGBx4HlvmpAwy3hJ8rpJzD98Zg  
W+1+RPjh+/PjB0047bfDQoUMa+2o6/fTTJ//yyy+Tjx49OjxOhsxFJA+PobE/  
PJ5G3kmSrcLyZFtb2wNr1qw5UoWyKhaRCwltWbLklsaqthCEqypa7CggwqD/  
bbZ9bPsuueSSTx955JFjjupOyYaecbt3756Nbo21acztGraZEQR97zPW1vHcc899dYohN  
TioOYFo78ygvfMavl+Ygf8aQe  
+lhumZMWPGlgKt4YTMF8pp2bNnzzz86oRI7RSo0X3fyz78uoF20R7fii36akqggG/  
nZUA+12J0JVII8zrr08htA+BDleSzPM+t+YwDBw7cjo/LWa/3WRY  
+fs96Sy37jGpGIMhzM1foZgA9wweoAKnb0VbaL6uZRvGpD52+dTctZDbtqlfQuwgy  
+XqA+ZmaaDEkqkkPdk0IRP/  
OnwFwPUCmHjGPiPNMa2vrY5s2bfrCd9Cz0Ld06dKLjxw58iC67/  
JEpCFItBwSqeujqkvVCRTVPC/gpQ/  
yfEgA7tm6deuHVUXNU2GLFi26nAvgKXy43INKkej2atdEvqrRRP6rzRPdtIKRB9APANa  
9s2bNuqpeySPAZLt8kC/yKRGlpYVahK0wLi3i/  
0zVaiAcm8GVtos1VYMZoHfQL7O8p6fnW/  
9w1E5jZ2fnefQ7PQ0+N6axAnzUsJ5HTVSVp7OqEEj9PNzz3wWYNI/  
qqqlfZt7MEwCUy3GhNIFXXsjTTG/z/  
dQkj3KYppbeN3HixDkbN27cl9amSvkzv4Wph1mdhBiShjzq85jPVfV4o5JHgZJv8IG  
+cpgm+BcePny4V9hLb5ZL5gTS8ARXVpoe5k8B9AqA/  
VeWQORJt3yVz9jk3B0hzKOhoUxdy/QWpsE/+j1edPWAK/lt1oUA  
+qOrjnrOR7vxLliwnfVaVz/of7uN2/5Lrvkr5cusBsL5adzL11cyoNR5iLNt0qRJN45V8ggX  
+S4MhEUpnCqIKwaKRSU51/  
OZEIgrphnDn2Xr9MQlwFg7xuKbnqMDKQyEhSuJFIMoFpnbcTihUDST0Gk  
+D0C9xVWnyVNHR4M5Vo  
+FhTARNo4Yzl1i4pi9dDbvrlzmMPdTpMs0VDWYrx3Lt63SoWpqUpul2kQkml1OrsS5Ae  
ZYT/c9x9p7DRRNghchjx7Vx3Sbp0TgR5J1YQkjElwe8eOXE0b0+djxWgNxxhWio4h0Ms

+pVJ6H6eWr2qM64IKlzkmeIq48+4jWsA5yvBuedHLQYIR4H57ng7O2Vla81EA22bhwy  
A4tTD9eSPMYg1FxcWAKzB0Oaoxg5ZC2exRuBuCr0xuhlxYspnUrDcleGJ0pLhDPFEli  
GdHYUO1cuTTFSrMrJWM55lxCGaaKUaYE8BzQwytZ0+zAV0qDCwizCzjyK7xKrUjB6l  
RA9zvoGj3kaASA81Gij6qWAZiJd2AIDq27FSjGz5ism74VANOjMTuD4hzNnzvx7MaNC  
WnlEhKGwTJ7jhKRLzIqVkZpA3E  
+vhNGmT6zgsD4Hd4+v12qKOTZW0oShsBSmFp8VM8XOkqeYbGoCYcjKYoorpD1Tzz  
MJK/hW9dMRIs9YC3aM3SnFpCKQPiuHER2naKxwoCtFE  
+AriIXTRgSEqUMt1KEYGos6RTwVgfRNQRZPYu3tV7enjgFqZwfRjhuNZp5dhRDY7aT  
4qklhJplJ1UI29N7W8kkg5QVARdsuYPoo6TOizOBalDpU7qmCeBUsa/n9aU/  
ZwRzIFHYCmOjSTcplsY8l+LWsZSRjJBnlQem/  
Dj39liCnO3UcmzLJxTCmNhYXqFuiWK51sUO5xqlwhYYCxxE3nlmnbgssSwujlW1ZbH  
GckR3GgKZejK5MnoZBKzphw5GvG7gHWEsrl0ummJZqNuJQNwz9ZKg6fcBjB73FBYc  
9rNDwlq1Yqn/ibhY5EQgusFNjOWK  
+Enf53ExMOSxlyCshbkIp35GY5GPZZ0lhHGmwmD429X6uFPs2FjeCmthbsHAGtNYtx  
OBMO7SWEGSLcb1JZELMv4QsGJujWlsqZIA  
+lkbpneM8K4QKAY8SpTrZgrpoqt1TwzgfSnP4xLnA/DftIHLA2GBfl0CAhzYZ9Ui2la/  
cUxaZZhucREKNCqz9palv4wbcMClx/  
ZCHO9XmVZrLftypxAMNvqhMXhIFsGAQfssycQj/CmQuiTCAQqE  
+QsT1mxt8ZWtpvGspSB+  
+r5MFu7SZe6IFA9vReWFHjkTNgrtgbdw6lutzDTR7Mh21dWo4K8HwQcsDfFVla6EMj0  
CX9YbR3Y84Ne0KK7hRV7U2ydCASrTSxlkpPViRB6TwhYsbfG1olAZDIRSH  
+98YRHUGNEwAF7U2xljvkWRrVoKiT  
+ZZLR9yDuAQEr9tbYykQzgTz4FVQ0EAJmAnGfNN2S9LO2BsKrrlyxYm  
+NrcAwE4g8JgLPt391hXoDGeuAvSm2gspMIOUjoX4T2UAXqStXrNhBY  
+tEIDKZWOrYAfhXUcqsQ7Ym2LrSqDEUwRUAKzWD2rDUgMErNhXpQ1Eld8YsTAN  
vhp1B/HyCFixN/  
8BydwGqsYlb3IMwtmkCFhH162xIR1mApHHOsJrvQqS4hPkKiDALcyKvSm2Kj5zAlHGd  
GbHuZRTAZ5wuhwCEeb5lxBfO/  
8SZh8rZ3zhOdpMk3bv3j27MC3sZ4+AMBf2SutSTBXbpPKxnLIm0M8/  
MGxvrCDJFuMWJJELMv4QsGKumLr83MZMILmlcR9bXMW4QCALYB5krZhbYxqb6EQ  
gjDO954Vx13BPNk  
+fjY0MWxsCwlqYW3JZYxrrdiJQS0uLiUAYN2nPnj3z4kLDNlsEhLUwt5RijWms24IAfAnrc  
xj+dawkyZY+iVSfUktSRpA5gYAVa8VSMXXBz4IAUUh6W0zihSuinc/  
CnJ44QxB0QkAYC2tjZIMsC3WnlZDpNkahGpX/  
U2HhYT8TBISxdQaENZYjhjsTiGpvO1qGRjQl2OHKWJ5ALlikQACMVxizD0WxNGY7le  
5MID6l9h0qXrWUinPX8yWs0KloAc0gK2zB+I+GLBJ9NYqlMdsJcWcCKTvMNX+2jklO5h  
+zOHk2BjO5YOSw0JoUxFo6tSpL6Lsh0KFCfYXLV269OIEckHEgECE6SJDFon  
+EMXQmO2keCoCdXd3H0bV5pPqKu9RxY47cuTlg5Ulg4QFAWEqbC15kN0cxdCY7aS  
4tcCTOaM95pCs+1ViYS7+JjB5ZXFgkQSBCIs70oiWyjjGLtCFU7TOU5RQAPsA  
+6jb5ySWOFAVwp5ngrTPCoAleC0MBSW1tpHMVPsEhRRViR1DSTtMNn8AxUcvvyzzz  
77a1nrwsmKCAhDYVIRcJSAS8xGqRg+9Elg/iC8E0a/V6yAcmk4vrqzs/  
O8cjLhXGkEhJ0wLC1R/lxipZgVP2tL9UlgFYIRZkdw/  
hze39bPQZptZgdpYRZhd44VDZdYISrDG4G4n76CYR+VKqhUOkDcyB  
+E7y91PqQXR0CYCbviZ0unKkaKVWkJ2xlvBFKxGNfF5rjNhKYmRo8fZRDwamu  
+sSovrlSZg//Hoxg5ZC2exfutg0fKtRR1d/Hiyqbuo2F3BVeHaZplWY0NeBLyXAB5/  
o1rFzq4t47/oq10yFcyi9caSKUwMVu3o4GSJZY

+cSHA7ACgs0qLjO0zwwYYgYILeQai2HgF0TuBNmzYIPK49jRrMHC7yyf3vaKSQ2XCR  
NhgmUtG9INRbLx65/0WJutwtLm9vX0Xu3NdrOU+vY21g9vZUZf8jZaHmMc8mG5h1Vwfl  
+Wd3t7eeWBqbp9WKSx7DaQCZSjtmTvZfl/  
JgGLnBZQACzVRU1NU8ziTRzGIYuGdPMOxLhZAX2k8at7KFAON2DstOP8W60Jqoh  
+dFNR5JrV5uJC2s17r6gpfar2NTsOXXPNXyje+kkCa83Sz/4e/  
5/0GHXMc9fwW8G6aNWvWC7xpYPqsjGN5uckGefS0pTHGq1IY9SS3ru4U  
+StmzeQWVlhqW1vbA9Qi7xemGfdn67EVQMdMP5F8lc/  
g5NpgVjPifWfVxNosnkkjerQVS5YsuYj5Ku  
+S7vL4Gasb4I7+MNXXE4CTyf08LqhWW2rbZvUwQx51EqZ5EXPfxIkT52zcuHFf1r5Uh  
UBygqtKf3rexXpuGqcgzw6+Prq8p6fH/  
DGkNOVmnVcDo9HYInl4otA28PmedR7txj2F6VntZ9oGKjSaNsx3M2fOFIGWkt5aeM64/  
zv+MLwSXF/  
lav34zTffrOvaSPN5pkyZ8jdg6G1gc4kRi9HiP1NL3wh5Phl9lqvjqtVAsQPURDdTRb/  
AcZoqOlandsK9dM9/GCfU01YzCaktNBnMPJ+niJ  
+6xd8OebwNIBYp41dJVSeQLIBEd0Kip9INTSICcAw9z7S2tj62adOmL6Q/74smwEfzWu  
+CPD4eZESe5ZDn2Wr7XhMCycmoJtKE/DN8OB0RaSv9Hqt5z/  
tTHzp969B7W9GrN4s8EUcm6ra1uNo1T4xNzQgkAyDRHIB8mTVVwzp2Jt5CptdZVcNt  
A9hDcXottvio7wGoZ3056/U+bcBHNZhvwUfzbFBfdtSUQHICgGdwO3uN3TSP  
+KXwGATgXq7QHjo0d9FgHSol6DOdcl0iRX86oQ07eie7FN/  
pEvTX26APFV52iplf80JJMPUT8STlcZ70vS6lvJxOB0i/YT  
+t9n2se3Tf9UJtNpPqRc9SembhOhegO4FbK9ha/o+j8UI9L8/  
YcKE9mr081SyKxcEkpGrVq1qHRgYeJzd+yoZ7eM8QdDQSD+B7udK7o/2vyJ9UH/608/  
a4v9t6a83+nEJ7ZfJyE9G5iLkp1PDTGdfX0KdniVh0F  
+4PKke5jVr1hwpTKzVfm4IFAogAVgCs56AeG0XxfrdQtRNaq  
+IsuBURdsckcgOUG7aBok0iOp03wiFyBynucdyHMn7Z29ebMzlwQSSNRampS2kt3HW  
NuUNgaX4dmdjKivpQbKZY+7j06sTOlqwOhh/gfzeNXGWMeaSwAzcf6Er  
+vkuzDIK3nke25roNGBifqMuqmZLht9rpGOlctHrF217Nux4Fk3Blqdgkg3Q6KHWF0nqc  
WqcrWFNO+xroY4VR3LSgtC3REodpintfk0tEWk6+K0etxcmjdolK/  
29a56tTGoWwLFQFEjXQmJVrJ2kHZ2nJ7z7Q8QZwvrWmqc1J9YqaWvdU+gGLyurq4J  
+/fvv43jZZBJk7JSj/THuj1t9TVUvRS4QZ+VS/  
tlME82pVbTMAQqRIJaaQokWkjaAtb57F9QeL5a  
+xBGr2nvZO1jfzu1jb5s21BLQxJodlQglAZs5xNEjVVdynYaW69dGOg8hs69bD9m20e7Z  
ieEqelA52gcsjgeEwQaDZxe1jt48ODvSR8ex4JcGtM6n2ONmk  
+CANpqzGt4FJ3jQY41sq+txtAGSfsGkgYPoXHcT5/  
Nly7/2yJvWAICAYGAQEAgIBAQCAGBAICAYGAQEAgIBAQCAGBAICAYGAQEAgIB  
AQCAgEBAICAYEcLvB/Q079+h6myXwAAAAASUVORK5CYII=) 2x);  
}

```
@media (prefers-color-scheme: dark) {  
  .lookalike-url .icon {  
    filter: invert(1);  
  }  
}  
</style>
```

<style>/\* Copyright 2017 The Chromium Authors. All rights reserved.

\* Use of this source code is governed by a BSD-style license that can be

\* found in the LICENSE file. \*/

```
body.safe-browsing {
  --google-red-900: rgb(165, 14, 14);
  --google-red-700: rgb(197, 34, 31);
  --google-red-600: rgb(217, 48, 37);
  --google-red-500: rgb(234, 67, 53);
  --google-red-100: rgb(250, 210, 207);
  --google-red-50: rgb(252, 232, 230);
  background-color: var(--google-red-600);
  color: white;
}

.safe-browsing :-webkit-any(
  a, #details, #details-button, #proceed-button, h1, h2, p, .small-link) {
  color: white;
}

.safe-browsing button {
  background: white;
  border-color: white;
  color: var(--google-red-700);
}

.safe-browsing button:active {
  box-shadow: 0 1px 2px 0 rgba(165, 14, 14, .3),
    0 2px 6px 2px rgba(165, 14, 14, .15);
}

.safe-browsing button:hover {
  background: var(--google-red-50);
}

.safe-browsing .secondary-button {
  background-color: var(--google-red-600);
  border-color: var(--google-red-100);
  color: white;
}

.safe-browsing .secondary-button:active {
  box-shadow: 0 2px 3px rgba(0, 0, 0, .5);
}

.safe-browsing .secondary-button:hover {
  background-color: var(--google-red-500);
  border-color: var(--google-red-50);
}
```

}

```
.safe-browsing .error-code {  
  display: none;  
}
```

```
.safe-browsing .icon {  
  background-image: -webkit-image-set(  
    url(  
+VBMVEUAAAD//////////9/f3///9/f36+vr8/Pz7+/v7+/v7+/v7+/v7///  
//////////9/f3///9/f36+vr8/Pz7+/v7+/v7+/v7+/v7///  
r6+vn5+fk5OT5+fn19fX4+Pju7u7v7+/o6Ojx8fH09PTy8vLz8/Pj4+P39/fm5ubp6eni4uL8/  
Pzw8PDt7e329vbs7Ozg4ODh4eHe3t7l5eX6+vrd3d3q6urf39/c3NzbRDf7+/  
vb29vW1tbZ2dk+D9arAAAALXRSTIMA/  
elxBfACHj3YwhYJDyfOtr2WHObgEqpjellgY6c6mZLV3L2qKjOZemN  
+rUv7NpAAAACcUIEQVR4Xu2U53baQBQGAXiMaQZT3Wt6rnql9uKW9v4Pk5XEspG4  
WFmJ/  
GMeYM635cy7/8yePXv23N7uxnNwdHSwE1ENoLYLTz4NkM7vQHqFhKv4nlQCCIIUbF  
EOXHJxPVVYUY3nKSSpKFmIJBqANRdxPlcnTHRyGEPUAga0onuy4CMb1ZM5Bh/  
HmYiiGwhwE81TvIYA18VlokvY4DKKp1SBDSqlCKIyIJT5PU1AafL3FVC4q1uDLdS4+4r  
CXd06bKXO21fKdxcNKFzVzUFQ9KoDJRehr0z0tBgBpcrfVyZavAxM7uo2ABHNO8s2N  
TW4+8pEs4HUMxS+6rYAEZFBYI+Y8FQ3C5hoIImGMNVVvLp4X1FRZ2n0BU035X  
+u7jmgomWv/zzWR23IB7ich/  
b1DBeRQVNNty1FHYLDWVh1TwEXic4gs6105UfPdMrd1/  
HCffrnqTayrYkqP3qmSom7r9q84xyMDLIUx7Mylfn7OvIGmbYyUYmGbmry99UWnYOZ7  
sFc5OHb1b2HLbS9QV1vED3dPX9fQXFuWiEeF9kzpfMcfauZkbyn9+bQTfXwvm4W0iA  
3RNdQUyIV1lekkDOhSx3sdGh17+At0ctA0plDpr/gLqSvSI  
+kXt8mA9DVI20r3hGBM0iOwKmBtZXHOPX79fFXCJ/  
Ure79Gxr40Owuu9hK8LPp1mnR7JmWuv7ZsM+oH3FGZObFoWx3iY/  
UI1J2LVn0b7i6B2RZM1WumwL833MoH3FMQ1hPHIGsadnkv94X3EsctPE4xK8p89FvK  
84ExIRuoUa6Pt98fU1DJUMosj+3/S15OtrGOsB9LnYqm+e5w  
+es4JAhYQDdgAAAABJRu5ErkJggg==) 1x,  
    url(  
BMVEUAAAD//////////6+vr7+/v7+/v7+/v7+/f36+vr7+/v7+/v7/  
v//////////9/  
f3//////////8/Pz///5+fn//////////6+vr//////////8/Pz7+/v7+/v7///k5OT4eHf39/  
e3t7g4OD6+vrbRDf5+fnv7+/x8fH09PTY2Njc3Nz39/fq6ur4+Pjz8/Pn5+f7+/v7/  
vZ2dnW1tbt7e3U1NTV1dXI5eXr6+v19fXy8vLs7Oz29vba2trb29vo6Ojm5ubu7u7d3d38/  
Pzp6enj4+Pi4uLw8PD///  
+BQ30nAAAAQHRSTIMAHhYFDzEJAifiwvb98M76PdjqtoGpZII0c46qj1icTJ1KS42A6W  
Wb2WdadYK1cvCb/cJXV+KA9qioPc5l6Y36uRjySgAABXVJREFUeF7s1kuKwIAUBNDS  
+JtqggkkmODIH4gPRIgD11P730BDI6HV260Z1Wu4ZxUH/  
4tzzjnnnHPOOeecq2tEJdIsEsRkT  
+4RkeGRPA4Rj4YkG0RjfCLJ0xixCPwWEIkd73alwuTMu/
```

MEMTiwc4AeBhd2LgPolfyhhFzLBy3Eplc  
+uE6hteWTLaRGcz5ZjKBU8UUFodmNL24z6KQ0pJCpaaohksxpmifQKGhjlXurSbbZhr  
9qRG81yTYbaJtNuefcsFbTbLNZnwjE7zVINtsybdK4VttreytHdlml/  
zlUvTWjmyzFT9USd7akW02pUm22TV7WOveKttsvV4K1Vtlm12xp5XmrbLNBvYWhG  
+15YK3qjab0STb7Fcv5tmcOBKE4Q2Xc7KrTJW3yhs/  
2RsuZwwBjsQShPd0JxCyJVmA5v//getpgYextLSYKdHf  
+PbUO93No36Ht77eqHCvNnvASaBLp4rooClvpYGurpx92Sx6Kw3Uv873ZbOHvA7Qtd  
12m7ZZ6a00UN  
+225a7F5tt8XpA7Y61rJi1ViPeSgPZ7XZnuVh4zdsseisNhAEtzgOvaZt9yGsCdSwLeJKk  
16DNSm  
+lgSCghQAax9NGbfal1wQSD7YlgnE8H0a3iY4a8FYSyMKAxgJoMqBs1txbaSDs6AR5  
Zv6Asllzb6WBIKakiefzCQA5A8Jmjb2VBtoMKHXcUSM2e8x3ADoPAAh4Jr7v5G7IVKLjh  
r21DHS  
+DshP0xyAel3CZk29lQa6ebAUeLxe74KwWTNvpYGSFVDqOK4LPFF0QdisibfSQBsdj  
QFNo9Eol2zWxFtpINiJw+Fs5hcBTaeDwaibETar76000GZAnggleLqSyNxmd/  
luQOM4LngceDBP8ACQSnTY+L111geRtqxlMfLDAsiFkV8H1GWSyNBmW7xO  
+eiJyo7O87B4MCAa1lxazd9bUwglgFYd7a8Dwo4GlqYQgc02f291kEcd  
+YJHVmZusye8djKAlKxHHnjEiEUIVOTDFKKTpdxXdnRjghI7GjgUSqTNmvurTRRkMQ  
AtBp53lkjJuNRiY72cW8NiwdLHdyJgmcdkGKTGz2Gd  
+xPOhoGVAUjZgCo8zas33cW3lviFqWhyF2ND7YbaZM2qy5t9JEs8ITPexoJIFksUzTZo  
+5Tk3FiIWh7KAKpEzarLm30v/2ckczlYXdsGU6NvuY6wFdJjn+y4OXKR3EytP/  
WMNbNYD6Y9Ag8WCs/F4yo51t9oBrAoGJzJV/  
+WokJDow9NbaF7QJAKkNpMyb9KPfjL2VBrLtznLpQ0DIYNTKdrHZp1wfqAOiEjhMYsisl  
N9l9LSmtz7RBxLmGCSxW9k9pYye1LPZR9wACNQagIbutgaSfvTI3Fvp  
+xDwgDd6ZERAvm9mT7kB0HIZJIn4Ekp7W+KRRKfG3kqfYyAg4HHy6ZZwpB  
+daHgrVfG///  
3z5u1l38aODsbFrcH1opJ7VPgRabNnfPeal8+13REdPV4FBA4CRkQyZWekt2rU8M3IV  
REQns9AG4XmgxLRjc0+v0d4q1ZNVjznNw8WgsTiviaRXmh5K/19jzsaL  
+biwVwhsbeXdLUfffa+nrfSX9P4rYjnRcdFBWGSQ+KUf3/  
ybp7vuX6lFp7woaN9eDBxiUElwCDH/  
wfCWzUrXwTrgLCj14rGuhTSxx8Q3qpZLnSQ4MIXHS2He8u  
+xvqJ8FbdCmEFTXxcQdFAJkPP2qfVNvsLNy1vjg9WBKQsQMkpfiW8Vbt6SkdLFkb50  
VeEt2rX1MGA4MHUIhkRfvTFh4S3alcEHaSczmr20R+Et+rXCdu63DJsux/  
9+R7hrQZE0NFEHIVh/UV4q0F1B6zSEdlWF/lSsdn/  
AccLgKctjBheAAAAAEIFTkSuQmCC) 2x);  
}

@media (min-width: 240px) and (max-width: 420px) and  
(min-height: 401px),  
(min-width: 421px) and (min-height: 240px) and  
(max-height: 560px) {  
body.safe-browsing .nav-wrapper {  
background: var(--google-red-600);  
box-shadow: 0 -12px 24px var(--google-red-600);  
}  
}

.safe-browsing-billing .icon {  
background-image: -webkit-image-set(  
url(  
VEUAAADcRTfcRDfdRET/gIDcRjr/  
ZmbjVTncRDfcRTfcRDfdRDzgSTncRDjeSDvcRTjbRDfbRDjeRzvcRjfbRjjcRTjcRTjcRTf  
dRTfcRDjdRTjcRTjbRDjbRTjbRTjbRTfcRjjdRDrcRjfbRTjZQzfcRDjZRDfZRzbWQzXXRD  
XXQzbXQzbWQjXYSDvWQjbbRDfOQDPSQTTUQjXCPDDNPzPJPJLGPthVQjXMPzP  
RQTTWQjXLPzPDPDHYQzbAOzDTQTXHPTLIPjK8Oi++Oy/  
FPTHEPthPQDTQQDTUQTXBPDDKPjK/OzC9Oi/////

PQDPRQDS3OS66OS7TQTTEPDHXQjbMPjMBhLaWAAAAL3RSTIMA4tgPAhYFCcL9  
8B4x9ie1+s49WICbqXNKZY3puqcgVdLZnL2qKg9zmXpjfontV8LANsAAAJrSURBVHhe  
7ZTndulwFAY3ARlgBAg9vW1v173ROylby/u/ys02Fx3MNaxs9h/zAHM+Sfa8+M/s2LFjx  
+3tdjwH+/  
sHWxHVAerb8KSyANnUFkRXwLiK78llgJHJxRalwSMd11OGOeV4nsM9FO0dxhJdw4Lr  
OJ6jYy46PoohqgEHatE9JViiFNWTPiEITpIRRXcQ4C6aJ3EJAS4TkUQXsMJFFE+  
+CCsU8xFEBSAoiHsaQNIQ7yuQCFe3DiHUhtKIlzdKoRSFe0r8sXDAkSoumkligYalOkIf  
eWi56EESFm8r1w0fFll4epWgBA9qOMpmirCfeWijtoa9WSx6taAELFBRI/  
viIS3BJRlBk9/VFTsLrifUXRuNfXLU0y/7m6p0CKxqN+v6lJU/  
k3eJxu7Os5LWKDHi1tYstKG1zON1X3DGiRMR80Mx3fdCbc1+bQe3o2SJrYXcV0fFMx  
L9xXiz0987BBtux65qaCeF8IHCR3FabBTQ3xvk4M1yN5B/  
Mw2+urew8hTP1BM38Qnu5evK8gMw+7lcfH9E3ZIEBfMSO//Kf35+Cm6ua  
+rhbSYDeEa9CUyW3qK1Hlj5DBz8dWd0bWCd6Ult/uMPER+BmbV/JHrVG/  
a9MsEyV5fsK50R3frmBFXtCtVXmt73H4PhQ4t9k9rkJ55tYXwZrO4rCEUfPHfUEcuaZC  
/umw97TfaVpslu2tCb2lRWnBIKFtf+huwrjaa6Pxv7RfgW7nubJPtKI/X0puQO4k/Pfe/  
ovtLY7KbxVwve0/sE3VeaLoslBkEDvt8Hoq/hKGwQYvoq5OMnoq/hLABgc/  
FVn33PX7pAfE5QHR6fAAAAAEIFTkSuQmCC) 1x,  
url(  
BMVEUAAADcRDf/ZmbcRjrjVTn/  
gIDdRETdRDzZQzbXQzXWQzbXQjbWQzXZRDbbRDnWQjXWQzXYSDvbRTjcRTjbRT  
fcRjfcRTjcRTjdRjncRTfdRTndRTfdRDrbRTjcRDnbRDfbRDjbRjfcRjfbRTjcRTjdRTjbRjjcR  
TjcRDjcRjncRTncRTndRDnbRTjcRDfZQzbcRTfgSTncRDfcRjjZQzjcRTfVRDbcRDjcRDj  
WQzXeRzvbRDjXRDXXQzbXQzbbRDfeSDvWQjbVQjXIPjLOQDPXQjbCPDDNPzPUQ  
TXRQTS5OS7QQDTUQjW3OS7SQTTTPQDTFPDHPjK2OC26OS7HPjHOPzPLPJLMPj  
PRQDTGPDHTQTTEPTHLpZPGPTG7Oi/HPTLKPjLTQTXyQza9Oi/  
MPzPFPTHDPDHPDC/OzC+Oy+8Oi/AOzDWQjX////  
bRDd3undHAAAAQnRSTIMA2AUWCQIPHj39wvbO8DH64ifqqYFmtrVMc1IKS5x0nY6P  
WKqbjYDpZXWCZ1py8Jv9McJXV  
+KA9qioPc5l6Y36J7VmcHe8AAAFWUIEQVR4XuzWS4rCQBSG0euz56lSgiaEjHwgGh  
AhDnRF3/6HDY1la5WPjP4a3LOKY28555xzzjnnnHPOuSyzpPR7vb6lZAUrS8hgB7uBp  
aMEKC0Zhz3A/mCpaPjTWCK23GwtCcMjN8ehpWDN3doS8HPi7vRjejX/  
1CbX8qA1sdGZB  
+eRaW14sjGp8YQnk7EpVQQqE7peCFyupjMnYm4yGVGZ7q1EyTZbEEche2uUbLMIL  
5W6t4Zkm22lkm02561c89aQbLNTpPgq3hqSbbbmo1r41rhW8NaAaLMzvJlTvDUg2WzF



```

.safe-browsing-enhanced-protection-message .icon {
  background-image: url(
ZHRoPSlyNCIgaGVPZ2h0PSlyNCIgZmlsbD0iI0ZGRiI
+PHBhdGggZD0iTTkgMjFjMCAuNTUuNDUgMSAxIDFoNGMuNTUgMCAxLS40NSAxL
TF2LTFIOXYxem0zLTE5QzguMTQgMiA1IDUuMTQgNSA5YzAgMi4zOCxLjE5IDQuN
DcgMyA1Ljc0VjE3YzAgLjU1LjQ1IDEgMSAxLjU1IDAgMS0uNDUgMS0xdi0yLjI2Yz
EuODEtMS4yNyAzLTMuMzYgMy01Ljc0IDAtMy44Ni0zLjE0LTctNy03em0yLjg1IDExLjFs
LS44NS42VjE2aC00di0yLjNsLS44NS0uNkM3LjggMTluMTYgNyAxMC42MyA3IDljMC0

```

yLjc2IDluMjQtNSA1LTVzNSAyLjI0IDUgNWMwIDEuNjMtLjggMy4xNi0yLjE1IDQuMXoiLz48L3N2Zz4K);

}

</style>

<style>/\* Copyright 2017 The Chromium Authors. All rights reserved.

\* Use of this source code is governed by a BSD-style license that can be

\* found in the LICENSE file. \*/

.ssl .icon {

background-image: -webkit-image-set(

url(

ZmbjVTncRDfcRTfcRDfdRDzgSTncRDjeSDvcRTjbRDfbRDjeRzvcRjfbRjjcRTjcRTjcRTfdRTfcRDjdRTjcRTjbRDjbRTjbRTjbRTfcRjjdRDrcRjfbRTjZQzfcRDjZRDfZRzbWQzXXRDXXQzbXQzbWQjXYSDvWQjbbRDfOQDPSQTTUQjXCPDDNPzPJPjLGPTHVQjXMPzP RQTTWQjXLPzPDPDHYQzbAOzDTQTXHPTLIPjK8Oi++Oy/

FPTHEPTHPQDTQQDTUQTXBPDDKPjK/OzC9Oi/////

PQDPRQDS3OS66OS7TQTTEPDHXQjbMPjMBhLaWAAAAL3RSTIMA4tgPAhYFCcL98B4x9ie1+s49WICbqXNKZY3puqcqVdLZnL2qKg9zmXpjfontV8LANsAAAJrSURBVHhe7ZTndulwFAY3ARlgBAg9vW1v173ROylby/u/ys02Fx3MNaxs9h/zAHM+Sfa8+M/s2LFjx+3tdjwH+/

sHWxHVAerb8KSyANnUFkRXwLiK78llgJHJxRalwSMd11OGOeV4nsM9FO0dxhJdw4LrOJ6jYy46PoohqgEHatE9JViiFNWTPiEITpIRRxcQ4C6aJ3EJAS4TkUQXsMJFFE++CCsU8xFEBSAoiHsaQNIQ7yuQCFe3DiHUhtKllzdKoRSFe0r8sXDAkSoumkligYalOkIf eWi56EESFm8r1w0fFI4epWgBA9qOMpmirCfeWijtoa9WSx6taAELFBRI/

vilS3BJRlBk9/VFTsLrifUXRuNfXLU0y/7m6p0CKxqN+v6lJU/

k3eJxu7Os5LWKDHi1tYstKG1zON1X3DGiRMR80Mx3fdCbc1+bQe3o2SJrYXcV0fFMxL9xXiz0987BBtux65qaCeF8IHCR3FabBTQ3xvk4M1yN5B/

Mw2+urew8hTP1BM38Qnu5evK8gMw+7lcfH9E3ZIEBfMSO//Kf35+Cm6ua

+rhbSYDeEa9CUyW3qK1Hlj5DBz8dWd0bWCd6Ult/uMPEr+BmbV/JHrVG/

a9MsEyV5fsK50R3frmBFXtCtVXmt73H4PhQ4t9k9rkJ55tYXwZrO4rCEUfPHfUEcuaZC/umw97TfaVpslu2tCb2lRWnBIKFtf+huwrjaa6Pxx7RfgW7nubJPtKI/X0puQO4k/Pfe/

ovtLY7KbxVwve0/sE3VeaLoslBKEDvt8Hoq/hKGwQYvoq5OMnoq/hLAbgc/

FVn33PX7pAfE5QHR6fAAAAAEIFTkSuQmCC) 1x,

url(

gIDdRETdRDzZQzbXQzXWQzbXQjbWQzXZRDbbRDnWQjXWQzXYSDvbRTjcRTjbRTfcRjfcRTjcRTjdRjncRTfdRTndRTfdRDrbRTjcRDnbRDfbRDjbRjfcRjfbRTjcRTjdRTjbRjjcRTjcRDjcRjncRTncRTndRDnbRTjcRDfZQzbcRTfgSTncRDfcRjjZQzjcRTfVRDbcRDjcRDjWQzXeRzvbRDjXRDXXQzbXQzbbRDfeSDvWQjbVQjXIPjLOQDPXQjbCPDDNPzPUQTXRQTS5OS7QQDTUQjW3OS7SQTTTPQDTFPDHJPjK2OC26OS7HPjHOPzPLPjLMPjPRQDTGPDHTQTTEPTHLpZPGPTG7Oi/HPTLKPjLTQTXyQza9Oi/

MPzPFPTHDPDHBPDC/OzC+Oy+8Oi/AOzDWQjX////

bRDd3undHAAAAQnRSTIMA2AUWCQIPHj39wvbo8DH64ifqqYFmtrVMc1IKS5x0nY6P

WKqbjYDpZXWCZ1py8Jv9McJXV  
+KA9qioPc5l6Y36J7VmcHe8AAAFWUIEQVR4XuzWS4rCQBSG0euz56lSgiaEjHwgGh  
AhDnRF3/6HDY1la5WPjP4a3LOKY28555xzzjnnnHPOuSyzpPR7vb6lZAUrS8hgB7uBp  
aMEKC0Zhz3A/mCpaPjTWCK23GwtCcMjN8ehpWDN3doS8HPi7vRjejX/  
1CbX8qA1sdGZB  
+eRaW14sjGp8YQnk7EpVQQqE7peCFyupjMnYm4yGVGZ7q1EyTZbEEche2uUbLMIL  
5W6t4Zkm22lkm02561c89aQbLNTPPgg3hqSbbbmo1r41rhW8NaAaLMzvjITvDUg2WzF  
lyrBWwOCzc6Jkm12QQcL3Vtlmy3opFC9VbbZJR0tNW  
+Vbbahs0b41rhc8FbVZqdEyTb724t5/bYNA3G4e  
+80NYl0gGFkvaR779KKZUWuFKe7nllsT5X//2M5VMZiZB9DQj74xW8ffrwjP90Mb/  
07Vf5CbXYJg0BtO4toKS9vhYHGy1vDZg28FQY6tBZls8tYBehwNLtyt1nhrTDQaDQcW  
Aux2SJWAXpOBpWMWSvm4q0w0Gg4nFQqFTd/m72HIYBYQJV  
+w83bZu9jRaDJYEB4osjJ02aFt8JASUBRq  
+PlarMrWBGi8lQajVanXA5kopUcvBUEGrCAWhSoXs3PZtewKhA/  
MMbTbcpEa7l4KwwURZSHANnVnGz2CVYGmg6oZ1u1XGy2hNWBCA8BogE1m7ZI  
+ShNVMrdW2Wg/v  
+Amr2eRYCcGLBZU2+FgcSBESDfdZxdwGbNvBUGihKgnk1OjPAEwS5gsybeCgNNdT  
QLyAtqtRCwWQNvhYH4ndjtNnlAnlet1uIQsFI9b4WBpgNyaUCEJ45DwGa1vRUGanU6n  
McmB+ZSnlosES3nvm/tUpGm1tFPd5DDAyKFBjGpzRaxSjW5J0o8/MAQ4ZEyKua/b  
+0Np175bIMERDuaECFBZGqzBaxY9iAjIMbDK01U0OVZxcplE6BljLzFRixgQDwflCJaX  
cC+1ToKyOYHFvCOLjPiNmvrTBRI  
+oQoGTk2Z1YQyleiWhIEftWnx8Yf8RcyiMCEkyhic2u4xOWSw9MBBQENTQFI83a  
+iL2rdgpJ1rms45mByYzhbDNwt6qTtTIQC7r6FT/CLRQ02ZLWKc8OmK  
+LzooCykhKpl4q7p+7B/d0SjNggRbqGOzm1gPqL3PX3niZakOQsenf1PDWzWAxr  
+JBtEDQxnnJTISNmvrufBK75t45bORBNGSobcqb9DqBCjdQOI5E370xthbYaDRiljRDx  
KQwJk9a+o2u431gYZERBo/kcBlfvJ/TrSt6K1b+kDUHMkra2V3j5zRlprNbmADILbQ65S/  
z2ggyY82zL0VXsdQnnLdhSOKQzWbLWADIMpDgOrd3q958QiigrG3wusYzmNbXmY4s  
h+tangrVJ2Dgy97X9v0CmlLzliHcj3ZPTL  
+h6DN7mhYR5nxHI4mtKNbLCaMaX9QDDKFO6C36hDttcdJQFGLeTWRIlupocGOj62c  
Bb9WqesLTFwfm000MQgqz9IDLW  
+Hve35HM9Fnqw9HetBkNsF6+Yaet8Jf0+xbka0XbYspSMlg  
+5D8/8psnqdYv3qso1vsS9Hy6SaGQ6AYHP9ngLdqllVpilB8RygRQjGEdOsc4K26RGzk  
6YTjxhbDDdzXcfwC8Fbd8glPnR4Y62gBAM/  
a1WybfYVNyyUBiZFPXYCAH70GvFW7nFRHH7Egyl8uAd6qXZ7NAqoilG6ZKuBH184D  
3qpdAQIIWp0p9dE7wFv1q8Y6+njLoPl  
+9P4C4K0GRKSjgTyywvoAeKtBxVWU6YhorovcvA14q0HtouwU0Fw/+jzN8w/cQ/  
zg6ug2/QAAAABJRu5ErkJggg==) 2x);  
}

```
.ssl .legacy-tls.icon {  
  background-image: -webkit-image-set(  
    url(  
SR0IArs4c6QAAB21JREFUeAHtXF1IHFcU9ie2bovECqWxeWyLjRH60BYpKZHYpoFC  
U60/  
xKCt5ME3QaSpT6WUPEICEXyTUplojfgTUwshNpBgqZVQ86hGktdgSsFGQqr1t9+nd2
```

WZPefO7LjrjYzcJmZc8495zvf3LI3Zu  
+dzcoKt5CBklGQgZCBklFMZSB7r4G3tLS8sLCw8D7ivo1Ssrn5WYL9AZSC7OzsAuyzl  
HuCHcsjyOawZ7lbVFT0W09Pzz843rNtTwhqaGh4ZXV1tQFZfYZSDgKe85MhyFpBvTso  
V/Py8q5g  
+9OPn0TqpJSgurq6CpBxFuUEQO1LBJgH2zUQdgPlwuDg4LgHe18mKSGovr7+2Pr6+j  
kgOuLLVeKVJnJzc78eGBi4nXhVe42kEtTY2Fi8vLz8HVRMKXvY1GjRmvrz8/Pb+/  
r65pMVIWkEodV8vLGx8SPI2Z8scH78gKTFnJyc02hN1/3Ud9ZJCkG1tbVfwnEnyMlxBp  
DOkcQybG9ifwv6OezvRyKRv5eWljhyZeG4AMcvweYNnHKkq4TNcezzqXfbYLSBm46h  
oaELbrZu+l0R1Nra+vz8/HwPgH/uFgj6xwA  
+inlNt8Evvb29Tz3U2TFpamp6EbfvR4hVhXISisldpXKAWJeLi4tburu7/1VMXMW  
+Ccll9TKA/oTyni0KQC5B34V9J0abRZutVx1i70fcDti3YR  
+x1UPcSZRPEfsvm52m80WQaTm3beQA1Dr0F9EffANwDzUAu5GDqIPo975FrGbEytV  
8QT  
+JlnTMT0vyRRD6nEsAZLutOlPUDw8P86Eu5VtNTU05goygFGvBQNJI9ElfaHpNrrKuV  
WCHDHLOanoAmUKr  
+QBgZjWbZMtnZ2cflpWV9cPvUZRXFf9vHT58+OnMzMzvil4UJ0QQh3KQ8wM8iS0P5P  
SjVOGWWhCjpVCIXJ+AgD6EeA2lTAoFbB  
+CyKnp6en7kl6SiYIKhuYhcBYEic85JAethu9bad/Qyq8Ap/  
iwCpyLGEUPeX2Y9PTcwozNE7JGzhQCn0k7MwYAsaBMSXh4gZmLpJNknIqQebe6JT  
mAbB59zru7GanQyW5KvtHJe8ln1TUj3B/  
QiR033t0qvby7eWpB5sUzDgeu0jqE1bshJ85pkgQGU7XBG0dVy8lp6EoQrkQFKolv5Wi  
uF/dqKHcC93JObMSO2B4xuSnqbbErQQggDum4Mkt8CLR6D4CSGIIVgqLIFmtrJYi/  
BMlJf+yStq4g3lpOoAZjl1POc+bGHCvdVGyLaGVI5TQMpV8C+eLZGXUS9L3B+lJAuc/  
8FCyotkVS8jvGcFwNlnfOoweQj  
+LKJOXfKz53M1pFMdn2xlpno1Hklr0e8XdysYXRp9qCOPsAPd9x4jYQdC1OGHCBBX  
O5yVXMQCWIUzNgPG72AYGW  
+XuO6C3AQmImdidE5mimoZyqrXOVIGg5bxW3weHNRH/  
sinOSBgExE7sSWsyVtjaCSiRnuAraE7VkhiiZBbuYK8GrBIFtsRKC3AtU1gmA0bBrudK1  
bRQ7oMR  
+oMh9i1PxLqaA0bBrueotCAG25smdgTj74JRlyrkFu5gr81JvMTRHsVJ0aiZTSInFqWHX  
crUSFOv4WT5WWxA6rq1JPCc5nNRzyjLIXMOu5cq8VIKGEwnijGemEOLEacEu5sr6Nol  
eOQPwHGxzOjgJNwt2MVcmqRKEjmtOYUF8PIJsgyYWsVty1QICZiJBuAqVQcvaKx4Ldj  
FX+IVbEHR3pcBg  
+zgXEki6IMulmdgVjGKutFUJ4oJJOFxxOsRVyOcqC6c86OdmZUjc8hnmyFw1/  
CpBZjWpOLcOkqo0h0GVWzDfsa2cVQkyiV6VEkaw5gRECCrJft0y4iVmBUcYo5RWyt  
BXGoLw7Woccy  
+EAE7Ys4DfWiwFgog10yOgmpbZCWI65Bxj44ptdtwZQ4qusCIDcY2CRByu  
+G21tpKEJ3CyXnJOa5KhluxJF2QZMRlrBlm5Oa6htGVIMwljMP5hBKg2SxektRplxEbS  
GhWgEyY3BT1ttiVIJpxkbbkBVeG64tGgnirGUwjBmMcfC0np6Hn1RMua264/  
OUorog4xesMmupzkBMBMb  
+ivCPFAIbPa5k8tSAGwbRJOxyLk4UEgSKVZ4HYiMVCDhdQtXsF6rkF0aFZTf8zgoveE8s  
qgnEIXSzlth+SckggAtg0sZvgkkVX4Ca1R5Nq+0tJSfq  
+lvWpwbeAJrBW8zjWDEshUydjngJgxFA0bR+SvcPEuJYIhoRYUdYz  
+6JIZBizeKIEitD2X9+NqTGp6yluhn8Aw+70ZTSym/lX0zRiMxZiaJ2lIZk1vk/  
tqQQXlC0GnCDZmqQs/ZnFjyOjRJ/n+HArNn1PZDzipF5234uyD  
+YH9dXS6b6Jk5udQsfz9Xz

+o89VJxxITPeazBR7ADqFF8JuJtGyMTQyJPOe4AfXdSdscm4Xn52AjLh  
+21fWpy4yPep3JYaSrQP+Rys/  
Cx9BqzuPhb9wZO1nnKWIBTnDhHws4GbGcZ9pfU1hSCVUhAyEDIQMhAyEDAWfgP5  
qNU5RLQmxEAAAAAEIFTkSuQmCC) 1x,  
url(  
SR0IArs4c6QAAEp1JREFUeAHtnVuMFkUWx2dgRIBhvUxQSZTsw25wAUPiNQTRgFkv  
8YIbZhBcB8hK2NVkXnxRY0xMDFffJkHsyxskBFRGIJ4iWjioLJqdL3EENFZ35AELxnR  
HZFFBtjff+gePsbv0qe6+vv6+6Y66XR39aIT5/  
zPv6urq6q7m5rCEhAlCAQEAgIBgYBAQCAgEBAICAQEAgIBgYBAQCAgEBAICAQEAg  
IBgYBAQCAgEBBoTASaG9Ot8l6tWLFi4sGDB3+P1HStx44d0/  
a85ubmyWwnHz9+fHgbHTdxPEj6IMfD2+j423HjxvWTPryeeeaZX65fv/5/HI  
+pZUwQ6l477vjD0NDQAgiwgOBfynYa23E+l43OY+jcy/Zjtn0tLS19zz///Oc  
+y8ijroYkUEdHxxSCuBDAF7DOZ/  
+CWoAPmb6m3J2sfexv37Jly3e1sCPLMhuGQF1dXRP2799/  
G2TpBLCbWfuyBM5B9xB5XoVIPVOnTn2xu7v7sIOO3GWpewJR21xJG  
+ZukF3MenbuEC5u0A8kb6YNtY5a6YPilvWRWrcEWrx48Xyl8xA1znX1AXVxK6mR3oBl  
qzdv3qxbXd0tdUcgapybIY2IM6fu0C5jMER6j3U1NdlrZcRyd6puCARx5kCabtbLcoeiR4  
Mg0UesXRDpPY9qM1OVewltW7asjT6bJ0DgLG6y5t9dTpl6j55/0Ld2/  
YcOGAU86M1GT24BQ0zS3t7evxOvHWNsy8T7/  
SkWeB3t7e9dSK4IUuVtySSBuV9NoID8LWnNzh1htDHqHhvd3Nb21qb40qV67Y0tXUz  
yMzxd3Urt8wk5AnlOwjZXmAibk0n52MtNDbRq1arWgYGBx4HlvmpAwy3hJ8rpJzD98Zg  
W+1+RPjh+/PjB0047bfDQoUMa+2o6/fTTJ//yyy+Tjx49OjxOhsxFJA+PobE/  
PJ5G3kmSrcLyZFtb2wNr1qw5UoWyKhaRCwltWbLklsaqthCEqypa7CggwqD/  
bbZ9bPsuueSSTx955JFjjupOyYaecbt3756Nbo21acztGraZEQR97zPW1vHcc899dYohN  
TioOYFo78ygvfMavl+Ygf8aQe  
+lhumZMWPGlgKt4YTMF8pp2bNnzzz86oRI7RSo0X3fyz78uoF20R7fii36akqggG/  
nZUA+12J0JVII8zrr08htA+BDleSzPM+t+YwDBw7cjo/LWa/3WRY  
+fs96Sy37jGpGIMhzM1foZgA9wweoAKnb0VbaL6uZRvGpD52+dTctZDbtqlfQuwgy  
+XqA+ZmaaDEkqkkPdk0IRP/  
OnwFwPUCmHjGPiPNMa2vrY5s2bfrCd9Cz0Ld06dKLjxw58iC67/  
JEpCFItBwSqeujqkvVCRTVPC/gpQ/  
yfEgA7tm6deuHVUXNU2GLFi26nAvgKXy43INKkej2atdEvqrRRP6rzRPdtIKRB9APANa  
9s2bNuqpeySPAZLt8kC/yKRGlpYVahK0wLi3i/  
0zVaiAcm8GVtos1VYMZoHfQL7O8p6fnW/  
9w1E5jZ2fnefQ7PQ0+N6axAnzUsJ5HTVSVp7OqEEj9PNzz3wWYNI/  
qqqlfZt7MEwCUy3GhNIFXXsjTTG/z/  
dQkj3KYppbeN3HixDkbN27cl9amSvkzv4Wph1mdhBiShjzq85jPVfV4o5JHgZJv8IG  
+cpgm+BcePny4V9hLb5ZL5gTS8ARXVpoe5k8B9AqA/  
VeWQORJt3yVz9jk3B0hzKOhoUxdy/QWpsE/+j1edPWAK/lt1oUA  
+qOrjnrOR7vxLliwnfVaVz/of7uN2/5Lrvkr5cusBsL5adzL11cyoNR5iLnt0qRJN45V8ggX  
+S4MhEUpnCqIKwaKRSU51/  
OZEIgrphnDn2Xr9MQlwFg7xuKbnqMDKQyEhSuJFIMoFpnbcTihUDST0Gk  
+D0C9xVWnyVNHR4M5Vo  
+FhTARNo4Yzl1i4pi9dDbvrlzmMPdTpMs0VDWYrx3Lt63SoWpqUpul2kQkml1OrsS5Ae  
ZYT/c9x9p7DRRNghchjx7Vx3Sbp0TgR5J1YQkjElwe8eOXE0b0+djxWgNxxhWio4h0Ms

+pVJ6H6eWr2qM64IKlzkmeIq48+4jWsA5yvBuedHLQYIR4H57ng7O2Vla81EA22bhwy  
A4tTD9eSPMYg1FxcWAKzB0Oaoxg5ZC2exRuBuCr0xuhlxYspnUrDcleGJ0pLhDPFEli  
GdHYUO1cuTTFSrMrJWM55lxCGaaKUaYE8BzQwytZ0+zAV0qDCwizCzjyK7xKrUjB6l  
RA9zvoGj3kaASA81Gij6qWAZiJd2AIDq27FSjGz5ism74VANOjMTuD4hzNnzvx7MaNC  
WnlEhKGwTJ7jhKRLzIqVkZpA3E  
+vhNGmT6zgsD4Hd4+v12qKOTZW0oShsBSmFp8VM8XOkqeYbGoCYcjKYoorpD1Tzz  
MJK/hW9dMRIs9YC3aM3SnFpCKQPiuHER2naKxwoCtFE  
+AriXTRgSEqUMt1KEYGos6RTwVgfRNQRZPYu3tV7enjgFqZwfRjhuNZp5dhRDY7aT  
4qklhJplJ1UI29N7W8kkg5QVARdsuYPoo6TOizOBalDpU7qmCeBUsa/n9aU/  
ZwRzIFHYCmOjSTcplsY8l+LWsZSRjJBnlQem/  
Dj39liCnO3UcmzLJxTCmNhYXqFuiWK51sUO5xqlwhYYCxxE3nlmnbgssSwujlW1ZbH  
GckR3GgKZejK5MnoZBKzphw5GvG7gHWEsrl0ummJZqNuJQNwz9ZKg6fcBjB73FBYc  
9rNDwlq1Yqn/ibhY5EQgusFNjOWK  
+Enf53ExMOSxlyCshbkIp35GY5GPZZ0lhHGmwmD429X6uFPs2FjeCmthbsHAGtNYtx  
OBMO7SWEGSLcb1JZELMv4QsGJuJWlsqZIA  
+lkbpneM8K4QKAY8SpTrZgrpoqt1TwzgfSnP4xLnA/DftIHLA2GBfl0CAhzYZ9Ui2la/  
cUxaZZhucREKNCqz9palv4wbcMClx/  
ZCHO9XmVZrLftypxAMNvqhMXhIFsGAQfssycQj/CmQuiTCAQqE  
+QsT1mxt8ZWtpvGspSB+  
+r5MFu7SZe6IFA9vReWFHjkTNgrtgbdw6lutzDTR7Mh21dWo4K8HwQcsDfFVla6EMj0  
CX9YbR3Y84Ne0KK7hRV7U2ydCASrTSxlkpPViRB6TwhYsbfG1olAZDIRSH  
+98YRHUGNEwAF7U2xljvkWRrVoKiT  
+ZZLR9yDuAQEr9tbYykQzgTz4FVQ0EAJmAnGfNN2S9LO2BsKrrlyxYm  
+NrcAwE4g8JgLPt391hXoDGeuAvSm2gspMIOUjoX4T2UAXqStXrNhBY  
+tEIDKZWOrYAfhXUcqsQ7Ym2LrSqDEUwRUAKzWD2rDUgMErNhXpQ1Eld8YsTAN  
vhp1B/HyCFixN/  
8BydwGqsYlb3IMwtmkCFhH162xIR1mApHHOsJrvQqS4hPkKiDALcyKvSm2Kj5zAlHGd  
GbHuZRTAZ5wuhwCEeb5lxBfO/  
8SZh8rZ3zhOdpMk3bv3j27MC3sZ4+AMBf2SutSTBXbpPKxnLIm0M8/  
MGxvrCDJFuMWJJELMv4QsGKumLr83MZMILmlcR9bXMW4QCALYB5krZhbYxqb6EQ  
gjDO954Vx13BPNk  
+fjY0MWxsCwlqYW3JZYxrrdiJQS0uLiUAYN2nPnj3z4kLDNlsEhLUwt5RijWms24IAfAnrc  
xj+dawkyZY+iVSfUktSRpA5gYAVa8VSMXXBz4IAUUh6W0zihSuinc/  
CnJ44QxB0QkAYC2tjZIMsC3WnlZDpNkahGpX/  
U2HhYT8TBISxdQaENZYjhjsTiGpvO1qGRjQl2OHKWJ5ALlikQACMVxizD0WxNGY7le  
5MID6l9h0qXrWUinPX8yWs0KloAc0gK2zB+l+GLBJ9NYqlMdsJcWcCKTvMNX+2jklO5h  
+zOHk2BjO5YOSw0JoUxFo6tSpL6Lsh0KFCfYXLV269OIEckHEgECE6SJDFon  
+EMXQmO2keCoCdXd3H0bV5pPqKu9RxY47cuTlg5Ulg4QFAWEqbC15kN0cxdCY7aS  
4tcCTOaM95pCs+1ViYS7+Jb5ZXFGkQSBCls70oiWyjjGLtCFU7TOU5RQAPsA  
+6jb5ySWOFAVwp5ngrTPCoAleC0MBSW1tpHMVPsEhRRViR1DSTtMnN8AxUcvvyzzz  
77a1nrwsmKCAhDYVIRcJSAS8xGqRg+9Elg/iC8E0a/V6yAcmk4vrqzs/  
O8cjLhXGkEhJ0wLC1R/lxipZgVP2tL9UlgFYIRZkdw/  
hze39bPQZptZgdpYRZhd44VDZdYISrDG4G4n76CYR+VKqhUOkDcyB  
+E7y91PqQXR0CYCbviZ0unKkaKVWkJ2xlvBFKxGNfF5rjNhKYmRo8fZRDwamu  
+sSovrlSZg//Hoxg5ZC2exfutg0fKtRR1d/Hiyqbuo2F3BVeHaZpIWY0NeBLyXAB5/  
o1rFzq4t47/oq10yFcyi9caSKUwMVu3o4GSJZY

+cSHA7ACgs0qLjO0zwwYYgYILeQai2HgF0TuBNmzYIPK49jRrMHC7yyf3vaKSQ2XCR  
NhgmUt9INRbLx65/0WJutwtLm9vX0Xu3NdrOU+vY21g9vZUZf8jZaHmMc8mG5h1Vwfl  
+Wd3t7eeWBqbp9WKSx7DaQCZSjtmTvZfl/  
JgGLnBZQACzVRU1NU8ziTRzGIYuGdPMOxLhZAX2k8at7KFAON2DstOP8W60Jqoh  
+dFNR5JrV5uJC2s17r6gpfar2NTsOXXPNXyje+kkCa83Sz/4e/  
5/0GHXMc9fwW8G6aNWvWC7xpYPqsjGN5uckGefS0pTHGq1IY9SS3ru4U  
+StmzeQWVlhqW1vbA9Qi7xemGfdn67EVQMdMP5F8lc/  
g5NpgVjPifWFvxNosnkkjerQVS5YsuYj5Ku  
+S7vL4Gasb4l7+MNXXE4CTyf08LqhWW2rbZvUwQx51EqZ5EXPfxIkT52zcuHFf1r5Uh  
UBygqtKf3rexXpuGqcgzw6+Prq8p6fH/  
DGkNOVmnVcDo9HYInl4otA28PmedR7txj2F6VntZ9oGKjSaNsx3M2fOFIGWkt5aeM64/  
zv+MLwSXf/  
lav34zTffrOvaSPN5pkyZ8jdq6G1gc4kRi9HiP1NL3wh5Phl9lqvjqTVAsQPURDdTRb/  
AcZoqOlandsK9dM9/GCfU01YzCaktNBnMPJ+niJ  
+6xd8OebwNIBYp41dJVSeQLIBEd0Kip9INTSICcAw9z7S2tj62adOmL6Q/74smwEfzWu  
+CPD4eZESe5ZDn2Wr7XhMCycmoJtKE/DN8OB0RaSv9Hqt5z/  
tTHzp969B7W9GrN4s8EUcm6ra1uNo1T4xNzQgkAyDRHIB8mTVVwzp2Jt5CptdZVcNt  
A9hDcXottvio7wGoZ3056/U+bcBHNZhwUfzbFBfdtSUQHICgGdwO3uN3TSP  
+KXwGATgXq7QHjo0d9FgHSol6DOdcl0iRX86oQ07eie7FN/  
pEvTX26APFV52iplf80JJMPUT8STlcZ70vS6lvJxOB0i/YT  
+t9n2se3Tf9UJtNpPqRc9SembhOhegO4FbK9ha/o+j8UI9L8/  
YcKE9mr081SyKxcEkpGrVq1qHRgYeJzd+yoZ7eM8QdDQSD+B7udK7o/2vyJ9UH/608/  
a4v9t6a83+nEJ7ZfJyE9G5iLkp1PDTGdfX0KdniVh0F  
+4PKke5jVr1hwpTKzVfm4IFAogAVgCs56AeG0XxfrdQtRNaq  
+IsuBURdsckcgOUG7aBok0iOp03wiFyBynucdyHMn7Z29ebMzlwQSSNRampS2kt3HW  
NuUNgaX4dmdjKivpQbKZY+7j06sTOlqwOhh/gfzeNXGWMeaSwAzcf6Er  
+vkuzDIK3nke25roNGBifqMuqmZLht9rpGOlctHrF217Nux4Fk3Blqdgkg3Q6KHWF0nqc  
WqcrWFNO+xroY4VR3LSgtC3REodpintfk0tEWk6+K0etxcmjdoIK/  
29a56tTGoWwLFQFEjXQmJVrJ2kHZ2nJ7z7Q8QZwvrWmqc1J9YqaWvdU+gGLyurq4J  
+/fvv43jZZBJk7JSj/THuj1t9TVUvRS4QZ+VS/  
tlME82pVbTMAQqRIJaaQokWkjaAtb57F9QeL5a  
+xBGr2nvZO1jfzu1jb5s21BLQxJodlQglAZs5xNEjVVdynYaW69dGOg8hs69bD9m20e7Z  
ieEqelA52gcsjgeEwQaDZxe1jt48ODvSR8ex4JcGtM6n2ONmk  
+CANpqzGt4FJ3jQY41sq+txtAGSfsGkgyPoXHcT5/  
Nly7/2yJvWAICAYGAQEAgIBAQCAGBAICAYGAQEAgIBAQCAGBAICAYGAQEAgIB  
AQCAgEBAICAYEcLvB/Q079+h6myXwAAAAASUVORK5CYII=) 2x);  
}

```
@media (prefers-color-scheme: dark) {  
  .ssl .legacy-tls.icon {  
    filter: invert(1);  
  }  
}
```

```
.ssl-opt-in .checkbox::after {  
  border-color: #696969;
```

```

}

.ssl-opt-in .checkbox::before {
  border-color: #696969;
}

.ssl-enhanced-protection-message {
  background-color: rgb(248, 249, 250); /* google-gray-50 */
  border: 1px solid rgb(218, 220, 224); /* google-gray-300 */
}

@media (prefers-color-scheme: dark) {
  .ssl-enhanced-protection-message {
    background-color: rgb(32, 33, 36); /* google-gray-900 */
    border: 1px solid rgb(95, 99, 104); /* google-gray-700 */
  }
}

.ssl-enhanced-protection-message .icon {
  background-image: url(
ZHRoPSlyNCIgaGVpZ2h0PSlyNCIgZmlsbD0ilzY5Njk2OSI
+PHBhdGggZD0iTTkgMjFjMCAuNTUuNDUgMSAxIDFoNGMuNTUgMCAxLS40NSAxL
TF2LTFIbWVjZ3YzAgLjU1LjQ1IDEgMSAxIDFoNGMuNTUgMCAxLS40NSAxL
DcgMyA1LjU1LjQ1IDEgMSAxIDFoNGMuNTUgMCAxLS40NSAxLjU1LjQ1IDEgMSAx
IDFoNGMuNTUgMCAxLS40NSAxLjU1LjQ1IDEgMSAxIDFoNGMuNTUgMCAxLS40NSAxL
EuODEtMS4yNyAzLTMuMzYgMy01LjU1LjQ1IDEgMSAxIDFoNGMuNTUgMCAxLS40NSAxL
LS44NS42VjE2aC00di0yLjNsLS44NS0uNkM3LjggMTIuMTYgNyAxMC42MyA3IDljMC0
yLjc2IDluMjQ1NSAxLTUuNSAyLjU1LjQ1IDEgMSAxIDFoNGMuNTUgMCAxLS40NSAxL
z48L3N2Zz4K);
}

.ssl-enhanced-protection-message a:link {
  color: rgb(26, 115, 232); /* google-blue-600 */
}

@media (prefers-color-scheme: dark) {
  .ssl-enhanced-protection-message a:link {
    color: rgb(138, 180, 248); /* google-blue-300 */
  }
}
</style>
<script>// Copyright (c) 2013 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.

/**

```



\* @fileoverview Assertion support.

\*/

/\*\*

\* Verify `!condition` is truthy and return `!condition` if so.

\* @template T

\* @param {T} condition A condition to check for truthiness. Note that this

\* may be used to test whether a value is defined or not, and we don't want

\* to force a cast to Boolean.

\* @param {string=} opt\_message A message to show on failure.

\* @return {T} A non-null `!condition`.

\* @closurePrimitive {asserts.truthy}

\* @suppress {reportUnknownTypes} because T is not sufficiently constrained.

\*/

/\* #export \*/ function assert(condition, opt\_message) {

if (!condition) {

let message = 'Assertion failed';

if (opt\_message) {

message = message + ': ' + opt\_message;

}

const error = new Error(message);

const global = function() {

const thisOrSelf = this || self;

/\*\* @type {boolean} \*/

thisOrSelf.traceAssertionsForTesting;

return thisOrSelf;

}());

if (global.traceAssertionsForTesting) {

console.warn(error.stack);

}

throw error;

}

return condition;

}

/\*\*

\* Call this from places in the code that should never be reached.

\*

\* For example, handling all the values of enum with a `switch()` like this:

\*

\* function getValueFromEnum(enum) {

\* switch (enum) {

\* case ENUM\_FIRST\_OF\_TWO:

\* return first

\* case ENUM\_LAST\_OF\_TWO:

\* return last;

```

*   }
*   assertNotReached();
*   return document;
* }
*
* This code should only be hit in the case of serious programmer error or
* unexpected input.
*
* @param {string=} opt_message A message to show when this is hit.
* @closurePrimitive {asserts.fail}
*/
/* #export */ function assertNotReached(opt_message) {
  assert(false, opt_message || 'Unreachable code hit');
}

/**
 * @param {*} value The value to check.
 * @param {function(new: T, ...)} type A user-defined constructor.
 * @param {string=} opt_message A message to show when this is hit.
 * @return {T}
 * @template T
 */
/* #export */ function assertInstanceOf(value, type, opt_message) {
  // We don't use assert immediately here so that we avoid constructing an error
  // message if we don't have to.
  if (!(value instanceof type)) {
    assertNotReached(
      opt_message ||
      'Value ' + value + ' is not a[n] ' + (type.name || typeof type));
  }
  return value;
}

/* #ignore */ console.warn('crbug/1173575, non-JS module files deprecated.');
```

</script>

<script>// Copyright (c) 2012 The Chromium Authors. All rights reserved.  
// Use of this source code is governed by a BSD-style license that can be  
// found in the LICENSE file.

```

// #import {assertInstanceOf} from './assert.m.js';
// #import {dispatchSimpleEvent} from './cr.m.js';

/**
 * Alias for document.getElementById. Found elements must be HTMLElements.
 * @param {string} id The ID of the element to find.
 * @return {HTMLElement} The found element or null if not found.
```

```

*/
/* #export */ function $(id) {
  // Disable getElementById restriction here, since we are instructing other
  // places to re-use the $() that is defined here.
  // eslint-disable-next-line no-restricted-properties
  const el = document.getElementById(id);
  return el ? assertInstanceOf(el, HTMLElement) : null;
}

// TODO(devlin): This should return SVGElement, but closure compiler is missing
// those externs.
/**
 * Alias for document.getElementById. Found elements must be SVGElements.
 * @param {string} id The ID of the element to find.
 * @return {Element} The found element or null if not found.
 */
/* #export */ function getSVGElement(id) {
  // Disable getElementById restriction here, since it is not suitable for SVG
  // elements.
  // eslint-disable-next-line no-restricted-properties
  const el = document.getElementById(id);
  return el ? assertInstanceOf(el, Element) : null;
}

/**
 * @return {?Element} The currently focused element (including elements that are
 *   behind a shadow root), or null if nothing is focused.
 */
/* #export */ function getDeepActiveElement() {
  let a = document.activeElement;
  while (a && a.shadowRoot && a.shadowRoot.activeElement) {
    a = a.shadowRoot.activeElement;
  }
  return a;
}

//

/**
 * @param {Node} el A node to search for ancestors with lclassName.
 * @param {string} className A class to search for.
 * @return {Element} A node with class of lclassName or null if none is found.
 */
/* #export */ function findAncestorByClass(el, className) {
  return /** @type {Element} */ (findAncestor(el, function(el) {
    return el.classList && el.classList.contains(className);
  }));
}

```

```

    }));
}

/**
 * Return the first ancestor for which the {@code predicate} returns true.
 * @param {Node} node The node to check.
 * @param {function(Node):boolean} predicate The function that tests the
 *     nodes.
 * @param {boolean=} includeShadowHosts
 * @return {Node} The found ancestor or null if not found.
 */
/* #export */ function findAncestor(node, predicate, includeShadowHosts) {
    while (node !== null) {
        if (predicate(node)) {
            break;
        }
        node = includeShadowHosts && node instanceof ShadowRoot ? node.host :
            node.parentNode;
    }
    return node;
}

/**
 * Disables text selection and dragging, with optional callbacks to specify
 * overrides.
 * @param {function(Event):boolean=} opt_allowSelectStart Unless this function
 *     is defined and returns true, the onselectionstart event will be
 *     suppressed.
 * @param {function(Event):boolean=} opt_allowDragStart Unless this function
 *     is defined and returns true, the ondragstart event will be suppressed.
 */
/* #export */ function disableTextSelectAndDrag(
    opt_allowSelectStart, opt_allowDragStart) {
    // Disable text selection.
    document.onselectstart = function(e) {
        if (!(opt_allowSelectStart && opt_allowSelectStart.call(this, e))) {
            e.preventDefault();
        }
    };

    // Disable dragging.
    document.ondragstart = function(e) {
        if (!(opt_allowDragStart && opt_allowDragStart.call(this, e))) {
            e.preventDefault();
        }
    };
};

```

```

}

/**
 * Check the directionality of the page.
 * @return {boolean} True if Chrome is running an RTL UI.
 */
/* #export */ function isRTL() {
    return document.documentElement.dir === 'rtl';
}

/**
 * Get an element that's known to exist by its ID. We use this instead of just
 * calling getElementById and not checking the result because this lets us
 * satisfy the JSCompiler type system.
 * @param {string} id The identifier name.
 * @return {!HTMLElement} the Element.
 */
/* #export */ function getRequiredElement(id) {
    return assertInstanceOf(
        $(id), HTMLElement, 'Missing required element: ' + id);
}

/**
 * Query an element that's known to exist by a selector. We use this instead of
 * just calling querySelector and not checking the result because this lets us
 * satisfy the JSCompiler type system.
 * @param {string} selectors CSS selectors to query the element.
 * @param {(!Document!!DocumentFragment!!Element)=} opt_context An optional
 * context object for querySelector.
 * @return {!HTMLElement} the Element.
 */
/* #export */ function queryRequiredElement(selectors, opt_context) {
    const element = (opt_context || document).querySelector(selectors);
    return assertInstanceOf(
        element, HTMLElement, 'Missing required element: ' + selectors);
}

/**
 * Creates a new URL which is the old URL with a GET param of key=value.
 * @param {string} url The base URL. There is not sanity checking on the URL so
 * it must be passed in a proper format.
 * @param {string} key The key of the param.
 * @param {string} value The value of the param.
 * @return {string} The new URL.
 */
/* #export */ function appendParam(url, key, value) {

```

```

const param = encodeURIComponent(key) + '=' + encodeURIComponent(value);

if (url.indexOf('?') === -1) {
  return url + '?' + param;
}
return url + '&' + param;
}

/**
 * Creates an element of a specified type with a specified class name.
 * @param {string} type The node type.
 * @param {string} className The class name to use.
 * @return {Element} The created element.
 */
/* #export */ function createElementWithClassName(type, className) {
  const elm = document.createElement(type);
  elm.className = className;
  return elm;
}

/**
 * transitionend does not always fire (e.g. when animation is aborted
 * or when no paint happens during the animation). This function sets up
 * a timer and emulate the event if it is not fired when the timer expires.
 * @param {HTMLElement} el The element to watch for transitionend.
 * @param {number=} opt_timeout The maximum wait time in milliseconds for the
 *   transitionend to happen. If not specified, it is fetched from !ell
 *   using the transitionDuration style value.
 */
/* #export */ function ensureTransitionEndEvent(el, opt_timeout) {
  if (opt_timeout === undefined) {
    const style = getComputedStyle(el);
    opt_timeout = parseFloat(style.transitionDuration) * 1000;

    // Give an additional 50ms buffer for the animation to complete.
    opt_timeout += 50;
  }

  let fired = false;
  el.addEventListener('transitionend', function f(e) {
    el.removeEventListener('transitionend', f);
    fired = true;
  });
  window.setTimeout(function() {
    if (!fired) {
      cr.dispatchSimpleEvent(el, 'transitionend', true);
    }
  }, opt_timeout);
}

```

```

    }
  }, opt_timeOut);
}

/**
 * Alias for document.scrollTop getter.
 * @param {!HTMLDocument} doc The document node where information will be
 *   queried from.
 * @return {number} The Y document scroll offset.
 */
/* #export */ function scrollTopForDocument(doc) {
  return doc.documentElement.scrollTop || doc.body.scrollTop;
}

/**
 * Alias for document.scrollTop setter.
 * @param {!HTMLDocument} doc The document node where information will be
 *   queried from.
 * @param {number} value The target Y scroll offset.
 */
/* #export */ function setScrollTopForDocument(doc, value) {
  doc.documentElement.scrollTop = doc.body.scrollTop = value;
}

/**
 * Alias for document.scrollLeft getter.
 * @param {!HTMLDocument} doc The document node where information will be
 *   queried from.
 * @return {number} The X document scroll offset.
 */
/* #export */ function scrollLeftForDocument(doc) {
  return doc.documentElement.scrollLeft || doc.body.scrollLeft;
}

/**
 * Alias for document.scrollLeft setter.
 * @param {!HTMLDocument} doc The document node where information will be
 *   queried from.
 * @param {number} value The target X scroll offset.
 */
/* #export */ function setScrollLeftForDocument(doc, value) {
  doc.documentElement.scrollLeft = doc.body.scrollLeft = value;
}

/**
 * Replaces '&', '<', '>', '"', and "'" characters with their HTML encoding.

```

```

* @param {string} original The original string.
* @return {string} The string with all the characters mentioned above replaced.
*/
/* #export */ function HTMLEscape(original) {
    return original.replace(/&/g, '&amp;')
        .replace(/</g, '&lt;')
        .replace(/>/g, '&gt;')
        .replace(/"/g, '&quot;')
        .replace(/'/g, '&#39;');
}

/**
* Shortens the provided string (if necessary) to a string of length at most
* |maxLength|.
* @param {string} original The original string.
* @param {number} maxLength The maximum length allowed for the string.
* @return {string} The original string if its length does not exceed
*     |maxLength|. Otherwise the first |maxLength| - 1 characters with '...'
*     appended.
*/
/* #export */ function elide(original, maxLength) {
    if (original.length <= maxLength) {
        return original;
    }
    return original.substring(0, maxLength - 1) + '\u2026';
}

/**
* Quote a string so it can be used in a regular expression.
* @param {string} str The source string.
* @return {string} The escaped string.
*/
/* #export */ function quoteString(str) {
    return str.replace(/([\\.\+*\?[\^\]\$\\)\{\}\=\!\<\>\|\/:])/g, '\\$1');
}

/**
* Calls |callback| and stops listening the first time any event in |eventNames|
* is triggered on |target|.
* @param {!EventTarget} target
* @param {!Array<string>|string} eventNames Array or space-delimited string of
*     event names to listen to (e.g. 'click mousedown').
* @param {function(!Event)} callback Called at most once. The
*     optional return value is passed on by the listener.
*/
/* #export */ function listenOnce(target, eventNames, callback) {

```



```

if (!Array.isArray(eventNames)) {
  eventNames = eventNames.split(/ +/);
}

const removeAllAndCallCallback = function(event) {
  eventNames.forEach(function(eventName) {
    target.removeEventListener(eventName, removeAllAndCallCallback, false);
  });
  return callback(event);
};

eventNames.forEach(function(eventName) {
  target.addEventListener(eventName, removeAllAndCallCallback, false);
});
}

/**
 * @param {!Event} e
 * @return {boolean} Whether a modifier key was down when processing lel.
 */
/* #export */ function hasKeyModifiers(e) {
  return !(e.altKey || e.ctrlKey || e.metaKey || e.shiftKey);
}

/**
 * @param {!Element} el
 * @return {boolean} Whether the element is interactive via text input.
 */
/* #export */ function isTextInputElement(el) {
  return el.tagName === 'INPUT' || el.tagName === 'TEXTAREA';
}

/* #ignore */ console.warn('crbug/1173575, non-JS module files deprecated.');
```

</script>

<script>// Copyright 2014 The Chromium Authors. All rights reserved.  
// Use of this source code is governed by a BSD-style license that can be  
// found in the LICENSE file.

```

const CAPTIVEPORTAL_CMD_OPEN_LOGIN_PAGE = 'openLoginPage';
</script>
<script>// Copyright 2014 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.
```

```

function setupSSLDebuggingInfo() {
  if (loadTimeData.getString('type') !== 'SSL') {

```

```

    return;
}

// The titles are not internationalized because this is debugging information
// for bug reports, help center posts, etc.
appendDebuggingField('Subject', loadTimeData.getString('subject'));
appendDebuggingField('Issuer', loadTimeData.getString('issuer'));
appendDebuggingField('Expires on', loadTimeData.getString('expirationDate'));
appendDebuggingField('Current date', loadTimeData.getString('currentDate'));
appendDebuggingField('PEM encoded chain', loadTimeData.getString('pem'),
    true);
const ctlInfo = loadTimeData.getString('ctl');
if (ctlInfo) {
    appendDebuggingField('Certificate Transparency', ctlInfo);
}

$('error-code').addEventListener('click', toggleDebuggingInfo);
$('error-code').setAttribute('role', 'button');
$('error-code').setAttribute('aria-expanded', false);
}
</script>
<script>// Copyright 2015 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.

'use strict';

// Other constants defined in security_interstitial_page.h.
const SB_BOX_CHECKED = 'boxchecked';
const SB_DISPLAY_CHECK_BOX = 'displaycheckbox';

// This sets up the Extended Safe Browsing Reporting opt-in, either for
// reporting malware or invalid certificate chains. Does nothing if the
// interstitial type is not SAFEBROWSING or SSL or CAPTIVE_PORTAL.
function setupExtendedReportingCheckbox() {
    const interstitialType = loadTimeData.getString('type');
    if (interstitialType !== 'SAFEBROWSING' && interstitialType !== 'SSL' &&
        interstitialType !== 'CAPTIVE_PORTAL') {
        return;
    }

    if (!loadTimeData.getBoolean(SB_DISPLAY_CHECK_BOX)) {
        return;
    }

    if ($('privacy-link')) {

```

```

    $('privacy-link').addEventListener('click', function() {
sendCommand(SecurityInterstitialCommandId.CMD_OPEN_REPORTING_PRIVACY);
    return false;
    });
    $('privacy-link').addEventListener('mousedown', function() {
    return false;
    });
}
$('opt-in-checkbox').checked = loadTimeData.getBoolean(SB_BOX_CHECKED);
$('extended-reporting-opt-in').classList.remove('hidden');

const billing =
    interstitialType === 'SAFEBROWSING' && loadTimeData.getBoolean('billing');

let className = 'ssl-opt-in';
if (interstitialType === 'SAFEBROWSING' && !billing) {
    className = 'safe-browsing-opt-in';
}

$('extended-reporting-opt-in').classList.add(className);

$('body').classList.add('extended-reporting-has-checkbox');

if ($('whitepaper-link')) {
    $('whitepaper-link').addEventListener('click', function(event) {
        sendCommand(SecurityInterstitialCommandId.CMD_OPEN_WHITEPAPER);
    });
}

$('opt-in-checkbox').addEventListener('click', function() {
    sendCommand($('opt-in-checkbox').checked ?
        SecurityInterstitialCommandId.CMD_DO_REPORT :
        SecurityInterstitialCommandId.CMD_DONT_REPORT);
});
}
</script>
<script>// Copyright 2020 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.

'use strict';

// Other constants defined in security_interstitial_page.h.
const SB_DISPLAY_ENHANCED_PROTECTION_MESSAGE =
    'displayEnhancedProtectionMessage';

```

```

// This sets up the enhanced protection message.
function setupEnhancedProtectionMessage() {
  const interstitialType = loadTimeData.getString('type');
  if (interstitialType !== 'SAFEBROWSING' && interstitialType !== 'SSL' &&
      interstitialType !== 'CAPTIVE_PORTAL') {
    return;
  }

  if (!
loadTimeData.getBoolean(SB_DISPLAY_ENHANCED_PROTECTION_MESSAGE)) {
    return;
  }

  if ($('#enhanced-protection-link')) {
    if (mobileNav) {
      // To make sure the touch area of the link is larger than the
      // minimum touch area for accessibility, make the whole block tappable.
      $('#enhanced-protection-message').addEventListener('click', function() {
        sendCommand(SecurityInterstitialCommandId
                    .CMD_OPEN_ENHANCED_PROTECTION_SETTINGS);
        return false;
      });
    } else {
      $('#enhanced-protection-link').addEventListener('click', function() {
        sendCommand(SecurityInterstitialCommandId
                    .CMD_OPEN_ENHANCED_PROTECTION_SETTINGS);
        return false;
      });
    }
  }
  $('#enhanced-protection-message').classList.remove('hidden');

  const billing =
    interstitialType === 'SAFEBROWSING' && loadTimeData.getBoolean('billing');

  let className = 'ssl-enhanced-protection-message';
  if (interstitialType === 'SAFEBROWSING' && !billing) {
    className = 'safe-browsing-enhanced-protection-message';
  }

  $('#enhanced-protection-message').classList.add(className);
}
</script>
<script>// Copyright 2015 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be

```

// found in the LICENSE file.

```
let mobileNav = false;
```

```
/**
```

```
 * For small screen mobile the navigation buttons are moved
 * below the advanced text.
```

```
*/
```

```
function onResize() {
```

```
  const helpOuterBox = document.querySelector('#details');
```

```
  const mainContent = document.querySelector('#main-content');
```

```
  const mediaQuery = '(min-width: 240px) and (max-width: 420px) and ' +
    '(min-height: 401px), ' +
    '(max-height: 560px) and (min-height: 240px) and ' +
    '(min-width: 421px)';
```

```
  const detailsHidden = helpOuterBox.classList.contains(HIDDEN_CLASS);
```

```
  const runnerContainer = document.querySelector('.runner-container');
```

```
  // Check for change in nav status.
```

```
  if (mobileNav !== window.matchMedia(mediaQuery).matches) {
    mobileNav = !mobileNav;
```

```
  // Handle showing the top content / details sections according to state.
```

```
  if (mobileNav) {
```

```
    mainContent.classList.toggle(HIDDEN_CLASS, !detailsHidden);
```

```
    helpOuterBox.classList.toggle(HIDDEN_CLASS, detailsHidden);
```

```
    if (runnerContainer) {
```

```
      runnerContainer.classList.toggle(HIDDEN_CLASS, !detailsHidden);
```

```
    }
```

```
  } else if (!detailsHidden) {
```

```
    // Non mobile nav with visible details.
```

```
    mainContent.classList.remove(HIDDEN_CLASS);
```

```
    helpOuterBox.classList.remove(HIDDEN_CLASS);
```

```
    if (runnerContainer) {
```

```
      runnerContainer.classList.remove(HIDDEN_CLASS);
```

```
    }
```

```
  }
```

```
}
```

```
}
```

```
function setupMobileNav() {
```

```
  window.addEventListener('resize', onResize);
```

```
  onResize();
```

```
}
```

```
document.addEventListener('DOMContentLoaded', setupMobileNav);
</script>
<script>// Copyright 2017 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.
```

```
// This is the shared code for security interstitials. It is used for both SSL
// interstitials and Safe Browsing interstitials.
```

```
/**
 * @typedef {{
 *   dontProceed: function(),
 *   proceed: function(),
 *   showMoreSection: function(),
 *   openHelpCenter: function(),
 *   openDiagnostic: function(),
 *   reload: function(),
 *   openDateSettings: function(),
 *   openLogin: function(),
 *   doReport: function(),
 *   dontReport: function(),
 *   openReportingPrivacy: function(),
 *   openWhitepaper: function(),
 *   reportPhishingError: function(),
 * }}
 */
// eslint-disable-next-line no-var
var certificateErrorPageController;

// Should match security_interstitials::SecurityInterstitialCommand
/** @enum {number} */
const SecurityInterstitialCommandId = {
  CMD_DONT_PROCEED: 0,
  CMD_PROCEED: 1,
  // Ways for user to get more information
  CMD_SHOW_MORE_SECTION: 2,
  CMD_OPEN_HELP_CENTER: 3,
  CMD_OPEN_DIAGNOSTIC: 4,
  // Primary button actions
  CMD_RELOAD: 5,
  CMD_OPEN_DATE_SETTINGS: 6,
  CMD_OPEN_LOGIN: 7,
  // Safe Browsing Extended Reporting
  CMD_DO_REPORT: 8,
  CMD_DONT_REPORT: 9,
  CMD_OPEN_REPORTING_PRIVACY: 10,
```

```
CMD_OPEN_WHITEPAPER: 11,  
// Report a phishing error.  
CMD_REPORT_PHISHING_ERROR: 12,  
// Open enhanced protection settings.  
CMD_OPEN_ENHANCED_PROTECTION_SETTINGS: 13,  
};
```

```
const HIDDEN_CLASS = 'hidden';
```

```
/**  
 * A convenience method for sending commands to the parent page.  
 * @param {SecurityInterstitialCommandId} cmd The command to send.  
 */  
function sendCommand(cmd) {  
  if (window.certificateErrorPageController) {  
    switch (cmd) {  
      case SecurityInterstitialCommandId.CMD_DONT_PROCEED:  
        certificateErrorPageController.dontProceed();  
        break;  
      case SecurityInterstitialCommandId.CMD_PROCEED:  
        certificateErrorPageController.proceed();  
        break;  
      case SecurityInterstitialCommandId.CMD_SHOW_MORE_SECTION:  
        certificateErrorPageController.showMoreSection();  
        break;  
      case SecurityInterstitialCommandId.CMD_OPEN_HELP_CENTER:  
        certificateErrorPageController.openHelpCenter();  
        break;  
      case SecurityInterstitialCommandId.CMD_OPEN_DIAGNOSTIC:  
        certificateErrorPageController.openDiagnostic();  
        break;  
      case SecurityInterstitialCommandId.CMD_RELOAD:  
        certificateErrorPageController.reload();  
        break;  
      case SecurityInterstitialCommandId.CMD_OPEN_DATE_SETTINGS:  
        certificateErrorPageController.openDateSettings();  
        break;  
      case SecurityInterstitialCommandId.CMD_OPEN_LOGIN:  
        certificateErrorPageController.openLogin();  
        break;  
      case SecurityInterstitialCommandId.CMD_DO_REPORT:  
        certificateErrorPageController.doReport();  
        break;  
      case SecurityInterstitialCommandId.CMD_DONT_REPORT:  
        certificateErrorPageController.dontReport();  
        break;
```

```

    case SecurityInterstitialCommandId.CMD_OPEN_REPORTING_PRIVACY:
        certificateErrorPageController.openReportingPrivacy();
        break;
    case SecurityInterstitialCommandId.CMD_OPEN_WHITEPAPER:
        certificateErrorPageController.openWhitepaper();
        break;
    case SecurityInterstitialCommandId.CMD_REPORT_PHISHING_ERROR:
        certificateErrorPageController.reportPhishingError();
        break;
    case
SecurityInterstitialCommandId.CMD_OPEN_ENHANCED_PROTECTION_SETTINGS:
        certificateErrorPageController.openEnhancedProtectionSettings();
        break;
    }
    return;
}
//
window.domAutomationController.send(cmd);
//
//
}

/**
 * Call this to stop clicks on <a href="#"> links from scrolling to the top of
 * the page (and possibly showing a # in the link).
 */
function preventDefaultOnPoundLinkClicks() {
    document.addEventListener('click', function(e) {
        const anchor = findAncestor(** @type {Node} */ (e.target), function(el) {
            return el.tagName === 'A';
        });
        // Use getAttribute() to prevent URL normalization.
        if (anchor && anchor.getAttribute('href') === '#') {
            e.preventDefault();
        }
    });
}

//

//
</script>
<script>// Copyright 2014 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.

```



```
// This is the shared code for the new (Chrome 37) security interstitials. It is
// used for both SSL interstitials and Safe Browsing interstitials.
```

```
let expandedDetails = false;
let keyPressState = 0;
```

```
/**
 * This allows errors to be skipped by typing a secret phrase into the page.
 * @param {string} e The key that was just pressed.
 */
function handleKeypress(e) {
  // HTTPS errors are serious and should not be ignored. For testing purposes,
  // other approaches are both safer and have fewer side-effects.
  // See https://goo.gl/ZcZixP for more details.
  const BYPASS_SEQUENCE = window.atob('dGhpc2lzdW5zYWZl');
  if (BYPASS_SEQUENCE.charCodeAt(keyPressState) === e.keyCode) {
    keyPressState++;
    if (keyPressState === BYPASS_SEQUENCE.length) {
      sendCommand(SecurityInterstitialCommandId.CMD_PROCEED);
      keyPressState = 0;
    }
  } else {
    keyPressState = 0;
  }
}
```

```
/**
 * This appends a piece of debugging information to the end of the warning.
 * When complete, the caller must also make the debugging div
 * (error-debugging-info) visible.
 * @param {string} title The name of this debugging field.
 * @param {string} value The value of the debugging field.
 * @param {boolean=} fixedWidth If true, the value field is displayed fixed
 * width.
 */
function appendDebuggingField(title, value, fixedWidth) {
  // The values input here are not trusted. Never use innerHTML on these
  // values!
  const spanTitle = document.createElement('span');
  spanTitle.classList.add('debugging-title');
  spanTitle.innerText = title + ': ';

  const spanValue = document.createElement('span');
  spanValue.classList.add('debugging-content');
  if (fixedWidth) {
    spanValue.classList.add('debugging-content-fixed-width');
```

```

}
spanValue.innerText = value;

const pElem = document.createElement('p');
pElem.classList.add('debugging-content');
pElem.appendChild(spanTitle);
pElem.appendChild(spanValue);
$('error-debugging-info').appendChild(pElem);
}

function toggleDebuggingInfo() {
  const hiddenDebug = $('error-debugging-info').classList.toggle(HIDDEN_CLASS);
  $('error-code').setAttribute('aria-expanded', !hiddenDebug);
}

function setupEvents() {
  const overridable = loadTimeData.getBoolean('overridable');
  const interstitialType = loadTimeData.getString('type');
  const ssl = interstitialType === 'SSL';
  const captivePortal = interstitialType === 'CAPTIVE_PORTAL';
  const badClock = ssl && loadTimeData.getBoolean('bad_clock');
  const lookalike = interstitialType === 'LOOKALIKE';
  const billing =
    interstitialType === 'SAFEBROWSING' && loadTimeData.getBoolean('billing');
  const originPolicy = interstitialType === 'ORIGIN_POLICY';
  const blockedInterception = interstitialType === 'BLOCKED_INTERCEPTION';
  const legacyTls = interstitialType === 'LEGACY_TLS';
  const insecureForm = interstitialType === 'INSECURE_FORM';
  const httpsOnly = interstitialType === 'HTTPS_ONLY';
  const hidePrimaryButton = loadTimeData.getBoolean('hide_primary_button');
  const showRecurrentErrorParagraph = loadTimeData.getBoolean(
    'show_recurrent_error_paragraph');

  if (ssl || originPolicy || blockedInterception || legacyTls) {
    $('body').classList.add(badClock ? 'bad-clock' : 'ssl');
    if (loadTimeData.valueExists('errorCode')) {
      $('error-code').textContent = loadTimeData.getString('errorCode');
      $('error-code').classList.remove(HIDDEN_CLASS);
    }
  } else if (captivePortal) {
    $('body').classList.add('captive-portal');
  } else if (billing) {
    $('body').classList.add('safe-browsing-billing');
  } else if (lookalike) {
    $('body').classList.add('lookalike-url');
  } else if (insecureForm) {

```

```

    $('body').classList.add('insecure-form');
} else if (httpsOnly) {
    $('body').classList.add('https-only');
} else {
    $('body').classList.add('safe-browsing');
    // Override the default theme color.
    document.querySelector('meta[name=theme-color]').setAttribute('content',
        'rgb(217, 48, 37)');
}

$('icon').classList.add('icon');

if (legacyTls) {
    $('icon').classList.add('legacy-tls');
}

if (hidePrimaryButton) {
    $('primary-button').classList.add(HIDDEN_CLASS);
} else {
    $('primary-button').addEventListener('click', function() {
        switch (interstitialType) {
            case 'CAPTIVE_PORTAL':
                sendCommand(SecurityInterstitialCommandId.CMD_OPEN_LOGIN);
                break;

            case 'SSL':
            case 'LEGACY_TLS':
                if (badClock) {

sendCommand(SecurityInterstitialCommandId.CMD_OPEN_DATE_SETTINGS);
                } else if (overridable) {
                    sendCommand(SecurityInterstitialCommandId.CMD_DONT_PROCEED);
                } else {
                    sendCommand(SecurityInterstitialCommandId.CMD_RELOAD);
                }
                break;

            case 'SAFEBROWSING':
            case 'ORIGIN_POLICY':
                sendCommand(SecurityInterstitialCommandId.CMD_DONT_PROCEED);
                break;
            case 'HTTPS_ONLY':
            case 'INSECURE_FORM':
            case 'LOOKALIKE':
                sendCommand(SecurityInterstitialCommandId.CMD_DONT_PROCEED);
                break;
        }
    });
}

```

```

        default:
            throw 'Invalid interstitial type';
        }
    });
}

if (lookalike || insecureForm || httpsOnly) {
    const proceedButton = 'proceed-button';
    $(proceedButton).classList.remove(HIDDEN_CLASS);
    $(proceedButton).textContent = loadTimeData.getString('proceedButtonText');
    $(proceedButton).addEventListener('click', function(event) {
        sendCommand(SecurityInterstitialCommandId.CMD_PROCEED);
    });
}

if (lookalike) {
    // Lookalike interstitials with a suggested URL have a link in the title:
    // "Did you mean <link>example.com</link>?". Handle those clicks. Lookalike
    // interstitials without a suggested URL don't have this link.
    const dontProceedLink = 'dont-proceed-link';
    if ($(dontProceedLink)) {
        $(dontProceedLink).addEventListener('click', function(event) {
            sendCommand(SecurityInterstitialCommandId.CMD_DONT_PROCEED);
        });
    }
}

if (overridable) {
    const overrideElement = billing ? 'proceed-button' : 'proceed-link';
    // Captive portal page isn't overridable.
    $(overrideElement).addEventListener('click', function(event) {
        sendCommand(SecurityInterstitialCommandId.CMD_PROCEED);
    });

    if (ssl) {
        $(overrideElement).classList.add('small-link');
    } else if (billing) {
        $(overrideElement).classList.remove(HIDDEN_CLASS);
        $(overrideElement).textContent =
            loadTimeData.getString('proceedButtonText');
    }
} else if (!ssl) {
    $('final-paragraph').classList.add(HIDDEN_CLASS);
}

```

```

if (!ssl || !showRecurrentErrorParagraph) {
    $('recurrent-error-message').classList.add(HIDDEN_CLASS);
} else {
    $('body').classList.add('showing-recurrent-error-message');
}

if ($('#diagnostic-link')) {
    $('#diagnostic-link').addEventListener('click', function(event) {
        sendCommand(SecurityInterstitialCommandId.CMD_OPEN_DIAGNOSTIC);
    });
}

if ($('#learn-more-link')) {
    $('#learn-more-link').addEventListener('click', function(event) {
        sendCommand(SecurityInterstitialCommandId.CMD_OPEN_HELP_CENTER);
    });
}

if (captivePortal || billing || lookalike || insecureForm || httpsOnly) {
    // Captive portal, billing, lookalike pages, insecure form, and
    // HTTPS only mode interstitials don't have details buttons.
    $('details-button').classList.add('hidden');
} else {
    $('details-button')
        .setAttribute(
            'aria-expanded', !$('#details').classList.contains(HIDDEN_CLASS));
    $('details-button').addEventListener('click', function(event) {
        const hiddenDetails = $('#details').classList.toggle(HIDDEN_CLASS);
        $('details-button').setAttribute('aria-expanded', !hiddenDetails);

        if (mobileNav) {
            // Details appear over the main content on small screens.
            $('main-content').classList.toggle(HIDDEN_CLASS, !hiddenDetails);
        } else {
            $('main-content').classList.remove(HIDDEN_CLASS);
        }

        $('details-button').innerText = hiddenDetails ?
            loadTimeData.getString('openDetails') :
            loadTimeData.getString('closeDetails');
        if (!expandedDetails) {
            // Record a histogram entry only the first time that details is opened.
            sendCommand(SecurityInterstitialCommandId.CMD_SHOW_MORE_SECTION);
            expandedDetails = true;
        }
    });
}

```

```
}

if ($('#report-error-link')) {
  $('#report-error-link').addEventListener('click', function(event) {

sendCommand(SecurityInterstitialCommandId.CMD_REPORT_PHISHING_ERROR);
  });
}

if (lookalike) {
  console.log(
    'Chrome has determined that ' +
    loadTimeData.getString('lookalikeRequestHostname') +
    ' could be fake or fraudulent.\n\n' +
    'If you believe this is shown in error please visit ' +
    'https://g.co/chrome/lookalike-warnings');
}

preventDefaultOnPoundLinkClicks();
setupExtendedReportingCheckbox();
setupEnhancedProtectionMessage();
setupSSLDebuggingInfo();
document.addEventListener('keypress', handleKeypress);
}

document.addEventListener('DOMContentLoaded', setupEvents);
</script>
</head>

-----


BMVEUAAADcRDf/ZmbcRjrjVTn/
gIDdRETdRDzZQzbXQzXWQzbXQjbWQzXZRDbbRDnWQjXWQzXYSDvbRTjcRTjbRT
fcRjfcRTjcRTjdRjncRTfdRTndRTfdRDrbRTjcRDnbRDfbRDjbRjfcRjfbRTjcRTjdRTjbRjjcR
TjcRDjcRjncRTncRTndRDnbRTjcRDfZQzbcRTfgSTncRDfcRjjZQzjcRTfVRDbcRDjcRDj
WQzXeRzvbRDjXRDXXQzbXQzbbbRDfeSDvWQjbVQjXIPjLOQDPXQjbCPDDNPzPUQ
TXRQTS5OS7QQDTUQjW3OS7SQTTPQDTFPDHJPjK2OC26OS7HPjHOPzPLPjLMPj
PRQDTGPDHTQTTEPTHLpzPGPTG7Oi/HPTLKPjLTQTXYQza9Oi/
MPzPFPTHDPDHBPDC/OzC+Oy+8Oi/AOzDWQjX////
bRDd3undHAAAAQnRSTIMA2AUWCQIPHj39wvbO8DH64ifqqYFmtrVMc1IKS5x0nY6P
WKqbjYDpZXWCZ1py8Jv9McJXV
+KA9qioPc5l6Y36J7VmcHe8AAAFWUIEQVR4XuzWS4rCQBSG0euz56lSgiaEjHwgGh
AhDnRF3/6HDY1la5WPjP4a3LOKY28555xzzjnnnHPOuSyzpPR7vb6lZAUrS8hgB7uBp
```

aMEKC0Zhz3A/mCpaPjTWCK23GwtCcMjN8ehpWDN3doS8HPi7vRjejX/  
1CbX8qA1sdGZB  
+eRaW14sjGp8YQnk7EpVQQqE7peCFyupjMnYm4yGVGZ7q1EyTZbEEche2uUbLMIL  
5W6t4Zkm22lkm02561c89aQbLNTppgq3hqSbbbmo1r41rhW8NaAaLMzvjlTvDUg2WzF  
lyrBWwOCzc6Jkm12QQcL3Vtlmy3opFC9VbbZJR0tNW  
+Vbbahs0b41rhc8FbVZqdEyTb724t5/bYNA3G4e  
+80NYI0gGFkvaR779KKZUWuFKe7nllsT5X//2M5VMZiZB9DQj74xW8ffrwjP90Mb/  
07Vf5CbXYJg0BtO4toKS9vhYHGY1vDZg28FQY6tBZls8tYBehwNLtyt1nhrTDQaDQcW  
Aux2SJWAxpOBpWMWSvm4q0w0Gg4nFQqFTd/m72HIYBYQJV  
+w83bZu9jRaDJYEB4osjJ02aFt8JASUBRq  
+PlarMrWBGi8lQajVanXA5kopUcvBUEGrCAWhSoXs3PZtewKhA/  
MMbTbcpEa7l4KwwURZSHANnVnGz2CVYGmg6oZ1u1XGy2hNWBCA8BogE1m7ZI  
+ShNVMrdW2Wg/v  
+Amr2eRYCcGLBZU2+FgcSBESDfdZxdwGbNvBUGihKgnk1OjPAEwS5gsybeCgNNdT  
QLyAtqtRCwWQNvhYH4ndjtNnlAnlet1uIQsFI9b4WBpgNyaUCEJ45DwGa1vRUGanU6n  
McmB+ZSnlosES3nvm/tUpGm1tFPd5DDAyKFBjGpzRaxSjW5J0o8/MAQ4ZEyKua/b  
+0Np175blMERDuaECFBZGqzBaxY9iAjIMbDK01U0OVZxcplE6BljLzFRixgQDwflCJaX  
cC+1ToKyOYHFvCOljPiNmvrTBRI  
+oQoGTk2Z1YQyleiWhlEftWnx8Yf8RcyiMCEkyhic2u4xOWSw9MBBQENTQFI83a  
+iL2rdgpJ1rms45mByYzhbDNwt6qTtTIQC7r6FT/CLRQ02ZLWKc8OmK  
+LzooCykhKpl4q7p+7B/d0SjNggRbqGOzm1gPqL3PX3niZakOQsenf1PDWzWAxr  
+JBtEDQxnnJTISNmvrufBK75t45bORBNGSobcqb9DqBCjdQOI5E370xthbYaDRiljRDx  
KQwJk9a+o2u431gYZERBo/kcBlfvJ/TrSt6K1b+kDUHMkra2V3j5zRlprNbmADILbQ65S/  
z2ggyY82zL0VXsdQnnLdhSOKQzWbLWADIMpDgOrd3q958QiigrG3wusYzmNbXmY4s  
h+tangrVJ2Dgy97X9v0CmlLzzIHcj3ZPTL  
+h6DN7mhYR5nxHI4mtKNbLCamaX9QDDKFO6C36hDttcdJQFGLeTWRIlupocGOj62c  
Bb9WqesLTFwfm000MQgqz9IDLW  
+Hve35HM9Fnqw9HetBkNsF6+Yaet8Jf0+xbka0XbYspSMlg  
+5D8/8psnqdYv3qso1vsS9Hy6SaGQ6AYHP9ngLdqllVpilB8RygRQjGEdOsc4K26RGzk  
6YTxjhbDDdzXcfwC8Fbd8glPnR4Y62gBAM/  
a1WybfYVNyyUBiZFPXYCAH70GvFW7nFRHH7Egyl8uAd6qXZ7NAqoilG6ZKuBH184D  
3qpdAQIIWp0p9dE7wFv1q8Y6+njLoPI  
+9P4C4K0GRKSjgTyywvoAeKtBxVWU6YhorovcvA14q0HtouwU0Fw/+jzN8w/cQ/  
zg6ug2/QAAAABJRU5ErkJggg==

-----

```
<div class="interstitial-wrapper">
  <div id="main-content" class="hidden">
    <div class="icon" id="icon"></div>
    <div id="main-message">
      <h1>Your connection is not private</h1>
      <p>Attackers might be trying to steal your information from
<strong>accounts.shopify.com</strong> (for example, passwords, messages, or credit
```

cards). [Learn more](#)

net::ERR\_CERT\_DATE\_INVALID

Subject:accounts.shopify.com

Issuer:R3

Expires on:Nov 13, 2021

Current date:Oct 5, 2021

PEM encoded chain:-----BEGIN CERTIFICATE-----  
MIIF2DCCBMCGAwIBAgISA3IP  
+wuRtzmomjRFbuWpyrsCMA0GCSqGSIb3DQEBCwUA  
MDIx CzAJBgNVBAYTAIV  
TMRy wFAYDVQQKEw1MZXRQncyBFbmNyeXB0MQswCQYDVQQD  
EwJSMzAeFw  
0yMTA4MTUxNjE3MDNaFw0yMTEyMTMxNjE3MDFaMB8xHTAbBgNVBAMT  
FGFj  
Y291bnRzLnNob3BpZnkuY29tMIIlBjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB  
CgK  
CAQEAmNO6kENQqqUM0WgozXecZGcPzrlyEbrU/  
N7mdStXBNjXN9WAEbLSTGrw  
gkyShj1dbDXVJudJ7YdQXJrl/  
KJbm87u8a1HEZI4Dmj1FxBIAd+Gs1RsFdQd0nEw  
5Zgi0Sojj6d87SEqpS2i0Vz6V  
+hTrbuy+XXP5+J8piNiRp2lkrZQkWKaifZaLyvH  
w2S2OFtEJxH3SSoKfsA/  
ST4ZPRDFlppsXgmpFf7sU5u1E3XckNjxl220Q29/  
vtls  
ErLP9noTy1wdKzXJYbu4IT6nVOSs  
+TGLQBcBCfNqyaG19jMrL299G4QyNy2h6uWX  
MwPm/Opq3qk3Bnl5uZgyV  
+mx7EMe1QIDAQABo4IC+TCCAvUwDgYDVR0PAQH/  
BAQD  
AgWgMB0GA1UdJQQWMBQGCCsGAQUFBwMBBggrBgEFBQcDAjAMBg  
NVHRMBAf8EAjAA  
MB0GA1UdDgQWBBR/TRz5xF8Ma/  
wcElldjg8fyl3L3zAfBgNVHSMEGDAWgBQULrMX  
t1hWy65QCUDmH6+dixTCxjBVB  
ggrBgEFBQcBAQRJMEcwIQYIKwYBBQUHMAAGGFWh0  
dHA6Ly9yMy5vLmxlbmN  
yLm9yZzAiBggrBgEFBQcwAoYWAHR0cDovL3IzLmkubGVu  
Y3lub3JnLzCBYQYDV  
R0RBIHBMIG  
+gh5hY2NvdW50cy1kZW1vLnNob3BpZnlibG91  
ZC5jb22CFGFjY291bnRzLnNob3B  
pZnkuY29tgh9jdXN0b21lcnMtZGVtby5zaG9w  
aWZ5Y2xvdWQuY29tgiZpZGVudGlo  
eS1tYWluLWNsdXN0ZXluc2hvcGlmeWNsb3V  
kLmNvbYlIaWRIbnRpdHktbWVyY2h  
hbnQuc2hvcGlmeWNsb3VklmNvbYlIaWRIbnR  
pdHktbWVyY2h  
bTBMbG9VHSAERTBDMAgGBmeBDAECATA3BgsrBgEE  
AYLfEwEBATAoMCYGC  
CsGAQUFBwIBFhpodHRwOi8vY3BzLmxldHNlbnNyeXB0Lm9y  
ZzCCAQMGCisGA  
QQB1nkCBAIEgfQEgEA7wB1AESUZS6w7s6vxEAH2Kj+KMDa5oK+  
2MsxtT/  
TM5a1toGoAAABe0rRIFkAAAQDAEYwRAIgc9G4Kf  
+YE3BwY4IHwrPfCyx  
HBfmiWZVYY3syP6PMtQCICbrsrvVPy0eYoa  
+way7bJOxuRABwT4Sfd4Nd14ITkrE  
AHYAfT7y+I//  
iFVoJMLAyp5SiXkrxQ54CX8uapdomX4i8NcAAAF7StEgegAABAMA  
RzBFAiBQFP/  
Jq9s0RX2lddzPRWorKmK2vHb0gBjdeR4DafmJfglhAOGXW8pNxx  
+g  
AgkEhlLco7khwJUOAcDqcKCIG9HS9FOJ8MA0GCSqGSIb3DQEBCwUAA4IBAQ



BcKanD<br>DWicj0QRpqqbMyHi1uwU0vJoOuCXAB5TMevD1OGhl2HGfV8blgxWlfOU  
LxGoEd/k<br>SA0t9X94AFhu+FhBQnrssUH/D  
+xQFKSbaZcQIB5j1nEFNdtpokxW5JyhjS5MXiYO<br>dWfgQcEhx3HlnYVT  
+Z1CpNslgCSOaoOawx+b1y1/Kj5ZIEld2rZBIOxh+/ifPP9h<br>Z8Y/kug  
+O2Vu7xfPWn4RtVeG39FpK3FcbOxkTA09K1CHtPQf+KZ/  
Ou5BcLn8BJi5<br>MVXdjQTeKuzilyV7QM+dAxHn3/  
reMXmlGengm7nj19K2P3I7Z70dX1sEwUXQR8Wr<br>/+lfVfwGqqjAuskK<br>-----END  
CERTIFICATE-----<br>-----BEGIN CERTIFICATE-----  
<br>MIIFFjCCAv6gAwIBAgIRAJErCErPDBinU/  
bWLiWnX1owDQYJKoZIhvcNAQELBQAw<br>TzELMAkGA1UEBhMCVVMxKTAnBgN  
VBAoTIEludGVybWV0IFNIY3VyaXR5IFJlc2Vh<br>cmNoIEdyb3VwMRUwEwYDVQQD  
EwxJU1JHIFJvb3QgWDEwHhcNMjAwOTA0MDAwMDAw<br>WhcNMjUwOTE1MTYw  
MDAwWjAyMQswCQYDVQQGEwJVUzEWMBQGA1UEChMNTGV0J3Mg<br>RW5jcml  
wdDELMAkGA1UEAxMCUjMwggiEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK<br>  
AoIBAQC7AhUozPaglNMPEuyNVZLD+ILxmaZ6QoinXSaqtSu5xUyxr45r  
+XXlo9cP<br>R5QUVTVXjJ6oojkZ9YI8QqlObvU7wy7bjcCwXPNZOOftz2nwWgsbvsCU  
JCWH+jdx<br>sxPnHKzhm+/  
b5DtFukWWqcFTzjTIUu61ru2P3mBw4qVUq7ZtDpelQDRrK9O8Zutm<br>NHZ6a4uPVy  
mZ+DAXXbpyb/uBxa3Shlg9F8fnCbvxK/eG3MHacV3URuPMrSXBiLxg<br>Z3Vms/  
EY96Jc5IP/Ooi2R6X/ExjqmAl3P51T+c8B5fWmcBcUr2Ok/5mzk53cU6cG<br>/  
kiFHaFpriV1uxPMUgP17VGhi9sVAgMBAAGjggEIMIIBBDAOBgNVHQ8BAf8EBAMC<br>  
>AYYwHQYDVR0IBBYwFAYIKwYBBQUHAWIGCCsGAQUFBwMBMBIGA1UdEwEB/  
wQIMAYB<br>Af8CAQAwHQYDVR0OBBYEFBQusxe3WFbLrIAJQOYfr52LFMLGMB8  
GA1UdIwQYMBaA<br>FHm0WeZ7tuXkAXOACIjIGlj26ZtuMDIGCCsGAQUFBwEBBCY  
wJDAiBggrBgEFBQcw<br>AoYWaHR0cDovL3gxLmkubGVuY3lub3JnLzAnBgNVHR8EI  
DAeMBygGqAYhhZodHRw<br>Oi8veDEuYy5sZW5jci5vcmcvMCIGA1UdIAQbMBkwCA  
YgZ4EMAQIBMA0GCysGAQQB<br>gt8TAQEBMA0GCSqGSIb3DQEBCwUAA4ICAQ  
CFyk5HPqP3hUSFvNVneLKYy611TR6W<br>PTNlclQtdq+34IL9fzLdwALduO/  
ZelN7klJ+m74uyA+eitRY8kc607TkC53wl<br>ikfmZW4/RvTZ8M6UK  
+5UzhK8jCdLuMGYL6KvzXGRSgi3yLgjewQtCPkIVz6D2QQz<br>CkcheAmCJ8MqyJu  
5zlzyZMjAvnnAT45tRAxekrsu94sQ4egdRCnbWSDtY7kh  
+Blm<br>IJNXoB1IBMEKlq4QDUOXoRgffuDghje1WrG9ML+Hbisq/  
yFOGwXD9RiX8F6sw6W4<br>avAuvDsue5L3sz85K+EC4Y/wFVDNvZo4TYXao6Z0f  
+IQKc0t8DQYzk1OXVu8rp2<br>yJMC6alLbBfODALZvYH7n7do1AZIs4l9d1P4jnkDrQox  
B3UqQ9hVI3LEKQ73xF1O<br>yK5GhDDX8oVfGKF5u  
+declsH4YaTw7mP3GFxJSqv3+0IUfJoi5Lc5da149p90lds<br>hCExroL1+7mrylkXPeF  
M5TgO9r0rvZaBFOvV2z0gp35Z0+L4WPIbuEjN/lxPFin+<br>HIUjr8gRsl3qfJOQFy/  
9rKIJR0Y/8Omwt/  
8oTWgy1mdeHmnmjk7j1nYsvC9JSQ6Zv<br>MldITTKB3zhThV1+XWYp6rjd5JW1zbVW  
EkLNxE7GJThEUG3szgBVGP7pSWTUTsqX<br>nLRbwHOoq7hHwg==<br>-----END  
CERTIFICATE-----<br>-----BEGIN CERTIFICATE-----  
<br>MIIFYDCCBEigAwIBAgIQQAf3ITfU6UK47naqPGQKtzANBgkqhkiG9w0BAQsFAD  
A/  
<br>MSQwlgYDVQQKEExtEaWdpdGFsIFNpZ25hdHVyZSBUCnVzdCBDby4xZzAVBgNV  
BAMT<br>DkRRTVCBSb290IENBIFgzMB4XDTIxMDEyMDE5MTQwM1oXDTI0MDkzMD  
E4MTQwM1ow<br>TzELMAkGA1UEBhMCVVMxKTAnBgNVBAoTIEludGVybWV0IFNIY

3VyaXR5IFJlc2Vh<br>cmNoIEdyb3VwMRUwEwYDVQQDEwxJU1JHIFJvb3QgWDEwg  
gliMA0GCSqGSIb3DQEB<br>AQUAA4ICDwAwggIKAoICAQCt6CRz9BQ385ueK1coHI  
e+3LffOJCMbjzmV6B493XC<br>ov71am72AE8o295ohmxEk7axY/0UEmu/  
H9LqMZshftEzPLpI9d1537O4/xLxIZpL<br>wYqGcWIKZmZsj348cL  
+tKSIG8+TA5oCu4kuPt5l+IAOf00eXfJIII1PoOK5PCm  
+D<br>LtFJV4yAdLbaL9A4jXsDcCEbdfIwPPqPrt3aY6vrFk/CjhFLfs8L6P  
+1dy70sntK<br>4EwSJQxwjQMpoOFTJOwT2e4ZvxCzSow/  
iaNhUd6shweU9GNx7C7ib1uYgeGJXDR5<br>bHbvO5BieebbpJovJsXQEOEO3tkQjhb  
7t/eo98flAgeYjzYlIefiN5YNNnWe+w5y<br>sR2bvAP5SQXYgd0FtCrWQemsAXaVCg/  
Y39W9Eh81LygXbNKYwagJZHduRze6zqxZ<br>Xmidf3LWicUGQSk  
+WT7dJvUkyRGnWqNMQB9GoZm1pzpRboY7nn1ypxIFeFntPIF4<br>FQsDj43QLwWy  
PntKHEtzBRL8xurgUBN8Q5N0s8p0544fAQjQMNRbcTa0B7rBMDBc<br>SLeCO5imfW  
CKoqMpgsy6vYMEG6KDA0Gh1gXxG8K28Kh8hjtGqEgqiNx2mna/  
H2ql<br>PRmP6zjzZN7IKw0KKP/32+IVQtQi0Cdd4Xn+GOdwIK1O5tmLOsbdJ1Fu/  
7xk9TND<br>TwIDAQABo4IBRjCCAUIwDwYDVR0TAQH/BAUwAwEB/  
zAOBgNVHQ8BAf8EBAMCAQYw<br>SwYIKwYBBQUHAQEEPzA9MDsGCCsGAQUF  
BzAChi9odHRwOi8vYXBwcy5pZGVudHJ1<br>c3QuY29tL3Jvb3RzL2RzdHJvb3RjYXgz  
LnA3YzAfBgNVHSMEGDAWgBTEp7Gkeyxx<br>  
+tvhS5B1/8QVYIWJEDBUBgNVHSAETTLBMAgGBmeBDAECATA/  
BgsrBgEEAYLfEwEB<br>ATAwMC4GCCsGAQUFBwIBFiJodHRwOi8vY3BzLnJvb3Qte  
DEubGV0c2VuY3J5cHQw<br>b3JnMDwGA1UdHwQ1MDMwMaAvoC2GK2h0dHA6Ly9  
jcmwuaWRlbnRydXN0LmNvbS9E<br>U1RST09UQ0FYM0NSTC5jcmwwHQYDVR0OB  
BYEFHm0WeZ7tuXkAXOACljlGlj26Ztu<br>MA0GCSqGSIb3DQEBECwUAA4IBAQAkCw  
Bslm7/DILQrt2M51oGrS+o44+/  
yQoDFVDC<br>5WxCu2+b9LRPwkSICHXM6webFGJueN7sJ7o5XPWioW5WIHAQU7  
G75K/  
QosMrAdSW<br>9MUgNTP52GE24HGNTLi1qoJFlcDyqSMo59ahy2cl2qBDLKobkx/  
J3vWraV0T9VuG<br>WCLKTVXkcGdtwlfFRjIbZ4pYg1htmf5X6DYO8A4jqv2II9DjXA6U  
SbW1FzXSLr9O<br>he8Y4IWS6wY7bCkjCWDcRQJMEhg76fsO3txE  
+FiYruq9RUWhiF1myv4Q6W  
+CyBFC<br>Dfvp7OOGAN6dEOM4+qR9sdjoSYKEBpsr6GtPAQw4dy753ec5<br>-----  
END CERTIFICATE-----<br>-----BEGIN CERTIFICATE-----  
<br>MIIDSjCCAjKgAwIBAgIQRK  
+wgNajJ7qJMDmGLvhAazANBgkqhkiG9w0BAQUFADA/  
<br>MSQwlgYDVQQKEExtEaWdpdGFsIFNpZ25hdHVyZSBUCnVzdCBDby4xZzAVBgNV  
BAMT<br>DkRTRVCBSb290IENBIFgzMB4XDTAwMDkzMDIxMTIxOVVoXDTIxMDkzMDE  
0MDExNVow<br>PzEkMCIGA1UEChMbRGlnaXRhbCBTaWduYXR1cmUgVHJ1c3QgQ  
28uMRcwFQYDVQQD<br>Ew5EU1QgUm9vdCBDQSBYmzCCASlWdQYJKoZIhvcNA  
QEBBQADggEPADCCAQoCggEB<br>AN  
+v6ZdQCINXtMxiZfaQguzH0yxrMMpb7NnDfcdAwRgUi+DoM3ZJKuM/  
IUmTrE4O<br>rz5Iy2Xu/  
NMhD2XSKtkyj4zl93ewEnu1lcCJo6m67XMuegwGMOifooUMM0RoOEQ<br>OLI5CjH9  
UL2AZd  
+3UWODyOKIYepLYYHsUmu5ouJLGiifSKOeDNoJjj4XLh7dIN9b<br>xiaqKqy69cK3FCx  
olkHRyxXtqqzTWMIn/5WgTe1QLyNau7Fqckh49ZLOMxt+/  
yUFw<br>7BZy1SbsOFU5Q9D8/

RhcQPGX69Wam40dutolucbY38EVAjqr2m7xPi71XAicPNaD<br>aeQQmxkqtilX4+U9m  
5/wAl0CAwEAAaNCMEAwDwYDVR0TAQH/BAUwAwEB/  
zAOBgNV<br>HQ8BAf8EBAMCAQYwHQYDVR0OBBYEFMSnsaR7LHH62+FLkHX/  
xBVghYkQMA0GCSqG<br>Slb3DQEBBQUAA4IBAQCjGiybFwBcqR7uKGY3Or  
+Dxz9LwwmglSBd49IZRNI+DT69<br>ikugdB/  
OEIKcdBodfpga3csTS7MgROSR6cz8faXbauX  
+5v3gTt23ADq1cEmv8uXr<br>AvHRAosZy5Q6XkjEGB5YGV8eAlrwDPGxrancWYaLbu  
mR9YbK  
+rlmM6pZW87ipxZz<br>R8srzJmwN0jP41ZL9c8PDHlyh8bwRLtTcm1D9SZlmlJnt1ir/  
md2cXjbDaJWFBM5<br>JDGFOqqCWjBH4d1QB7wCCZAA62RjYJJsWvljJEubSfZGL  
+T0yjWW06XyxV3bqxbYo<br>Ob8VZRzI9neWagqNdwwYkQsEjgfbKbYK7p2CNTUQ<b  
r>-----END CERTIFICATE-----<br></span></p><p class="debugging-content"><span  
class="debugging-title">Certificate Transparency: </span><span class="debugging-  
content"><br><br>SCT Cloudflare 'Nimbus2021' Log (Embedded in certificate,  
Verified)<br><br>SCT Google 'Xenon2021' log (Embedded in certificate, Verified)</  
span></p></div>

</div>

</div>

<div id="recurrent-error-message" class="hidden">

Warnings may be common while websites update their security. This should  
improve soon.

</div>

<div id="extended-reporting-opt-in" class="hidden">

<label>

<div class="checkboxes">

<input type="checkbox" id="opt-in-checkbox">

<span class="checkbox"></span>

</div>

<span id="opt-in-label">Help improve security on the web for everyone by  
sending <a href="#" id="whitepaper-link">URLs of some pages you visit, limited system  
information, and some page content</a> to Google. <a id="privacy-link"  
href="#">Privacy policy</a></span>

</label>

</div>

<div id="enhanced-protection-message" class="hidden">

<label>

<div class="icon"></div>

<div>To get Chrome's highest level of security, <a href="#" id="enhanced-  
protection-link">turn on enhanced protection</a></div>

</label>

</div>

</div>

<div class="nav-wrapper">

<button id="primary-button">Reload</button>

<button id="proceed-button" class="secondary-button small-link hidden"></button>

<button id="details-button" class="secondary-button small-link" aria-

```
expanded="true">Hide advanced</button>
</div>
<div id="details" class="">
  <p>accounts.shopify.com normally uses encryption to protect your information.
When Chrome tried to connect to accounts.shopify.com this time, the website sent back
unusual and incorrect credentials. This may happen when an attacker is trying to
pretend to be accounts.shopify.com, or a Wi-Fi sign-in screen has interrupted the
connection. Your information is still secure because Chrome stopped the connection
before any data was exchanged.</p>
  <p id="final-paragraph">You cannot visit accounts.shopify.com right now because
the website uses HSTS. Network errors and attacks are usually temporary, so this page
will probably work later.</p>
</div>
</div>
```

-----

```
<style>/* Copyright 2014 The Chromium Authors. All rights reserved.
* Use of this source code is governed by a BSD-style license that can be
* found in the LICENSE file. */

/* This file is dynamically processed by a C++ data source handler to fill in
* some per-platform/locale styles that dramatically alter the page. This is
* done to reduce flicker, as JS may not run before the page is rendered.
*
* There are two ways to include this stylesheet:
* 1. via its chrome://resources/ URL in HTML, i.e.:
*
*   <link rel="stylesheet" href="chrome://resources/css/text_defaults.css">
*
* 2. via the webui::AppendWebUICSSTextDefaults() method to directly append it
* to an HTML string.
* Otherwise its placeholders won't be expanded. */
```

```
body {
  font-family: system-ui, sans-serif;
  font-size: 75%;
}
```

```
button {
  font-family: system-ui, sans-serif;
}
</style>
```

-----

```
<script>// Copyright (c) 2012 The Chromium Authors. All rights reserved.  
// Use of this source code is governed by a BSD-style license that can be  
// found in the LICENSE file.
```

```
/**  
 * @fileoverview This file defines a singleton which provides access to all data  
 * that is available as soon as the page's resources are loaded (before DOM  
 * content has finished loading). This data includes both localized strings and  
 * any data that is important to have ready from a very early stage (e.g. things  
 * that must be displayed right away).  
 *  
 * Note that loadTimeData is not guaranteed to be consistent between page  
 * refreshes (https://crbug.com/740629) and should not contain values that might  
 * change if the page is re-opened later.  
 */
```

```
/** @type {!LoadTimeData} */  
// eslint-disable-next-line no-var  
var loadTimeData;
```

```
class LoadTimeData {  
  constructor() {  
    /** @type {?Object} */  
    this.data_ = null;  
  }
```

```
  /**  
   * Sets the backing object.  
   *  
   * Note that there is no getter for |data_| to discourage abuse of the form:  
   *  
   *   var value = loadTimeData.data()['key'];  
   *  
   * @param {Object} value The de-serialized page data.  
   */
```

```
  set data(value) {  
    expect(!this.data_, 'Re-setting data.');
```

```
    this.data_ = value;  
  }  
  
  /**  
   * @param {string} id An ID of a value that might exist.
```

```

* @return {boolean} True if lidl is a key in the dictionary.
*/
valueExists(id) {
  return id in this.data_;
}

/**
 * Fetches a value, expecting that it exists.
 * @param {string} id The key that identifies the desired value.
 * @return {*} The corresponding value.
 */
getValue(id) {
  expect(this.data_, 'No data. Did you remember to include strings.js?');
  const value = this.data_[id];
  expect(typeof value !== 'undefined', 'Could not find value for ' + id);
  return value;
}

/**
 * As above, but also makes sure that the value is a string.
 * @param {string} id The key that identifies the desired string.
 * @return {string} The corresponding string value.
 */
getString(id) {
  const value = this.getValue(id);
  expectIsType(id, value, 'string');
  return /** @type {string} */ (value);
}

/**
 * Returns a formatted localized string where $1 to $9 are replaced by the
 * second to the tenth argument.
 * @param {string} id The ID of the string we want.
 * @param {...(string|number)} var_args The extra values to include in the
 *   formatted output.
 * @return {string} The formatted string.
 */
getStringF(id, var_args) {
  const value = this.getString(id);
  if (!value) {
    return "";
  }

  const args = Array.prototype.slice.call(arguments);
  args[0] = value;
  return this.substituteString.apply(this, args);
}

```

```
}
```

```
/**
```

```
* Returns a formatted localized string where $1 to $9 are replaced by the
* second to the tenth argument. Any standalone $ signs must be escaped as
* $$.
```

`@param {string} label` The label to substitute through.

`This is not an resource ID.`

`@param {...(string|number)} var_args` The extra values to include in the formatted output.

`@return {string}` The formatted string.

```
*/
```

```
substituteString(label, var_args) {
  const varArgs = arguments;
  return label.replace(/\$(.|\n)/g, function(m) {
    expect(m.match(/\$[1-9]/), 'Unescaped $ found in localized string.');
```

`return m === '$$' ? '$' : varArgs[m[1]];`

```
  });
}
```

```
/**
```

```
* Returns a formatted string where $1 to $9 are replaced by the second to
* tenth argument, split apart into a list of pieces describing how the
* substitution was performed. Any standalone $ signs must be escaped as $$.
* @param {string} label A localized string to substitute through.
*   This is not an resource ID.
* @param {...(string|number)} var_args The extra values to include in the
*   formatted output.
* @return {!Array<!{value: string, arg: (null|string)}>} The formatted
*   string pieces.
*/
```

```
getSubstitutedStringPieces(label, var_args) {
  const varArgs = arguments;
  // Split the string by separately matching all occurrences of $1-9 and of
  // non $1-9 pieces.
  const pieces = (label.match(/(\$(1-9)|((\^$\)|\$(1-9)|$))+/g) ||
    []).map(function(p) {
    // Pieces that are not $1-9 should be returned after replacing $$
    // with $.
    if (!p.match(/\^$\$(1-9)$/)) {
      expect(
        (p.match(/\$/g) || []).length % 2 === 0,
        'Unescaped $ found in localized string.');
```

`return {value: p.replace(/\$\$/g, '$'), arg: null};`

```
    }
  }
}
```

```

        // Otherwise, return the substitution value.
        return {value: varArgs[p[1]], arg: p};
    });

    return pieces;
}

/**
 * As above, but also makes sure that the value is a boolean.
 * @param {string} id The key that identifies the desired boolean.
 * @return {boolean} The corresponding boolean value.
 */
getBoolean(id) {
    const value = this.getValue(id);
    expectIsType(id, value, 'boolean');
    return /** @type {boolean} */ (value);
}

/**
 * As above, but also makes sure that the value is an integer.
 * @param {string} id The key that identifies the desired number.
 * @return {number} The corresponding number value.
 */
getInteger(id) {
    const value = this.getValue(id);
    expectIsType(id, value, 'number');
    expect(value === Math.floor(value), 'Number isn\'t integer: ' + value);
    return /** @type {number} */ (value);
}

/**
 * Override values in loadTimeData with the values found in Ireplacementsl.
 * @param {Object} replacements The dictionary object of keys to replace.
 */
overrideValues(replacements) {
    expect(
        typeof replacements === 'object',
        'Replacements must be a dictionary object.');
    for (const key in replacements) {
        this.data_[key] = replacements[key];
    }
}

/**
 * Reset loadTimeData's data. Should only be used in tests.
 * @param {?Object} newData The data to restore to, when null restores to

```



```

    *   unset state.
    */
    resetForTesting(newData = null) {
        this.data_ = newData;
    }

    /**
     * @return {boolean} Whether loadTimeData.data has been set.
     */
    isInitialized() {
        return this.data_ !== null;
    }
}

/**
 * Checks condition, throws error message if expectation fails.
 * @param {*} condition The condition to check for truthiness.
 * @param {string} message The message to display if the check fails.
 */
function expect(condition, message) {
    if (!condition) {
        throw new Error(
            'Unexpected condition on ' + document.location.href + ': ' + message);
    }
}

/**
 * Checks that the given value has the given type.
 * @param {string} id The id of the value (only used for error message).
 * @param {*} value The value to check the type on.
 * @param {string} type The type we expect lvalue to be.
 */
function expectIsType(id, value, type) {
    expect(
        typeof value === type, '[' + value + ']' (' + id + ') is not a ' + type);
}

expect(!loadTimeData, 'should only include this file once');
loadTimeData = new LoadTimeData;

// Expose lloadTimeData directly on lwindowl, since within a JS module the
// scope is local and not all files have been updated to import the exported
// lloadTimeData explicitly.
window.loadTimeData = loadTimeData;

console.warn('crbug/1173575, non-JS module files deprecated.');
```

-----

```
<script>loadTimeData.data = {"bad_clock":false,"closeDetails":"Hide advanced","ct":"\n\nSCT Cloudflare 'Nimbus2021' Log (Embedded in certificate, Verified)\n\nSCT Google 'Xenon2021' log (Embedded in certificate, Verified)","currentDate":"Oct 5, 2021","displayEnhancedProtectionMessage":false,"displaycheckbox":false,"enhancedProtectionMessage":"To get Chrome's highest level of security, \u003Ca href=\"\#\" id=\"enhanced-protection-link\">turn on enhanced protection\u003C/a>","errorCode":"net::ERR_CERT_DATE_INVALID","errorType":1,"expirationDate":"Nov 13, 2021","explanationParagraph":"accounts.shopify.com normally uses encryption to protect your information. When Chrome tried to connect to accounts.shopify.com this time, the website sent back unusual and incorrect credentials. This may happen when an attacker is trying to pretend to be accounts.shopify.com, or a Wi-Fi sign-in screen has interrupted the connection. Your information is still secure because Chrome stopped the connection before any data was exchanged.","finalParagraph":"You cannot visit accounts.shopify.com right now because the website uses HSTS. Network errors and attacks are usually temporary, so this page will probably work later.","fontfamily":"system-ui, sans-serif","fontsize":"75%","heading":"Your connection is not private","hide_primary_button":false,"issuer":"R3","language":"en","openDetails":"Advanced","optInLink":"Help improve security on the web for everyone by sending \u003Ca href=\"\#\" id=\"whitepaper-link\">URLs of some pages you visit, limited system information, and some page content\u003C/a> to Google. \u003Ca id=\"privacy-link\" href=\"\#\">Privacy policy\u003C/a>","overridable":false,"pem":"-----BEGIN CERTIFICATE-----\nMIIF2DCCBMCGAwIBAgISA3IP+wuRtzmomjRFbuWpyrsCMA0GCSqGSIb3DQEBCwUA\n\nMDIx CzAJBgNVBAYTAIVTMRYwFAYDVQQKEw1MZXQncyBFbmNyeXB0MQswCQYDVQQD\n\nEwJSMzAeFw0yMTA4MTUxNjE3MDNaFw0yMTE3MTUxNjE3MDFaMB8xHTAbBgNVBAMT\n\nFGFjY291bnRzLnNob3BpZnkuY29tMIIlBlJANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB\n\nCgKCAQEAmNO6kENQqqUM0WgozXecZGcPzrlyEbrU/N7mdStXBNjXN9WAEbLSTGrw\ngkyShj1dbDXVJudJ7YdQXJr/KJbm87u8a1HEZI4Dmj1FxBIAd+Gs1RsFdQd0nEw\n\n5Zgi0Sojj6d87SEqpS2i0Vz6V+hTrbuy+XXP5+J8piNiRp2lkrZQkWKaifZaLyvH\n\nnw2S2OFtEJxH3SSoKfsA/ST4ZPRDFIppsXgmpFf7sU5u1E3XckNJxl220Q29/vtIs\n\nErLP9noTy1wdKzXJYbu4IT6nVOSS+TGLQBcBCfNQyaG19jMrL299G4QyNy2h6uWX\n\nMwPm/Opq3qk3Bnl5uZgyV+mx7EMe1QIDAQABo4IC+TCCAvUwDgYDVR0PAQH/BAQD\n\nAgWgMB0GA1UdJQQWMBQGCCsGAQUFBwMBBggrBgEFBQcDAjAMBgNVHRMBAf8EAjAA\n\nnMB0GA1UdDgQWBBR/TRz5xF8Ma/wcElldjg8fyl3L3zAfBgNVHSMEGDAWgBQULrMX\n\nnt1hWy65QCUDmH6+dixTCxjBVBggrBgEFBQcBAQRJMEcwIQYIKwYBBQUHMAGGF
```

Wh0\ndHA6Ly9yMy5vLmxlbmNyLm9yZzAiBggrBgEFBQcwAoYWaHR0cDovL3lzLmkub  
GVu\`nY3lub3JnLzCBYQYDVR0RBIHBMIG  
+gh5hY2NvdW50cy1kZW1vLnNob3BpZnlibG91\`nZC5jb22CFGFjY291bnRzLnNob3BpZ  
nkuY29tgh9jdXN0b21lcnMtZGVtby5zaG9w  
\`naWZ5Y2xvdWQuY29tgiZpZGVudGl0eS1tYWluLWNsdXN0ZXluc2hvcGlmeWNSb3Vk  
\`nLmNvbYliaWRlbnRpdHktbWVyY2hhbnQuc2hvcGlmeWNSb3VkLmNvbYlZaWRlbnRp  
\`ndHkuc2hvcGlmeWNSb3VkLmNvbTBMBgNVHSAERTBDMAgGBmeBDAECATA3Bgsr  
BgEE  
\`nAYLfEwEBATAoMCYGCCsGAQUFBwIBFhpodHRwOi8vY3BzLmxldHNIbmNyeXB0L  
m9y\`nZzCCAQMGCisGAQQB1nkCBAIEgfQEgfEA7wB1AESUZS6w7s6vxEAH2Kj  
+KMDa5oK+\`n2MsxtT/TM5a1toGoAAABe0rRIFkAAQAQDAEYwRAIgcU9G4Kf  
+YE3BwY4IHwrPfCyx\`nHBfmiWZVYY3syP6PMtQCICbrsrVPy0eYoa  
+way7bJOxuRAbwT4Sfd4Nd14ITkrE\`nAHYAfT7y+l//  
iFVoJMLAyp5SiXkrxQ54CX8uapdomX4i8NcAAAF7StEgegAABAMA\`nRzBFAiBQFP/  
Jq9s0RX2lddzPRWorKmK2vHb0gBjdeR4DafmJfglhAOGXW8pNxx+q  
\`nAgkEhILco7khwJUOAcDqcKCI9HS9FOJ8MA0GCSqGSIb3DQEBCwUAA4IBAQBcKa  
nD\`nDWicj0QRpqqbMyHi1uwU0vJoOuCXAB5TMevD1OGhl2HGfV8blgxWlfoULxGoEd/  
k\`nSA0t9X94AFhu+FhBQnrssUH/D+xQFKSbaZcQIB5j1nEFNdtpokxW5JyhjS5MXiYO  
\`ndWfgQcEhx3HlnYVT+Z1CpNslgCSOaoOawx+b1y1/Kj5ZIEld2rZBIOxh+/ifPP9h\`nZ8Y/  
kug+O2Vu7xfPWn4RtVeG39FpK3FcbOxkTA09K1CHtPQf+KZ/  
Ou5BcLn8BJi5\`nMVXdjQTeKuzilyV7QM+dAxHn3/  
reMXmlGengm7nj9K2P3I7Z70dX1sEwUXQR8Wr\`n/+lfVfwGqqjAuskK\`n-----END  
CERTIFICATE-----\`n-----BEGIN CERTIFICATE-----  
\`nMIIFfjCCAv6gAwIBAgIRAJErCERPDBinU/bWLiWnX1owDQYJKoZIhvcNAQELBQAww  
\`nTzELMAkGA1UEBhMCVVMxKTAnBgNVBAoTIEludGVybmV0IFNIY3VyaXR5IFJlc2Vh  
\`ncmNoIEdyb3VwMRUwEwYDVQQDEwxJU1JHIFJvb3QgWDEwHhcNMjAwOTA0MDAwMDAw  
wMDAw  
\`nWhcNMjUwOTE1MTYwMDAwWjAyaMQswCQYDVQQGEwJVUzEWMBQGA1UEChM  
NTGV0J3Mg  
\`nRW5jcmlwdDELMAkGA1UEAxMCUjMwggeiEiMA0GCSqGSIb3DQEBAQUAA4IBDwAw  
ggEK\`nAoIBAQC7AhUozPaglNMPEuyNVZLD+ILxmaZ6QoinXSaqtsu5xUyxr45r  
+XXlo9cP  
\`nR5QUVTVXjJ6oojkZ9YI8QqlObvU7wy7bjcCwXPNZOOftz2nwWgsbvsCUJCWH+jdx  
\`nsxPnHKzhm+/b5DtFukWWqcFTzjTIUu61ru2P3mBw4qVUq7ZtDpelQDRrK9O8Zutm  
\`nNHZ6a4uPVymZ+DAXXbpyb/uBxa3Shlg9F8fnCbvxK/eG3MHacV3URuPMrSXBilXg  
\`nZ3Vms/EY96Jc5IP/Ooi2R6X/ExjqmAl3P51T+c8B5fWmcBcUr2Ok/5mzk53cU6cG\`n/  
kiFHaFpriV1uxPMUgP17VGhi9sVAgMBAAGjggEIMIIBBDAOBgNVHQ8BAf8EBAMC  
\`nAYYwHQYDVR0IBBYwFAYIKwYBBQUHAwIGCCsGAQUFBwMBMBIGA1UdEwEB/  
wQIMAYB  
\`nAf8CAQAwHQYDVR0OBBYEFBQusxe3WFbLrlAJQOYfr52LFMLGMB8GA1UdIwQY  
MBaA  
\`nFHm0WeZ7tuXkAXOACijlGij26ZtuMDIGCCsGAQUFBwEBBCYwJDAiBggrBgEFBQc  
w  
\`nAoYWaHR0cDovL3gxLmkubGVuY3lub3JnLzAnBgNVHR8EIDAeMBygGqAYhhZodH  
Rw  
\`nOi8veDEuYy5sZW5jci5vcmcvMCIGA1UdIAQbMBkwCAYGZ4EMAQIBMA0GCysGAQ

QB

\ngt8TAQEBMA0GCSqGSib3DQEBCwUAA4ICAQCfYk5HPqP3hUSFvNVneLKYy611T  
R6W\nPTNlclQtgaDqw+34IL9fzLdwALduO/ZelN7klJ+m74uyA+eitRY8kc607TkC53wl  
\nikfmZW4/RvTZ8M6UK+5UzhK8jCdLuMGYL6KvzXGRSgi3yLgjewQtCPkIVz6D2QQz  
\nCkcheAmCJ8MqyJu5zlzyZMjAvnnAT45tRAxekrsu94sQ4egdRCnbWSDtY7kh+Blm  
\nIJNXoB1IBMEKlq4QDUOXoRgffuDghje1WrG9ML+Hbisq/  
yFOGwXD9RiX8F6sw6W4\nnavAuvDszue5L3sz85K+EC4Y/wFVDNvZo4TYXao6Z0f  
+lQKc0t8DQYzk1OXVu8rp2\nnyJMC6alLbBfODALZvYH7n7do1AZIs4l9d1P4jnkDrQoxB3  
UqQ9hVI3LEKQ73xF1O\nnyK5GhDDX8oVfGKF5u  
+declsH4YaTw7mP3GFxJSqv3+0IUFJoi5Lc5da149p90lds  
\nhCExroL1+7mrylkXPeFM5TgO9r0rvZaBFOvV2z0gp35Z0+L4WPIbuEjN/lxPFin+  
\nHIUjr8gRsl3qfJOQFy/9rKIJR0Y/8Omwt/8oTWgy1mdeHmnmjk7j1nYsvC9JSQ6Zv  
\nMIdITTKB3zhThV1+XWYp6rjd5JW1zbVWEkLNxE7GJThEUG3szgBVGP7pSWTUTsq  
X\nnnLRbwHOoq7hHwg==\n-----END CERTIFICATE-----\n-----BEGIN CERTIFICATE-----  
\nMIIFYDCCBEigAwIBAgIQQAF3ITfU6UK47naqPGQKtzANBgkqhkiG9w0BAQsFADA/  
\nMSQWlgYDVQQKEExtEaWdpdGFsIFNpZ25hdHVyZSBUCnVzdCBDby4xFzAVBgNVBA  
MT  
\nDkRtVCBSb290IENBIFgzMB4XDTIxMDEyMDE5MTQwM1oXDTI0MDkzMDE4MTQw  
M1ow  
\nTzELMAkGA1UEBhMCVVMxKTAnBgNVBAoTIEludGVybmV0IFNIY3VyaXR5IFJlc2Vh  
\ncmNoIEdyb3VwMRUwEwYDVQQDEwXJU1JHIFJvb3QgWDEwggliMA0GCSqGSib3D  
QEB\nAQUAA4ICDwAwggIKAoICAQCt6CRz9BQ385ueK1coHle  
+3LffOJCMbjzmV6B493XC\nnov71am72AE8o295ohmxEk7axY/0UEmu/  
H9LqMZshftEzPLpI9d1537O4/xLxIZpL\nnwYqGcWIKZmZsj348cL  
+tKSIG8+TA5oCu4kuPt5l+IAOf00eXfJIII1PoOK5PCm+D  
\nLtFJV4yAdLbaL9A4jXsDcCEbdfIwPPqPrt3aY6vrFk/CjhFLfs8L6P+1dy70sntK  
\n4EwSJQxwjQMpoOFTJOwT2e4ZvxCzSow/  
iaNhUd6shweU9GNx7C7ib1uYgeGJXDR5\nnbHbvO5BieebbpJovJsXQEOEO3tkQjhb7t/  
eo98flAgeYjzYllefI5YNNnWe+w5y\nnsR2bvAP5SQXYgd0FtCrWQemsAXaVCg/  
Y39W9Eh81LygXbNKYwagJZHduRze6zqxZ\nnXmidf3LWicUGQSk  
+WT7dJvUkyRGnWqNMQB9GoZm1pzpRboY7nn1ypxIFeFntPIF4\nnFQsDj43QLwWyPn  
tKHETzBRL8xurgUBN8Q5N0s8p0544fAQjQMNRbcTa0B7rBMDBC  
\nSLeCO5imfWCKoqMpgsy6vYMEG6KDA0Gh1gXxG8K28Kh8hjtGqEgqiNx2mna/H2ql  
\nPRmP6zjzZN7IKw0KKP/32+IVQtQi0Cdd4Xn+GOdwiK1O5tmLOsbdJ1Fu/7xk9TND  
\nTwIDAQABo4IBRjCCAUIwDwYDVR0TAQH/BAUwAwEB/  
zAOBgNVHQ8BAf8EBAMCAQYw  
\nSwYIKwYBBQUHAQEEPzA9MDsGCCsGAQUFBzACHi9odHRwOi8vYXBwcy5pZGVu  
dHJ1\nc3QuY29tL3Jvb3RzL2RzdHJvb3RjYXgzLnA3YzAfBgNVHSMEGDAWgBTEp7G  
keyxx\n+tvhS5B1/8QVYIWJEDBUBgNVHSAETTLMAgGBmeBDAECATA/  
BgSrBgEEAYLfEwEB  
\nATAwMC4GCCsGAQUFBwIBFiJodHRwOi8vY3BzLnJvb3QteDEubGV0c2VuY3J5cHQ  
u  
\nb3JnMDwGA1UdHwQ1MDMwMaAvoC2GK2h0dHA6Ly9jcmwuaWRlbnRydXN0LmNv  
bS9E  
\nU1RST09UQ0FYM0NSTC5jcmwwHQYDVR0OBBYEFHm0WeZ7tuXkAXOAClJlGlj26  
Ztu\nMA0GCSqGSib3DQEBCwUAA4IBAQAkCwBslm7/DILQrt2M51oGrS+o44+/

yQoDFVDC  
\\n5WxCu2+b9LRPwkSICHXM6webFGJueN7sJ7o5XPWioW5WIHAQU7G75K/  
QosMrAdSW\\n9MUgNTP52GE24HGNTLi1qoJFicDyqSMo59ahy2cl2qBDLKobkx/  
J3vWraV0T9VuG  
\\nWCLKTVXkcGdtwlfFRjIBz4pYg1htmf5X6DYO8A4jqv2II9DjXA6USbW1FzXSLr9O  
\\nhe8Y4IWS6wY7bCkjCWDcRQJMEhg76fsO3txE+FiYruq9RUWhiF1myv4Q6W  
+CyBFC\\nDfvp7OOGAN6dEOM4+qR9sdjoSYKEBpsr6GtPAQw4dy753ec5\\n-----END  
CERTIFICATE-----\\n-----BEGIN CERTIFICATE-----\\nMIIDSjCCAjKgAwIBAgIQRK  
+wgNajJ7qJMDmGLvhAazANBgbkqhkiG9w0BAQUFADA/  
\\nMSQwlgYDVQQKEExtEaWdpdGFsIFNpZ25hdHVyZSBUcnVzdCBDby4xFzAVBgNVBA  
MT  
\\nDkRtVCBSb290IENBIFgzMB4XDTAwMDkzMDIxMTIxOVoXDTIxMDkzMDE0MDExN  
Vow  
\\nPzEkMCIGA1UEChMbRGlnaXRhbCBTaWduYXR1cmUgVHJ1c3QgQ28uMRcwFQYD  
VQQD  
\\nEw5EU1QgUm9vdCBDQSBYMzCCASlWdQYJKoZIhvcNAQEBBQADggEPADCCAQ  
oCggEB\\nAN+v6ZdQCINXtMxiZfaQguzH0yxrMMpb7NnDfcdAwRgUi+DoM3ZJKuM/  
IUmrE4O\\nrz5Iy2Xu/  
NMhD2XSKtkyj4zl93ewEnu1lcCJo6m67XMuegwGMOifooUMM0RoOEQ  
\\nOLI5CjH9UL2AZd+3UWODyOKIYepLYYHsUmu5ouJLGiiFSKOeDNoJjj4XLh7dIN9b  
\\nxiqKqy69cK3FCxolkHRyxXtqqzTWMI\\n/5WgTe1QLyNau7Fqckh49ZLOMxt+/yUFw  
\\n7BZy1SbsOFU5Q9D8/RhcQPGX69Wam40dutolucbY38EVAjqr2m7xPi71XAicPNaD  
\\naeQQmxkqtilX4+U9m5/wAl0CAwEAANCMEEAwDwYDVR0TAQH/BAUwAwEB/  
zAOBgNV\\nHQ8BAf8EBAMCAQYwHQYDVR0OBBYEFMSnsaR7LHH62+FLkHX/  
xBVghYkQMA0GCsG\\nSIlb3DQEBBQUAA4IBAQCjGiybFwBcqR7uKGY3Or  
+Dxz9LwwmgISBd49IZRNI+DT69\\nikugdB/  
OEIKcdBodfpga3csTS7MgROSR6cz8faXbauX+5v3gTt23ADq1cEmv8uXr  
\\nAvHRAosZy5Q6XkjEGB5YGV8eAlrwDPGxrancWYaLbumR9YbK+rlmM6pZW87ipxZz  
\\nR8srzJmwN0jP41ZL9c8PDHlyh8bwRLtTcm1D9SZlmlJnt1ir/  
md2cXjbDaJWFBM5\\nJDGFoqgCWjBH4d1QB7wCCZAA62RjYJsWvljJEubSfZGL  
+T0yjWW06XyxV3bqxbYo\\nOb8VZRzI9neWagqNdwwYkQsEjgfbKbYK7p2CNTUQ\\n-----  
END CERTIFICATE-----\\n","primaryButtonText":"Reload","primaryParagraph":"Attackers  
might be trying to steal your information from \\u003Cstrong>accounts.shopify.com  
\\u003C/strong> (for example, passwords, messages, or credit cards). \\u003Ca href=\\\"#  
\\\" id=\\\"learn-more-link\\\">Learn more\\u003C/a>","recurrentErrorParagraph":"Warnings  
may be common while websites update their security. This should improve  
soon.","show\_recurrent\_error\_paragraph":false,"subject":"accounts.shopify.com","tabTitl  
e":"Privacy error","textdirection":"ltr","type":"SSL"};</script>

-----  
  
// Copyright (c) 2013 The Chromium Authors. All rights reserved.  
// Use of this source code is governed by a BSD-style license that can be  
// found in the LICENSE file.

```

/**
 * @fileoverview Assertion support.
 */

/**
 * Verify lconditionl is truthy and return lconditionl if so.
 * @template T
 * @param {T} condition A condition to check for truthiness. Note that this
 *   may be used to test whether a value is defined or not, and we don't want
 *   to force a cast to Boolean.
 * @param {string=} opt_message A message to show on failure.
 * @return {T} A non-null lconditionl.
 * @closurePrimitive {asserts.truthy}
 * @suppress {reportUnknownTypes} because T is not sufficiently constrained.
 */
/* #export */ function assert(condition, opt_message) {
  if (!condition) {
    let message = 'Assertion failed';
    if (opt_message) {
      message = message + ': ' + opt_message;
    }
    const error = new Error(message);
    const global = function() {
      const thisOrSelf = this || self;
      /** @type {boolean} */
      thisOrSelf.traceAssertionsForTesting;
      return thisOrSelf;
    }();
    if (global.traceAssertionsForTesting) {
      console.warn(error.stack);
    }
    throw error;
  }
  return condition;
}

/**
 * Call this from places in the code that should never be reached.
 *
 * For example, handling all the values of enum with a switch() like this:
 *
 * function getValueFromEnum(enum) {
 *   switch (enum) {
 *     case ENUM_FIRST_OF_TWO:
 *       return first
 *     case ENUM_LAST_OF_TWO:

```

```

*     return last;
*   }
*   assertNotReached();
*   return document;
* }
*
* This code should only be hit in the case of serious programmer error or
* unexpected input.
*
* @param {string=} opt_message A message to show when this is hit.
* @closurePrimitive {asserts.fail}
*/
/* #export */ function assertNotReached(opt_message) {
  assert(false, opt_message || 'Unreachable code hit');
}

/**
* @param {*} value The value to check.
* @param {function(new: T, ...)} type A user-defined constructor.
* @param {string=} opt_message A message to show when this is hit.
* @return {T}
* @template T
*/
/* #export */ function assertInstanceOf(value, type, opt_message) {
  // We don't use assert immediately here so that we avoid constructing an error
  // message if we don't have to.
  if (!(value instanceof type)) {
    assertNotReached(
      opt_message ||
      'Value ' + value + ' is not a[n] ' + (type.name || typeof type));
  }
  return value;
}

/* #ignore */ console.warn('crbug/1173575, non-JS module files deprecated.');
```

-----

VM2261:2134 crbug/1173575, non-JS module files deprecated.  
 (anonymous) @ VM2261:2134

debugger:///VM2261

```
// Copyright (c) 2012 The Chromium Authors. All rights reserved.  
// Use of this source code is governed by a BSD-style license that can be  
// found in the LICENSE file.
```

```
/**  
 * @fileoverview This file defines a singleton which provides access to all data  
 * that is available as soon as the page's resources are loaded (before DOM  
 * content has finished loading). This data includes both localized strings and  
 * any data that is important to have ready from a very early stage (e.g. things  
 * that must be displayed right away).  
 *  
 * Note that loadTimeData is not guaranteed to be consistent between page  
 * refreshes (https://crbug.com/740629) and should not contain values that might  
 * change if the page is re-opened later.  
 */
```

```
/** @type {!LoadTimeData} */  
// eslint-disable-next-line no-var  
var loadTimeData;
```

```
class LoadTimeData {  
  constructor() {  
    /** @type {?Object} */  
    this.data_ = null;  
  }
```

```
  /**  
   * Sets the backing object.  
   *  
   * Note that there is no getter for |data_| to discourage abuse of the form:  
   *  
   *   var value = loadTimeData.data()['key'];  
   *  
   * @param {Object} value The de-serialized page data.  
   */
```

```
  set data(value) {  
    expect(!this.data_, 'Re-setting data.');
```

```
    this.data_ = value;  
  }
```

```
  /**  
   * @param {string} id An ID of a value that might exist.  
   * @return {boolean} True if |id| is a key in the dictionary.  
   */
```

```
  valueExists(id) {  
    return id in this.data_;
```



```
}
```

```
/**
```

```
 * Fetches a value, expecting that it exists.  
 * @param {string} id The key that identifies the desired value.  
 * @return {*} The corresponding value.  
 */
```

```
getValue(id) {  
  expect(this.data_, 'No data. Did you remember to include strings.js?');  
  const value = this.data_[id];  
  expect(typeof value !== 'undefined', 'Could not find value for ' + id);  
  return value;  
}
```

```
/**
```

```
 * As above, but also makes sure that the value is a string.  
 * @param {string} id The key that identifies the desired string.  
 * @return {string} The corresponding string value.  
 */
```

```
getString(id) {  
  const value = this.getValue(id);  
  expectIsType(id, value, 'string');  
  return /** @type {string} */ (value);  
}
```

```
/**
```

```
 * Returns a formatted localized string where $1 to $9 are replaced by the  
 * second to the tenth argument.  
 * @param {string} id The ID of the string we want.  
 * @param {...(string|number)} var_args The extra values to include in the  
 *   formatted output.  
 * @return {string} The formatted string.  
 */
```

```
getStringF(id, var_args) {  
  const value = this.getString(id);  
  if (!value) {  
    return '';  
  }  
  
  const args = Array.prototype.slice.call(arguments);  
  args[0] = value;  
  return this.substituteString.apply(this, args);  
}
```

```
/**
```

```
 * Returns a formatted localized string where $1 to $9 are replaced by the
```

```

* second to the tenth argument. Any standalone $ signs must be escaped as
* $$.
```

```

* @param {string} label The label to substitute through.
*   This is not an resource ID.
* @param {...(string|number)} var_args The extra values to include in the
*   formatted output.
* @return {string} The formatted string.
*/
substituteString(label, var_args) {
  const varArgs = arguments;
  return label.replace(/\$(.|\n)/g, function(m) {
    expect(m.match(/\$[1-9]/), 'Unescaped $ found in localized string.');
```

```

    return m === '$$' ? '$' : varArgs[m[1]];
  });
}

/**
* Returns a formatted string where $1 to $9 are replaced by the second to
* tenth argument, split apart into a list of pieces describing how the
* substitution was performed. Any standalone $ signs must be escaped as $$.
* @param {string} label A localized string to substitute through.
*   This is not an resource ID.
* @param {...(string|number)} var_args The extra values to include in the
*   formatted output.
* @return {!Array<!{value: string, arg: (null|string)}>}} The formatted
*   string pieces.
*/
getSubstitutedStringPieces(label, var_args) {
  const varArgs = arguments;
  // Split the string by separately matching all occurrences of $1-9 and of
  // non $1-9 pieces.
  const pieces = (label.match(/(\$(1-9))|((\^$\)|\$(1-9)|$))+/g) ||
    [].map(function(p) {
      // Pieces that are not $1-9 should be returned after replacing $$
      // with $.
      if (!p.match(/\^$\$(1-9)$/)) {
        expect(
          (p.match(/\$/g) || []).length % 2 === 0,
          'Unescaped $ found in localized string.');
```

```

        return {value: p.replace(/\$\$/g, '$'), arg: null};
      }

      // Otherwise, return the substitution value.
      return {value: varArgs[p[1]], arg: p};
    });
}

```

```

    return pieces;
}

/**
 * As above, but also makes sure that the value is a boolean.
 * @param {string} id The key that identifies the desired boolean.
 * @return {boolean} The corresponding boolean value.
 */
getBoolean(id) {
    const value = this.getValue(id);
    expectIsType(id, value, 'boolean');
    return /** @type {boolean} */ (value);
}

/**
 * As above, but also makes sure that the value is an integer.
 * @param {string} id The key that identifies the desired number.
 * @return {number} The corresponding number value.
 */
getInteger(id) {
    const value = this.getValue(id);
    expectIsType(id, value, 'number');
    expect(value === Math.floor(value), 'Number isn\'t integer: ' + value);
    return /** @type {number} */ (value);
}

/**
 * Override values in loadTimeData with the values found in Ireplacementsl.
 * @param {Object} replacements The dictionary object of keys to replace.
 */
overrideValues(replacements) {
    expect(
        typeof replacements === 'object',
        'Replacements must be a dictionary object.');"
    for (const key in replacements) {
        this.data_[key] = replacements[key];
    }
}

/**
 * Reset loadTimeData's data. Should only be used in tests.
 * @param {?Object} newData The data to restore to, when null restores to
 *     unset state.
 */
resetForTesting(newData = null) {
    this.data_ = newData;
}

```

```

}

/**
 * @return {boolean} Whether loadTimeData.data has been set.
 */
isInitialized() {
  return this.data_ !== null;
}
}

/**
 * Checks condition, throws error message if expectation fails.
 * @param {*} condition The condition to check for truthiness.
 * @param {string} message The message to display if the check fails.
 */
function expect(condition, message) {
  if (!condition) {
    throw new Error(
      'Unexpected condition on ' + document.location.href + ': ' + message);
  }
}

/**
 * Checks that the given value has the given type.
 * @param {string} id The id of the value (only used for error message).
 * @param {*} value The value to check the type on.
 * @param {string} type The type we expect lvalue to be.
 */
function expectIsType(id, value, type) {
  expect(
    typeof value === type, '[' + value + '] (' + id + ') is not a ' + type);
}

expect(!loadTimeData, 'should only include this file once');
loadTimeData = new LoadTimeData;

// Expose lloadTimeData directly on lwindowl, since within a JS module the
// scope is local and not all files have been updated to import the exported
// lloadTimeData explicitly.
window.loadTimeData = loadTimeData;

console.warn('crbug/1173575, non-JS module files deprecated.');
```