附件

注册验船师资格考试考务管理系统

报名操作指引

### 流程图

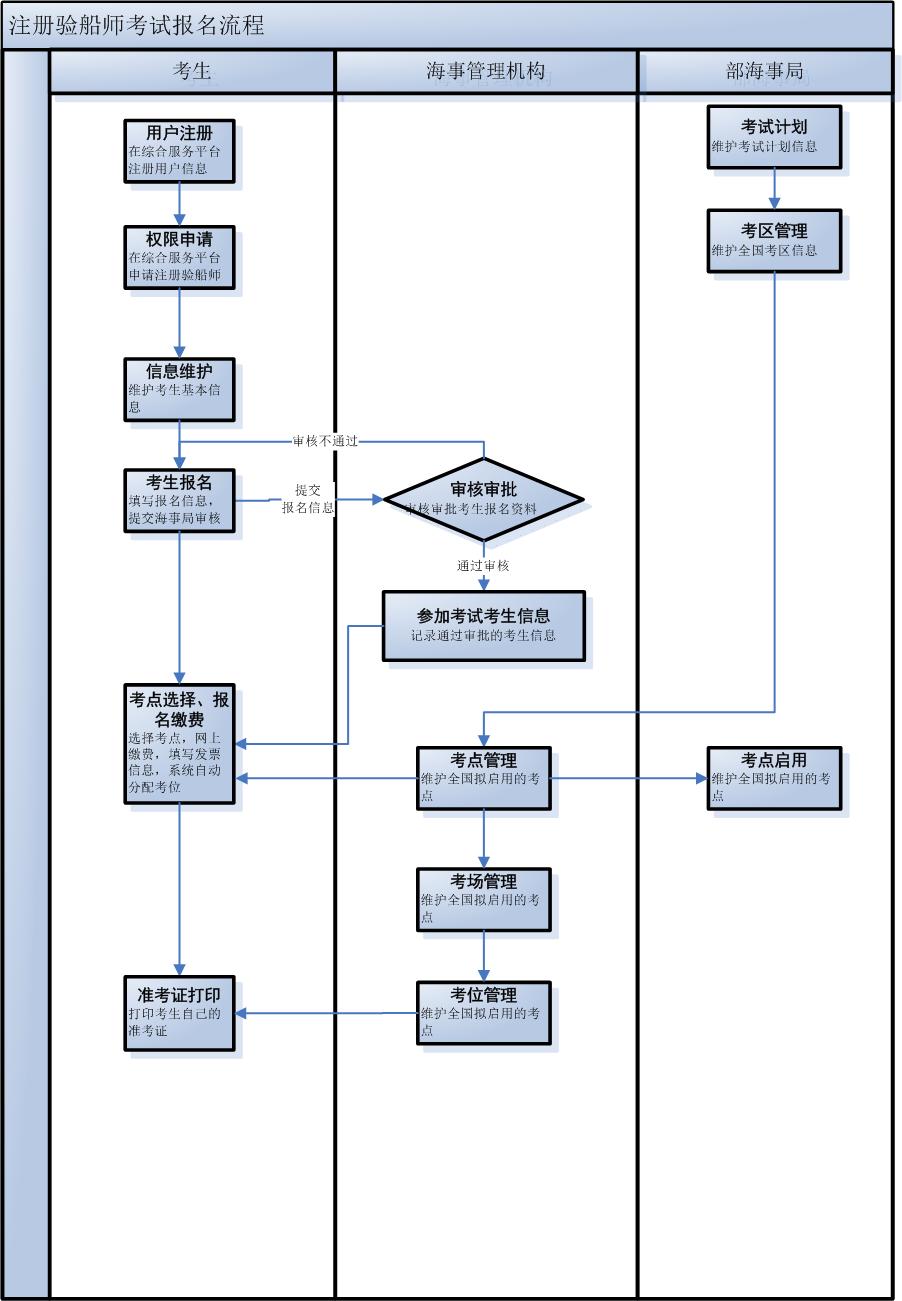


图1

### 注册

1. 网址

海事局官网（互联网）：[http://www.msa.gov.cn/](http://cspur.msa.gov.cn/UserReg/orgUserReg.do)



图2

“点击进入综合服务平台”超链，进入到中国海事综合服务平台登录页面，点击“用户注册”，选择自然人用户注册，填写相关信息。



图3

注册成功之后，请设置您的用户密码，用于您以后登录时使用。



图4

点击“下一步”操作，显示您提交注册的详细信息，包括：姓名、账号、手机号、邮箱、密码；点击“提交注册”提交注册信息，点击“上一步”返回到填写密码阶段。

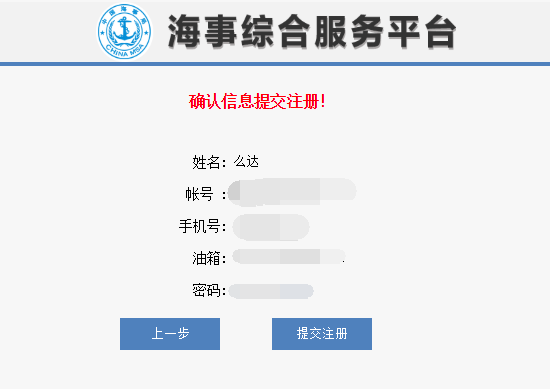


图5

### 权限申请

1、注册用户成功之后，在海事局官网进行登录，进入海事综合服务平台，申请权限功能，会延时显示，请您耐心等待3分钟左右。



图6

2、点击“权限申请”超链，选择“船舶检验系统”权限。



图7

1. 点击“船舶检验系统”超链后，显示“申请船舶检验权限”超链。



图8

4、申请“船舶检验系统”之后，会出现“注册验船师考试报名”图标，点击之后，就可以完成外网自然人申请验船师考试的权限申请步骤，进行报名。



图9

点击“船舶检验管理信息系统”，就可以进入我们的“注册验船师考试网上报名系统”了。



图10

### 信息维护

报名前需先维护个人基本信息，系统中会自动显示考生的“姓名”、“身份证号”、“电子邮箱”、“手机号码”信息，填写要求必填的考生信息点击保存按钮，保存数据信息。



图11

### 考生报名

1、报名第一步，填写报名信息，报名状态第一次考试默认为“新考”。点击“《专业技术人员资格考试违纪违规行为处理规定》”、“《注册验船师资格考试应考人守则》”超链，可以查看详细信息。如果早于报名开始时间或者晚于报名结束时间都无法进行报名。



图12

2、报名第二步，该步骤默认显示报名的期数，考生需要选择报考的考试等级。考试等级分为：A、B、C、D四个等级，不同等级需要满足不同的条件。这里也可以返回上一步，对数据进行修改。



图13

1. 报名第三步，上传验船师考试所需的图片或者扫描件等资料。可以勾选“申请免考”，申请免考资格。这里也可以返回上一步，对的数据进行修改。***注意：所有报名人员本次都需提交证件照及相关证明文件。证件照为1寸，底色为白色，jpg格式，像素为295\*413，大小在10K至30K以内。附件大小在50KB至512KB之间，如果有多张图片的请合成一张上传。***







图14

1. 确定并预览，进入预览画面，点击“超链”可以查看您所上传的图片信息。点击“编辑”图标，可以回到对应的步骤修改报名信息。选择资料审核提交的海事局信息，提交给对应的海事局进行审核。





图15

5、查看审核结果信息，考生可以查看自己的报名资料审核情况。如果不合格会提示需要修改信息，由考生重新编写相关信息，重新提交即可。



图16

### 选择考点、缴费

1、批准通过的考生，进入报名缴费阶段。该阶段需要部局用户启用考点，同时缴费时间开放之后才能进行缴费，如果早于缴费开始时间或者晚于缴费结束时间都无法进行缴费。



图17

2、缴费按钮，可以根据本期设置开放的考区进行选择，可以选择所在的省市以及考点的类型，只有D级的考生才可以选择纸质类型，系统将自动给考生分配考位信息。



图18

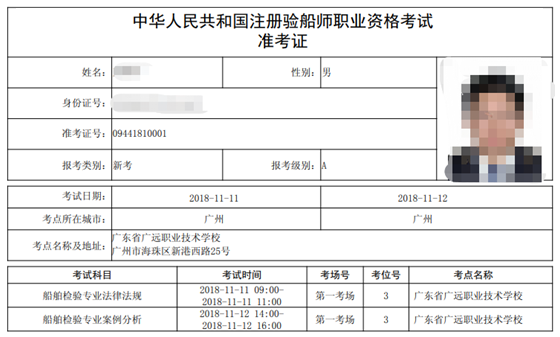
1. 点击“确认”按钮，进入缴费页面，选择支付方式，进行支付。**（具体操作详见附件一）**



图19

### 打印准考证

考生打印准考证相关信息，如果早于准考证打印开始时间或者晚于准考证打印结束时间都无法进行打印。



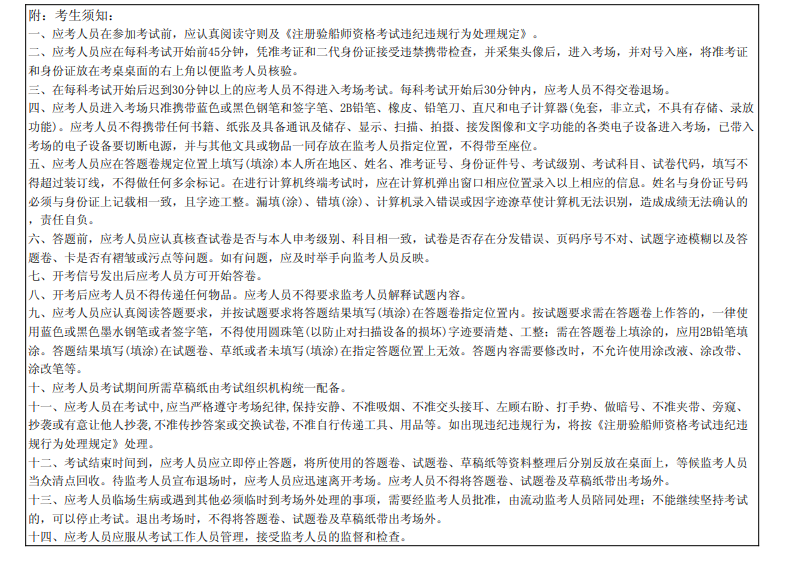


图20

**附件一：**

### **选择考点、缴费**

1、批准通过的考生，进入报名缴费阶段。该阶段需要部局用户启用考点，同时缴费时间开放之后才能进行缴费，如果早于缴费开始时间或者晚于缴费结束时间都无法进行缴费。***注：缴费功能请使用谷歌浏览器，下载地址“https://www.google.cn/intl/zh-CN/chrome/”。***



图17

2、缴费按钮，可以根据本期设置开放的考区进行选择，可以选择所在的省市以及考点的类型，只有D级的考生才可以选择纸质类型，系统将自动给考生分配考位信息。



图18

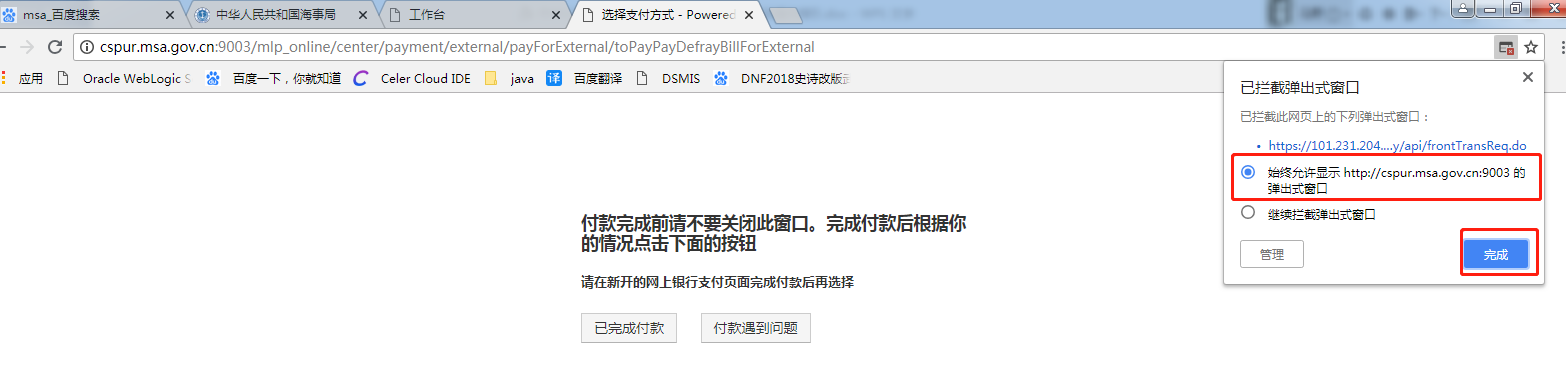
1. 点击“确认”按钮，进入缴费页面，选择支付方式，进行支付。

***注：本次试用不实际缴费，请您选择企业缴费，点击 “确认支付”按钮。***

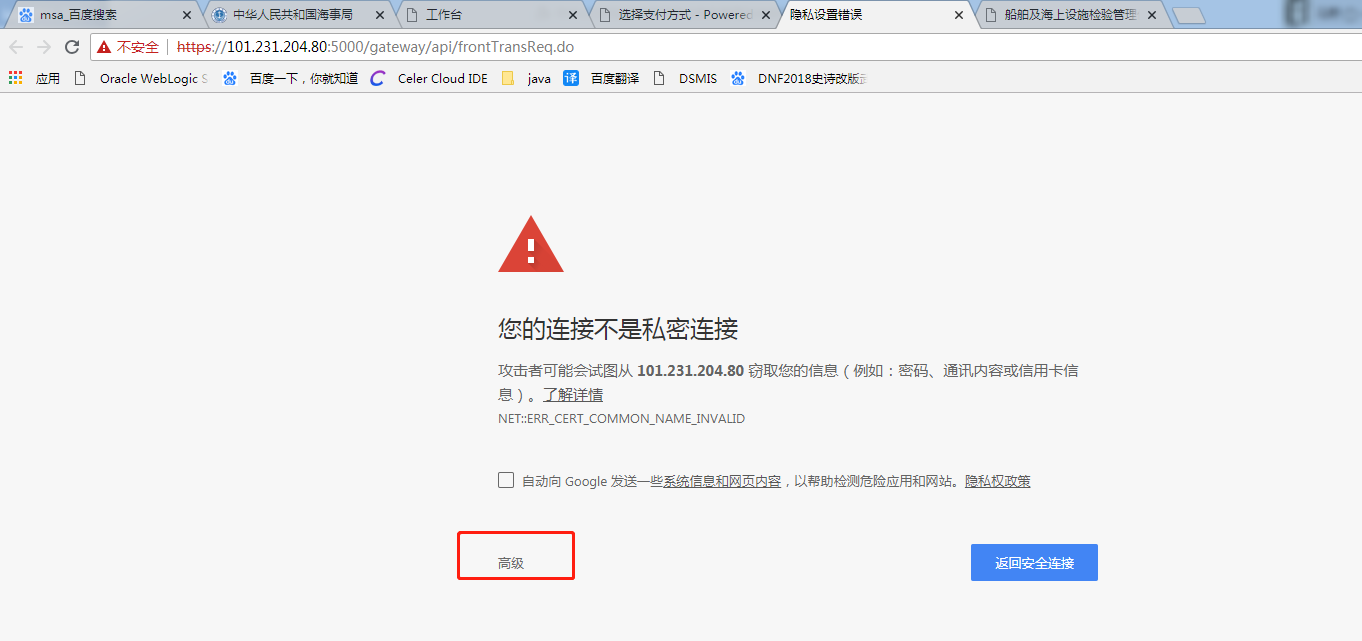


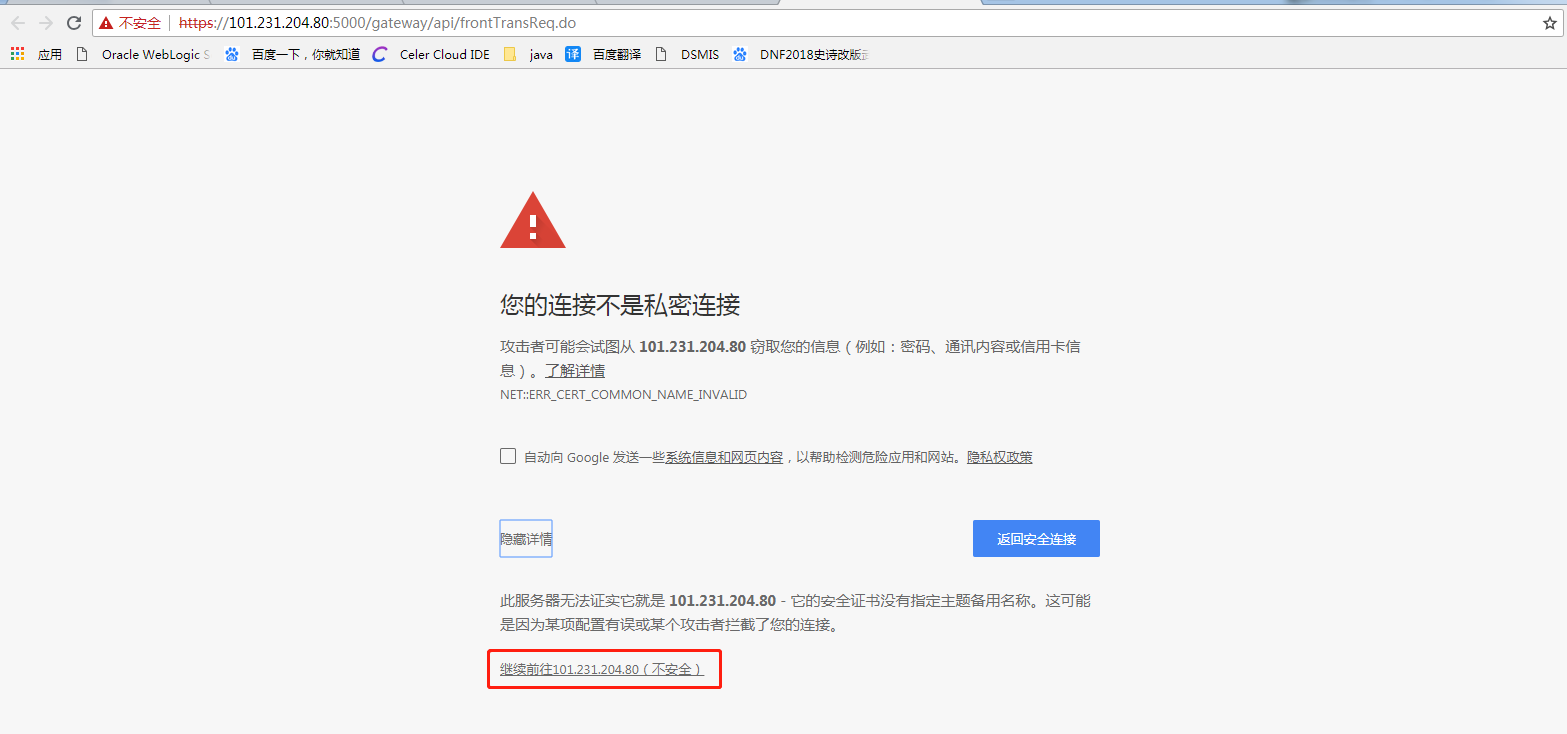
图19

点击“确认支付”按钮之后，弹窗被拦截，点击“允许弹出”。



点击“确认支付”后，可能会出现访问“不是私密链接”，点击“高级”，再点击“[继续前往101.231.204.80（不安全）](data:text/html;charset=utf-8,<!doctype html>%0A<html dir="ltr" lang="zh">%0A<head>%0A  <meta charset="utf-8">%0A  <meta name="viewport"%0A      content="initial-scale=1, minimum-scale=1, width=device-width">%0A  <title>%E9%9A%90%E7%A7%81%E8%AE%BE%E7%BD%AE%E9%94%99%E8%AF%AF</title>%0A  <style>/* Copyright 2017 The Chromium Authors. All rights reserved.%0A * Use of this source code is governed by a BSD-style license that can be%0A * found in the LICENSE file. */%0A%0Aa {%0A  color: rgb(88, 88, 88);%0A}%0A%0Abody {%0A  background-color: rgb(247, 247, 247);%0A  color: rgb(100, 100, 100);%0A  word-wrap: break-word;%0A}%0A%0A%23details-button {%0A  background: inherit;%0A  border: 0;%0A  float: none;%0A  margin: 0;%0A  padding: 10px 0;%0A  text-transform: uppercase;%0A}%0A%0A.hidden {%0A  display: none;%0A}%0A%0Ahtml {%0A  -webkit-text-size-adjust: 100%;%0A  font-size: 125%;%0A}%0A%0A.icon {%0A  background-repeat: no-repeat;%0A  background-size: 100%;%0A}</style>%0A  <style>/* Copyright 2014 The Chromium Authors. All rights reserved.%0A   Use of this source code is governed by a BSD-style license that can be%0A   found in the LICENSE file. */%0A%0Abutton {%0A  border: 0;%0A  border-radius: 2px;%0A  box-sizing: border-box;%0A  color: %23fff;%0A  cursor: pointer;%0A  float: right;%0A  font-size: .875em;%0A  margin: 0;%0A  padding: 10px 24px;%0A  transition: box-shadow 200ms cubic-bezier(0.4, 0, 0.2, 1);%0A  user-select: none;%0A}%0A%0A[dir='rtl'] button {%0A  float: left;%0A}%0A%0A.bad-clock button,%0A.captive-portal button,%0A.main-frame-blocked button,%0A.neterror button,%0A.offline button,%0A.pdf button,%0A.ssl button {%0A  background: rgb(66, 133, 244);%0A}%0A%0Abutton:active {%0A  background: rgb(50, 102, 213);%0A  outline: 0;%0A}%0A%0Abutton:hover {%0A  box-shadow: 0 1px 3px rgba(0, 0, 0, .50);%0A}%0A%0A%23debugging {%0A  display: inline;%0A  overflow: auto;%0A}%0A%0A.debugging-content {%0A  line-height: 1em;%0A  margin-bottom: 0;%0A  margin-top: 1em;%0A}%0A%0A.debugging-content-fixed-width {%0A  display: block;%0A  font-family: monospace;%0A  font-size: 1.2em;%0A  margin-top: 0.5em;%0A}%0A%0A.debugging-title {%0A  font-weight: bold;%0A}%0A%0A%23details {%0A  color: %23696969;%0A  margin: 0 0 50px;%0A}%0A%0A%23details p:not(:first-of-type) {%0A  margin-top: 20px;%0A}%0A%0A%23details-button:hover {%0A  box-shadow: inherit;%0A  text-decoration: underline;%0A}%0A%0A.error-code {%0A  color: %23646464;%0A  font-size: .86667em;%0A  text-transform: uppercase;%0A}%0A%0A%23error-debugging-info {%0A  font-size: 0.8em;%0A}%0A%0Ah1 {%0A  color: %23333;%0A  font-size: 1.6em;%0A  font-weight: normal;%0A  line-height: 1.25em;%0A  margin-bottom: 16px;%0A}%0A%0Ah2 {%0A  font-size: 1.2em;%0A  font-weight: normal;%0A}%0A%0A.icon {%0A  height: 72px;%0A  margin: 0 0 40px;%0A  width: 72px;%0A}%0A%0Ainput[type=checkbox] {%0A  opacity: 0;%0A}%0A%0Ainput[type=checkbox]:focus ~ .checkbox {%0A  outline: -webkit-focus-ring-color auto 5px;%0A}%0A%0A.interstitial-wrapper {%0A  box-sizing: border-box;%0A  font-size: 1em;%0A  line-height: 1.6em;%0A  margin: 14vh auto 0;%0A  max-width: 600px;%0A  width: 100%;%0A}%0A%0A%23main-message > p {%0A  display: inline;%0A}%0A%0A%23extended-reporting-opt-in {%0A  font-size: .875em;%0A  margin-top: 39px;%0A}%0A%0A%23extended-reporting-opt-in label {%0A  position: relative;%0A  display: flex;%0A  align-items: flex-start;%0A}%0A%0A.nav-wrapper {%0A  margin-top: 51px;%0A}%0A%0A.nav-wrapper::after {%0A  clear: both;%0A  content: '';%0A  display: table;%0A  width: 100%;%0A}%0A%0A.small-link {%0A  color: %23696969;%0A  font-size: .875em;%0A}%0A%0A.checkboxes {%0A  flex: 0 0 24px;%0A}%0A%0A.checkbox {%0A  background: transparent;%0A  border: 1px solid white;%0A  border-radius: 2px;%0A  display: block;%0A  height: 14px;%0A  left: 0;%0A  position: absolute;%0A  right: 0;%0A  top: 3px;%0A  width: 14px;%0A}%0A%0A.checkbox::before {%0A  background: transparent;%0A  border: 2px solid white;%0A  border-right-width: 0;%0A  border-top-width: 0;%0A  content: '';%0A  height: 4px;%0A  left: 2px;%0A  opacity: 0;%0A  position: absolute;%0A  top: 3px;%0A  transform: rotate(-45deg);%0A  width: 9px;%0A}%0A%0Ainput[type=checkbox]:checked ~ .checkbox::before {%0A  opacity: 1;%0A}%0A%0A@media (max-width: 700px) {%0A  .interstitial-wrapper {%0A    padding: 0 10%;%0A  }%0A%0A  %23error-debugging-info {%0A    overflow: auto;%0A  }%0A}%0A%0A@media (max-height: 600px) {%0A  .error-code {%0A    margin-top: 10px;%0A  }%0A}%0A%0A@media (max-width: 420px) {%0A  button,%0A  [dir='rtl'] button,%0A  .small-link {%0A    float: none;%0A    font-size: .825em;%0A    font-weight: 400;%0A    margin: 0;%0A    text-transform: uppercase;%0A    width: 100%;%0A  }%0A%0A  %23details {%0A    margin: 20px 0 20px 0;%0A  }%0A%0A  %23details p:not(:first-of-type) {%0A    margin-top: 10px;%0A  }%0A%0A  %23details-button {%0A    display: block;%0A    margin-top: 20px;%0A    text-align: center;%0A    width: 100%;%0A  }%0A%0A  .interstitial-wrapper {%0A    padding: 0 5%;%0A  }%0A%0A  %23extended-reporting-opt-in {%0A    margin-top: 24px;%0A  }%0A%0A  .nav-wrapper {%0A    margin-top: 30px;%0A  }%0A}%0A%0A/**%0A * Mobile specific styling.%0A * Navigation buttons are anchored to the bottom of the screen.%0A * Details message replaces the top content in its own scrollable area.%0A */%0A%0A@media (max-width: 420px) {%0A  %23details-button {%0A    border: 0;%0A    margin: 8px 0 0;%0A  }%0A%0A  .secondary-button {%0A    -webkit-margin-end: 0;%0A    margin-top: 16px;%0A  }%0A}%0A%0A/* Fixed nav. */%0A@media (min-width: 240px) and (max-width: 420px) and%0A       (min-height: 401px),%0A       (min-width: 421px) and (min-height: 240px) and%0A       (max-height: 560px) {%0A  body .nav-wrapper {%0A    background: %23f7f7f7;%0A    bottom: 0;%0A    box-shadow: 0 -22px 40px rgb(247, 247, 247);%0A    left: 0;%0A    margin: 0 auto;%0A    max-width: 736px;%0A    padding-left: 24px;%0A    padding-right: 24px;%0A    position: fixed;%0A    right: 0;%0A    width: 100%;%0A    z-index: 2;%0A  }%0A%0A  .interstitial-wrapper {%0A    max-width: 736px;%0A  }%0A%0A  %23details,%0A  %23main-content {%0A    padding-bottom: 40px;%0A  }%0A%0A  %23details {%0A    padding-top: 5.5vh;%0A  }%0A%0A  %23details-button:hover {%0A    box-shadow: none;%0A  }%0A}%0A%0A@media (max-width: 420px) and (orientation: portrait),%0A       (max-height: 560px) {%0A  body {%0A    margin: 0 auto;%0A  }%0A%0A  button,%0A  [dir='rtl'] button,%0A  button.small-link {%0A    font-family: Roboto-Regular,Helvetica;%0A    font-size: .933em;%0A    font-weight: 600;%0A    margin: 6px 0;%0A    text-transform: uppercase;%0A    transform: translatez(0);%0A  }%0A%0A  .nav-wrapper {%0A    box-sizing: border-box;%0A    padding-bottom: 8px;%0A    width: 100%;%0A  }%0A%0A  .error-code {%0A    margin-top: 0;%0A  }%0A%0A  %23details {%0A    box-sizing: border-box;%0A    height: auto;%0A    margin: 0;%0A    opacity: 1;%0A    transition: opacity 250ms cubic-bezier(0.4, 0, 0.2, 1);%0A  }%0A%0A  %23details.hidden,%0A  %23main-content.hidden {%0A    display: block;%0A    height: 0;%0A    opacity: 0;%0A    overflow: hidden;%0A    padding-bottom: 0;%0A    transition: none;%0A  }%0A%0A  %23details-button {%0A    padding-bottom: 16px;%0A    padding-top: 16px;%0A  }%0A%0A  h1 {%0A    font-size: 1.5em;%0A    margin-bottom: 8px;%0A  }%0A%0A  .icon {%0A    margin-bottom: 5.69vh;%0A  }%0A%0A  .interstitial-wrapper {%0A    box-sizing: border-box;%0A    margin: 7vh auto 12px;%0A    padding: 0 24px;%0A    position: relative;%0A  }%0A%0A  .interstitial-wrapper p {%0A    font-size: .95em;%0A    line-height: 1.61em;%0A    margin-top: 8px;%0A  }%0A%0A  %23main-content {%0A    margin: 0;%0A    transition: opacity 100ms cubic-bezier(0.4, 0, 0.2, 1);%0A  }%0A%0A  .small-link {%0A    border: 0;%0A  }%0A%0A  .suggested-left > %23control-buttons,%0A  .suggested-right > %23control-buttons {%0A    float: none;%0A    margin: 0;%0A  }%0A}%0A%0A@media (min-width: 421px) and (min-height: 500px) and (max-height: 560px) {%0A  .interstitial-wrapper {%0A    margin-top: 10vh;%0A  }%0A}%0A%0A@media (min-height: 400px) and (orientation:portrait) {%0A  .interstitial-wrapper {%0A    margin-bottom: 145px;%0A  }%0A}%0A%0A@media (min-height: 299px) {%0A  .nav-wrapper {%0A    padding-bottom: 16px;%0A  }%0A}%0A%0A@media (min-height: 500px) and (max-height: 650px) and (max-width: 414px) and%0A       (orientation: portrait) {%0A  .interstitial-wrapper {%0A    margin-top: 7vh;%0A  }%0A}%0A%0A@media (min-height: 650px) and (max-width: 414px) and (orientation: portrait) {%0A  .interstitial-wrapper {%0A    margin-top: 10vh;%0A  }%0A}%0A%0A/* Small mobile screens. No fixed nav. */%0A@media (max-height: 400px) and (orientation: portrait),%0A       (max-height: 239px) and (orientation: landscape),%0A       (max-width: 419px) and (max-height: 399px) {%0A  .interstitial-wrapper {%0A    display: flex;%0A    flex-direction: column;%0A    margin-bottom: 0;%0A  }%0A%0A  %23details {%0A    flex: 1 1 auto;%0A    order: 0;%0A  }%0A%0A  %23main-content {%0A    flex: 1 1 auto;%0A    order: 0;%0A  }%0A%0A  .nav-wrapper {%0A    flex: 0 1 auto;%0A    margin-top: 8px;%0A    order: 1;%0A    padding-left: 0;%0A    padding-right: 0;%0A    position: relative;%0A    width: 100%;%0A  }%0A}%0A%0A@media (max-width: 239px) and (orientation: portrait) {%0A  .nav-wrapper {%0A    padding-left: 0;%0A    padding-right: 0;%0A  }%0A}%0A</style>%0A  <style>/* Copyright 2017 The Chromium Authors. All rights reserved.%0A * Use of this source code is governed by a BSD-style license that can be%0A * found in the LICENSE file. */%0A%0A.bad-clock .icon {%0A  background-image: -webkit-image-set(%0A      url(data:image/png;base64,) 1x,%0A      url(data:image/png;base64,) 2x);%0A}%0A</style>%0A  <style>/* Copyright 2017 The Chromium Authors. All rights reserved.%0A * Use of this source code is governed by a BSD-style license that can be%0A * found in the LICENSE file. */%0A%0A.captive-portal .icon {%0A  background-image: -webkit-image-set(%0A      url(data:image/png;base64,) 1x,%0A      url(data:image/png;base64,) 2x);%0A}%0A</style>%0A  <style>/* Copyright 2017 The Chromium Authors. All rights reserved.%0A * Use of this source code is governed by a BSD-style license that can be%0A * found in the LICENSE file. */%0A%0Abody.safe-browsing {%0A  background-color: rgb(206, 52, 38);%0A  color: white;%0A}%0A%0A.safe-browsing :-webkit-any(%0A    a, %23details, %23details-button, h1, h2, p, .small-link) {%0A  color: white;%0A}%0A%0A.safe-browsing button {%0A  background-color: rgba(255, 255, 255, .15);%0A}%0A%0A.safe-browsing button:active {%0A  background-color: rgba(255, 255, 255, .25);%0A}%0A%0A.safe-browsing button:hover {%0A  box-shadow: 0 2px 3px rgba(0, 0, 0, .5);%0A}%0A%0A.safe-browsing .error-code {%0A  display: none;%0A}%0A%0A.safe-browsing .icon {%0A  background-image: -webkit-image-set(%0A      url(data:image/png;base64,) 1x,%0A      url(data:image/png;base64,) 2x);%0A}%0A%0A@media (min-width: 240px) and (max-width: 420px) and%0A       (min-height: 401px),%0A       (min-width: 421px) and (min-height: 240px) and%0A       (max-height: 560px) {%0A  body.safe-browsing .nav-wrapper {%0A    background: rgb(206, 52, 38);%0A    box-shadow: 0 -22px 40px rgb(206, 52, 38);%0A  }%0A}</style>%0A  <style>/* Copyright 2017 The Chromium Authors. All rights reserved.%0A * Use of this source code is governed by a BSD-style license that can be%0A * found in the LICENSE file. */%0A%0A.ssl .icon {%0A  background-image: -webkit-image-set(%0A      url(data:image/png;base64,) 1x,%0A      url(data:image/png;base64,) 2x);%0A}%0A%0A.ssl-opt-in .checkbox {%0A  border-color: %23696969;%0A}%0A%0A.ssl-opt-in .checkbox::before {%0A  border-color: %23696969;%0A}%0A</style>%0A  <script>// Copyright (c) 2012 The Chromium Authors. All rights reserved.%0A// Use of this source code is governed by a BSD-style license that can be%0A// found in the LICENSE file.%0A%0A// // Copyright (c) 2013 The Chromium Authors. All rights reserved.%0A// Use of this source code is governed by a BSD-style license that can be%0A// found in the LICENSE file.%0A%0A/**%0A * @fileoverview Assertion support.%0A */%0A%0A/**%0A * Verify |condition| is truthy and return |condition| if so.%0A * @template T%0A * @param {T} condition A condition to check for truthiness.  Note that this%0A *     may be used to test whether a value is defined or not, and we don't want%0A *     to force a cast to Boolean.%0A * @param {string=} opt_message A message to show on failure.%0A * @return {T} A non-null |condition|.%0A */%0Afunction assert(condition, opt_message) {%0A  if (!condition) {%0A    var message = 'Assertion failed';%0A    if (opt_message)%0A      message = message + ': ' + opt_message;%0A    var error = new Error(message);%0A    var global = function() {%0A      return this;%0A    }();%0A    if (global.traceAssertionsForTesting)%0A      console.warn(error.stack);%0A    throw error;%0A  }%0A  return condition;%0A}%0A%0A/**%0A * Call this from places in the code that should never be reached.%0A *%0A * For example, handling all the values of enum with a switch() like this:%0A *%0A *   function getValueFromEnum(enum) {%0A *     switch (enum) {%0A *       case ENUM_FIRST_OF_TWO:%0A *         return first%0A *       case ENUM_LAST_OF_TWO:%0A *         return last;%0A *     }%0A *     assertNotReached();%0A *     return document;%0A *   }%0A *%0A * This code should only be hit in the case of serious programmer error or%0A * unexpected input.%0A *%0A * @param {string=} opt_message A message to show when this is hit.%0A */%0Afunction assertNotReached(opt_message) {%0A  assert(false, opt_message || 'Unreachable code hit');%0A}%0A%0A/**%0A * @param {*} value The value to check.%0A * @param {function(new: T, ...)} type A user-defined constructor.%0A * @param {string=} opt_message A message to show when this is hit.%0A * @return {T}%0A * @template T%0A */%0Afunction assertInstanceof(value, type, opt_message) {%0A  // We don't use assert immediately here so that we avoid constructing an error%0A  // message if we don't have to.%0A  if (!(value instanceof type)) {%0A    assertNotReached(%0A        opt_message ||%0A        'Value ' + value + ' is not a[n] ' + (type.name || typeof type));%0A  }%0A  return value;%0A}%0A%0A%0A/**%0A * Alias for document.getElementById. Found elements must be HTMLElements.%0A * @param {string} id The ID of the element to find.%0A * @return {HTMLElement} The found element or null if not found.%0A */%0Afunction $(id) {%0A  // Disable getElementById restriction here, since we are instructing other%0A  // places to re-use the $() that is defined here.%0A  // eslint-disable-next-line no-restricted-properties%0A  var el = document.getElementById(id);%0A  return el ? assertInstanceof(el, HTMLElement) : null;%0A}%0A%0A// TODO(devlin): This should return SVGElement, but closure compiler is missing%0A// those externs.%0A/**%0A * Alias for document.getElementById. Found elements must be SVGElements.%0A * @param {string} id The ID of the element to find.%0A * @return {Element} The found element or null if not found.%0A */%0Afunction getSVGElement(id) {%0A  // Disable getElementById restriction here, since it is not suitable for SVG%0A  // elements.%0A  // eslint-disable-next-line no-restricted-properties%0A  var el = document.getElementById(id);%0A  return el ? assertInstanceof(el, Element) : null;%0A}%0A%0A/**%0A * Add an accessible message to the page that will be announced to%0A * users who have spoken feedback on, but will be invisible to all%0A * other users. It's removed right away so it doesn't clutter the DOM.%0A * @param {string} msg The text to be pronounced.%0A */%0Afunction announceAccessibleMessage(msg) {%0A  var element = document.createElement('div');%0A  element.setAttribute('aria-live', 'polite');%0A  element.style.position = 'fixed';%0A  element.style.left = '-9999px';%0A  element.style.height = '0px';%0A  element.innerText = msg;%0A  document.body.appendChild(element);%0A  window.setTimeout(function() {%0A    document.body.removeChild(element);%0A  }, 0);%0A}%0A%0A/**%0A * Generates a CSS url string.%0A * @param {string} s The URL to generate the CSS url for.%0A * @return {string} The CSS url string.%0A */%0Afunction getUrlForCss(s) {%0A  // http://www.w3.org/TR/css3-values/%23uris%0A  // Parentheses, commas, whitespace characters, single quotes (') and double%0A  // quotes (") appearing in a URI must be escaped with a backslash%0A  var s2 = s.replace(/(\\(|\\)|\\,|\\s|\\'|\\"|\\\\)/g, '\\\\$1');%0A  // WebKit has a bug when it comes to URLs that end with \\%0A  // https://bugs.webkit.org/show_bug.cgi?id=28885%0A  if (/\\\\\\\\$/.test(s2)) {%0A    // Add a space to work around the WebKit bug.%0A    s2 += ' ';%0A  }%0A  return 'url("' + s2 + '")';%0A}%0A%0A/**%0A * Parses query parameters from Location.%0A * @param {Location} location The URL to generate the CSS url for.%0A * @return {Object} Dictionary containing name value pairs for URL%0A */%0Afunction parseQueryParams(location) {%0A  var params = {};%0A  var query = unescape(location.search.substring(1));%0A  var vars = query.split('&');%0A  for (var i = 0; i < vars.length; i++) {%0A    var pair = vars[i].split('=');%0A    params[pair[0]] = pair[1];%0A  }%0A  return params;%0A}%0A%0A/**%0A * Creates a new URL by appending or replacing the given query key and value.%0A * Not supporting URL with username and password.%0A * @param {Location} location The original URL.%0A * @param {string} key The query parameter name.%0A * @param {string} value The query parameter value.%0A * @return {string} The constructed new URL.%0A */%0Afunction setQueryParam(location, key, value) {%0A  var query = parseQueryParams(location);%0A  query[encodeURIComponent(key)] = encodeURIComponent(value);%0A%0A  var newQuery = '';%0A  for (var q in query) {%0A    newQuery += (newQuery ? '&' : '?') + q + '=' + query[q];%0A  }%0A%0A  return location.origin + location.pathname + newQuery + location.hash;%0A}%0A%0A/**%0A * @param {Node} el A node to search for ancestors with |className|.%0A * @param {string} className A class to search for.%0A * @return {Element} A node with class of |className| or null if none is found.%0A */%0Afunction findAncestorByClass(el, className) {%0A  return /** @type {Element} */ (findAncestor(el, function(el) {%0A    return el.classList && el.classList.contains(className);%0A  }));%0A}%0A%0A/**%0A * Return the first ancestor for which the {@code predicate} returns true.%0A * @param {Node} node The node to check.%0A * @param {function(Node):boolean} predicate The function that tests the%0A *     nodes.%0A * @return {Node} The found ancestor or null if not found.%0A */%0Afunction findAncestor(node, predicate) {%0A  var last = false;%0A  while (node != null && !(last = predicate(node))) {%0A    node = node.parentNode;%0A  }%0A  return last ? node : null;%0A}%0A%0Afunction swapDomNodes(a, b) {%0A  var afterA = a.nextSibling;%0A  if (afterA == b) {%0A    swapDomNodes(b, a);%0A    return;%0A  }%0A  var aParent = a.parentNode;%0A  b.parentNode.replaceChild(a, b);%0A  aParent.insertBefore(b, afterA);%0A}%0A%0A/**%0A * Disables text selection and dragging, with optional whitelist callbacks.%0A * @param {function(Event):boolean=} opt_allowSelectStart Unless this function%0A *    is defined and returns true, the onselectionstart event will be%0A *    surpressed.%0A * @param {function(Event):boolean=} opt_allowDragStart Unless this function%0A *    is defined and returns true, the ondragstart event will be surpressed.%0A */%0Afunction disableTextSelectAndDrag(opt_allowSelectStart, opt_allowDragStart) {%0A  // Disable text selection.%0A  document.onselectstart = function(e) {%0A    if (!(opt_allowSelectStart && opt_allowSelectStart.call(this, e)))%0A      e.preventDefault();%0A  };%0A%0A  // Disable dragging.%0A  document.ondragstart = function(e) {%0A    if (!(opt_allowDragStart && opt_allowDragStart.call(this, e)))%0A      e.preventDefault();%0A  };%0A}%0A%0A/**%0A * Check the directionality of the page.%0A * @return {boolean} True if Chrome is running an RTL UI.%0A */%0Afunction isRTL() {%0A  return document.documentElement.dir == 'rtl';%0A}%0A%0A/**%0A * Get an element that's known to exist by its ID. We use this instead of just%0A * calling getElementById and not checking the result because this lets us%0A * satisfy the JSCompiler type system.%0A * @param {string} id The identifier name.%0A * @return {!HTMLElement} the Element.%0A */%0Afunction getRequiredElement(id) {%0A  return assertInstanceof(%0A      $(id), HTMLElement, 'Missing required element: ' + id);%0A}%0A%0A/**%0A * Query an element that's known to exist by a selector. We use this instead of%0A * just calling querySelector and not checking the result because this lets us%0A * satisfy the JSCompiler type system.%0A * @param {string} selectors CSS selectors to query the element.%0A * @param {(!Document|!DocumentFragment|!Element)=} opt_context An optional%0A *     context object for querySelector.%0A * @return {!HTMLElement} the Element.%0A */%0Afunction queryRequiredElement(selectors, opt_context) {%0A  var element = (opt_context || document).querySelector(selectors);%0A  return assertInstanceof(%0A      element, HTMLElement, 'Missing required element: ' + selectors);%0A}%0A%0A// Handle click on a link. If the link points to a chrome: or file: url, then%0A// call into the browser to do the navigation.%0A['click', 'auxclick'].forEach(function(eventName) {%0A  document.addEventListener(eventName, function(e) {%0A    if (e.button > 1)%0A      return;  // Ignore buttons other than left and middle.%0A    if (e.defaultPrevented)%0A      return;%0A%0A    var eventPath = e.path;%0A    var anchor = null;%0A    if (eventPath) {%0A      for (var i = 0; i < eventPath.length; i++) {%0A        var element = eventPath[i];%0A        if (element.tagName === 'A' && element.href) {%0A          anchor = element;%0A          break;%0A        }%0A      }%0A    }%0A%0A    // Fallback if Event.path is not available.%0A    var el = e.target;%0A    if (!anchor && el.nodeType == Node.ELEMENT_NODE &&%0A        el.webkitMatchesSelector('A, A *')) {%0A      while (el.tagName != 'A') {%0A        el = el.parentElement;%0A      }%0A      anchor = el;%0A    }%0A%0A    if (!anchor)%0A      return;%0A%0A    anchor = /** @type {!HTMLAnchorElement} */ (anchor);%0A    if ((anchor.protocol == 'file:' || anchor.protocol == 'about:') &&%0A        (e.button == 0 || e.button == 1)) {%0A      chrome.send('navigateToUrl', [%0A        anchor.href, anchor.target, e.button, e.altKey, e.ctrlKey, e.metaKey,%0A        e.shiftKey%0A      ]);%0A      e.preventDefault();%0A    }%0A  });%0A});%0A%0A/**%0A * Creates a new URL which is the old URL with a GET param of key=value.%0A * @param {string} url The base URL. There is not sanity checking on the URL so%0A *     it must be passed in a proper format.%0A * @param {string} key The key of the param.%0A * @param {string} value The value of the param.%0A * @return {string} The new URL.%0A */%0Afunction appendParam(url, key, value) {%0A  var param = encodeURIComponent(key) + '=' + encodeURIComponent(value);%0A%0A  if (url.indexOf('?') == -1)%0A    return url + '?' + param;%0A  return url + '&' + param;%0A}%0A%0A/**%0A * Creates an element of a specified type with a specified class name.%0A * @param {string} type The node type.%0A * @param {string} className The class name to use.%0A * @return {Element} The created element.%0A */%0Afunction createElementWithClassName(type, className) {%0A  var elm = document.createElement(type);%0A  elm.className = className;%0A  return elm;%0A}%0A%0A/**%0A * transitionend does not always fire (e.g. when animation is aborted%0A * or when no paint happens during the animation). This function sets up%0A * a timer and emulate the event if it is not fired when the timer expires.%0A * @param {!HTMLElement} el The element to watch for transitionend.%0A * @param {number=} opt_timeOut The maximum wait time in milliseconds for the%0A *     transitionend to happen. If not specified, it is fetched from |el|%0A *     using the transitionDuration style value.%0A */%0Afunction ensureTransitionEndEvent(el, opt_timeOut) {%0A  if (opt_timeOut === undefined) {%0A    var style = getComputedStyle(el);%0A    opt_timeOut = parseFloat(style.transitionDuration) * 1000;%0A%0A    // Give an additional 50ms buffer for the animation to complete.%0A    opt_timeOut += 50;%0A  }%0A%0A  var fired = false;%0A  el.addEventListener('transitionend', function f(e) {%0A    el.removeEventListener('transitionend', f);%0A    fired = true;%0A  });%0A  window.setTimeout(function() {%0A    if (!fired)%0A      cr.dispatchSimpleEvent(el, 'transitionend', true);%0A  }, opt_timeOut);%0A}%0A%0A/**%0A * Alias for document.scrollTop getter.%0A * @param {!HTMLDocument} doc The document node where information will be%0A *     queried from.%0A * @return {number} The Y document scroll offset.%0A */%0Afunction scrollTopForDocument(doc) {%0A  return doc.documentElement.scrollTop || doc.body.scrollTop;%0A}%0A%0A/**%0A * Alias for document.scrollTop setter.%0A * @param {!HTMLDocument} doc The document node where information will be%0A *     queried from.%0A * @param {number} value The target Y scroll offset.%0A */%0Afunction setScrollTopForDocument(doc, value) {%0A  doc.documentElement.scrollTop = doc.body.scrollTop = value;%0A}%0A%0A/**%0A * Alias for document.scrollLeft getter.%0A * @param {!HTMLDocument} doc The document node where information will be%0A *     queried from.%0A * @return {number} The X document scroll offset.%0A */%0Afunction scrollLeftForDocument(doc) {%0A  return doc.documentElement.scrollLeft || doc.body.scrollLeft;%0A}%0A%0A/**%0A * Alias for document.scrollLeft setter.%0A * @param {!HTMLDocument} doc The document node where information will be%0A *     queried from.%0A * @param {number} value The target X scroll offset.%0A */%0Afunction setScrollLeftForDocument(doc, value) {%0A  doc.documentElement.scrollLeft = doc.body.scrollLeft = value;%0A}%0A%0A/**%0A * Replaces '&', '<', '>', '"', and ''' characters with their HTML encoding.%0A * @param {string} original The original string.%0A * @return {string} The string with all the characters mentioned above replaced.%0A */%0Afunction HTMLEscape(original) {%0A  return original.replace(/&/g, '&amp;')%0A      .replace(/</g, '&lt;')%0A      .replace(/>/g, '&gt;')%0A      .replace(/"/g, '&quot;')%0A      .replace(/'/g, '&%2339;');%0A}%0A%0A/**%0A * Shortens the provided string (if necessary) to a string of length at most%0A * |maxLength|.%0A * @param {string} original The original string.%0A * @param {number} maxLength The maximum length allowed for the string.%0A * @return {string} The original string if its length does not exceed%0A *     |maxLength|. Otherwise the first |maxLength| - 1 characters with '...'%0A *     appended.%0A */%0Afunction elide(original, maxLength) {%0A  if (original.length <= maxLength)%0A    return original;%0A  return original.substring(0, maxLength - 1) + '\\u2026';%0A}%0A%0A/**%0A * Quote a string so it can be used in a regular expression.%0A * @param {string} str The source string.%0A * @return {string} The escaped string.%0A */%0Afunction quoteString(str) {%0A  return str.replace(/([\\\\\\.\\+\\*\\?\\[\\^\\]\\$\\(\\)\\{\\}\\=\\!\\<\\>\\|\\:])/g, '\\\\$1');%0A}%0A%0A/**%0A * Calls |callback| and stops listening the first time any event in |eventNames|%0A * is triggered on |target|.%0A * @param {!EventTarget} target%0A * @param {!Array<string>|string} eventNames Array or space-delimited string of%0A *     event names to listen to (e.g. 'click mousedown').%0A * @param {function(!Event)} callback Called at most once. The%0A *     optional return value is passed on by the listener.%0A */%0Afunction listenOnce(target, eventNames, callback) {%0A  if (!Array.isArray(eventNames))%0A    eventNames = eventNames.split(/ +/);%0A%0A  var removeAllAndCallCallback = function(event) {%0A    eventNames.forEach(function(eventName) {%0A      target.removeEventListener(eventName, removeAllAndCallCallback, false);%0A    });%0A    return callback(event);%0A  };%0A%0A  eventNames.forEach(function(eventName) {%0A    target.addEventListener(eventName, removeAllAndCallCallback, false);%0A  });%0A}%0A%0A//   /* is_ios */%0A%0A/**%0A * @param {!Event} e%0A * @return {boolean} Whether a modifier key was down when processing |e|.%0A */%0Afunction hasKeyModifiers(e) {%0A  return !!(e.altKey || e.ctrlKey || e.metaKey || e.shiftKey);%0A}%0A</script>%0A  <script>// Copyright 2014 The Chromium Authors. All rights reserved.%0A// Use of this source code is governed by a BSD-style license that can be%0A// found in the LICENSE file.%0A%0Avar CAPTIVEPORTAL_CMD_OPEN_LOGIN_PAGE = 'openLoginPage';%0A</script>%0A  <script>// Copyright 2014 The Chromium Authors. All rights reserved.%0A// Use of this source code is governed by a BSD-style license that can be%0A// found in the LICENSE file.%0A%0Afunction setupSSLDebuggingInfo() {%0A  if (loadTimeData.getString('type') != 'SSL')%0A    return;%0A%0A  // The titles are not internationalized because this is debugging information%0A  // for bug reports, help center posts, etc.%0A  appendDebuggingField('Subject', loadTimeData.getString('subject'));%0A  appendDebuggingField('Issuer', loadTimeData.getString('issuer'));%0A  appendDebuggingField('Expires on', loadTimeData.getString('expirationDate'));%0A  appendDebuggingField('Current date', loadTimeData.getString('currentDate'));%0A  appendDebuggingField('PEM encoded chain', loadTimeData.getString('pem'),%0A                       true);%0A%0A  $('error-code').addEventListener('click', toggleDebuggingInfo);%0A}%0A</script>%0A  <script>// Copyright 2015 The Chromium Authors. All rights reserved.%0A// Use of this source code is governed by a BSD-style license that can be%0A// found in the LICENSE file.%0A%0A'use strict';%0A%0A// Other constants defined in security_interstitial_page.h.%0Avar SB_BOX_CHECKED = 'boxchecked';%0Avar SB_DISPLAY_CHECK_BOX = 'displaycheckbox';%0A%0A// This sets up the Extended Safe Browsing Reporting opt-in, either for%0A// reporting malware or invalid certificate chains. Does nothing if the%0A// interstitial type is not SAFEBROWSING or SSL or CAPTIVE_PORTAL.%0Afunction setupExtendedReportingCheckbox() {%0A  var interstitialType = loadTimeData.getString('type');%0A  if (interstitialType != 'SAFEBROWSING' && interstitialType != 'SSL' &&%0A      interstitialType != 'CAPTIVE_PORTAL') {%0A    return;%0A  }%0A%0A  if (!loadTimeData.getBoolean(SB_DISPLAY_CHECK_BOX)) {%0A    return;%0A  }%0A%0A  $('opt-in-label').innerHTML = loadTimeData.getString('optInLink');%0A  $('opt-in-checkbox').checked = loadTimeData.getBoolean(SB_BOX_CHECKED);%0A  $('extended-reporting-opt-in').classList.remove('hidden');%0A%0A  var className = interstitialType == 'SAFEBROWSING' ?%0A                  'safe-browsing-opt-in' :%0A                  'ssl-opt-in';%0A  $('extended-reporting-opt-in').classList.add(className);%0A%0A  $('body').classList.add('extended-reporting-has-checkbox');%0A%0A  if ($('whitepaper-link')) {%0A    $('whitepaper-link').addEventListener('click', function(event) {%0A      sendCommand(SecurityInterstitialCommandId.CMD_OPEN_WHITEPAPER);%0A    });%0A  }%0A%0A  $('opt-in-checkbox').addEventListener('click', function() {%0A    sendCommand($('opt-in-checkbox').checked ?%0A                SecurityInterstitialCommandId.CMD_DO_REPORT :%0A                SecurityInterstitialCommandId.CMD_DONT_REPORT);%0A  });%0A}%0A</script>%0A  <script>// Copyright 2015 The Chromium Authors. All rights reserved.%0A// Use of this source code is governed by a BSD-style license that can be%0A// found in the LICENSE file.%0A%0Avar mobileNav = false;%0A%0A/**%0A * For small screen mobile the navigation buttons are moved%0A * below the advanced text.%0A */%0Afunction onResize() {%0A  var helpOuterBox = document.querySelector('%23details');%0A  var mainContent = document.querySelector('%23main-content');%0A  var mediaQuery = '(min-width: 240px) and (max-width: 420px) and ' +%0A      '(min-height: 401px), ' +%0A      '(max-height: 560px) and (min-height: 240px) and ' +%0A      '(min-width: 421px)';%0A%0A  var detailsHidden = helpOuterBox.classList.contains(HIDDEN_CLASS);%0A  var runnerContainer = document.querySelector('.runner-container');%0A%0A  // Check for change in nav status.%0A  if (mobileNav != window.matchMedia(mediaQuery).matches) {%0A    mobileNav = !mobileNav;%0A%0A    // Handle showing the top content / details sections according to state.%0A    if (mobileNav) {%0A      mainContent.classList.toggle(HIDDEN_CLASS, !detailsHidden);%0A      helpOuterBox.classList.toggle(HIDDEN_CLASS, detailsHidden);%0A      if (runnerContainer) {%0A        runnerContainer.classList.toggle(HIDDEN_CLASS, !detailsHidden);%0A      }%0A    } else if (!detailsHidden) {%0A      // Non mobile nav with visible details.%0A      mainContent.classList.remove(HIDDEN_CLASS);%0A      helpOuterBox.classList.remove(HIDDEN_CLASS);%0A      if (runnerContainer) {%0A        runnerContainer.classList.remove(HIDDEN_CLASS);%0A      }%0A    }%0A  }%0A}%0A%0Afunction setupMobileNav() {%0A  window.addEventListener('resize', onResize);%0A  onResize();%0A}%0A%0Adocument.addEventListener('DOMContentLoaded', setupMobileNav);%0A</script>%0A  <script>// Copyright 2017 The Chromium Authors. All rights reserved.%0A// Use of this source code is governed by a BSD-style license that can be%0A// found in the LICENSE file.%0A%0A// This is the shared code for security interstitials. It is used for both SSL%0A// interstitials and Safe Browsing interstitials.%0A%0A// Should match security_interstitials::SecurityInterstitialCommand%0A/** @enum| {string} */%0Avar SecurityInterstitialCommandId = {%0A  CMD_DONT_PROCEED: 0,%0A  CMD_PROCEED: 1,%0A  // Ways for user to get more information%0A  CMD_SHOW_MORE_SECTION: 2,%0A  CMD_OPEN_HELP_CENTER: 3,%0A  CMD_OPEN_DIAGNOSTIC: 4,%0A  // Primary button actions%0A  CMD_RELOAD: 5,%0A  CMD_OPEN_DATE_SETTINGS: 6,%0A  CMD_OPEN_LOGIN: 7,%0A  // Safe Browsing Extended Reporting%0A  CMD_DO_REPORT: 8,%0A  CMD_DONT_REPORT: 9,%0A  CMD_OPEN_REPORTING_PRIVACY: 10,%0A  CMD_OPEN_WHITEPAPER: 11,%0A  // Report a phishing error.%0A  CMD_REPORT_PHISHING_ERROR: 12%0A};%0A%0Avar HIDDEN_CLASS = 'hidden';%0A%0A/**%0A * A convenience method for sending commands to the parent page.%0A * @param {string} cmd  The command to send.%0A */%0Afunction sendCommand(cmd) {%0A  if (window.certificateErrorPageController) {%0A    switch (cmd) {%0A      case SecurityInterstitialCommandId.CMD_DONT_PROCEED:%0A        certificateErrorPageController.dontProceed();%0A        break;%0A      case SecurityInterstitialCommandId.CMD_PROCEED:%0A        certificateErrorPageController.proceed();%0A        break;%0A      case SecurityInterstitialCommandId.CMD_SHOW_MORE_SECTION:%0A        certificateErrorPageController.showMoreSection();%0A        break;%0A      case SecurityInterstitialCommandId.CMD_OPEN_HELP_CENTER:%0A        certificateErrorPageController.openHelpCenter();%0A        break;%0A      case SecurityInterstitialCommandId.CMD_OPEN_DIAGNOSTIC:%0A        certificateErrorPageController.openDiagnostic();%0A        break;%0A      case SecurityInterstitialCommandId.CMD_RELOAD:%0A        certificateErrorPageController.reload();%0A        break;%0A      case SecurityInterstitialCommandId.CMD_OPEN_DATE_SETTINGS:%0A        certificateErrorPageController.openDateSettings();%0A        break;%0A      case SecurityInterstitialCommandId.CMD_OPEN_LOGIN:%0A        certificateErrorPageController.openLogin();%0A        break;%0A      case SecurityInterstitialCommandId.CMD_DO_REPORT:%0A        certificateErrorPageController.doReport();%0A        break;%0A      case SecurityInterstitialCommandId.CMD_DONT_REPORT:%0A        certificateErrorPageController.dontReport();%0A        break;%0A      case SecurityInterstitialCommandId.CMD_OPEN_REPORTING_PRIVACY:%0A        certificateErrorPageController.openReportingPrivacy();%0A        break;%0A      case SecurityInterstitialCommandId.CMD_OPEN_WHITEPAPER:%0A        certificateErrorPageController.openWhitepaper();%0A        break;%0A      case SecurityInterstitialCommandId.CMD_REPORT_PHISHING_ERROR:%0A        certificateErrorPageController.reportPhishingError();%0A        break;%0A    }%0A    return;%0A  }%0A// %0A  window.domAutomationController.send(cmd);%0A// %0A// %0A}%0A%0A/**%0A * Call this to stop clicks on <a href="%23"> links from scrolling to the top of%0A * the page (and possibly showing a %23 in the link).%0A */%0Afunction preventDefaultOnPoundLinkClicks() {%0A  document.addEventListener('click', function(e) {%0A    var anchor = findAncestor(/** @type {Node} */ (e.target), function(el) {%0A      return el.tagName == 'A';%0A    });%0A    // Use getAttribute() to prevent URL normalization.%0A    if (anchor && anchor.getAttribute('href') == '%23')%0A      e.preventDefault();%0A  });%0A}%0A</script>%0A  <script>// Copyright 2014 The Chromium Authors. All rights reserved.%0A// Use of this source code is governed by a BSD-style license that can be%0A// found in the LICENSE file.%0A%0A// This is the shared code for the new (Chrome 37) security interstitials. It is%0A// used for both SSL interstitials and Safe Browsing interstitials.%0A%0Avar expandedDetails = false;%0Avar keyPressState = 0;%0A%0A/**%0A * This allows errors to be skippped by typing a secret phrase into the page.%0A * @param {string} e The key that was just pressed.%0A */%0Afunction handleKeypress(e) {%0A  // HTTPS errors are serious and should not be ignored. For testing purposes,%0A  // other approaches are both safer and have fewer side-effects.%0A  // See https://goo.gl/ZcZixP for more details.%0A  var BYPASS_SEQUENCE = window.atob('dGhpc2lzdW5zYWZl');%0A  if (BYPASS_SEQUENCE.charCodeAt(keyPressState) == e.keyCode) {%0A    keyPressState++;%0A    if (keyPressState == BYPASS_SEQUENCE.length) {%0A      sendCommand(SecurityInterstitialCommandId.CMD_PROCEED);%0A      keyPressState = 0;%0A    }%0A  } else {%0A    keyPressState = 0;%0A  }%0A}%0A%0A/**%0A * This appends a piece of debugging information to the end of the warning.%0A * When complete, the caller must also make the debugging div%0A * (error-debugging-info) visible.%0A * @param {string} title  The name of this debugging field.%0A * @param {string} value  The value of the debugging field.%0A * @param {boolean=} fixedWidth If true, the value field is displayed fixed%0A *                              width.%0A */%0Afunction appendDebuggingField(title, value, fixedWidth) {%0A  // The values input here are not trusted. Never use innerHTML on these%0A  // values!%0A  var spanTitle = document.createElement('span');%0A  spanTitle.classList.add('debugging-title');%0A  spanTitle.innerText = title + ': ';%0A%0A  var spanValue = document.createElement('span');%0A  spanValue.classList.add('debugging-content');%0A  if (fixedWidth) {%0A    spanValue.classList.add('debugging-content-fixed-width');%0A  }%0A  spanValue.innerText = value;%0A%0A  var pElem = document.createElement('p');%0A  pElem.classList.add('debugging-content');%0A  pElem.appendChild(spanTitle);%0A  pElem.appendChild(spanValue);%0A  $('error-debugging-info').appendChild(pElem);%0A}%0A%0Afunction toggleDebuggingInfo() {%0A  $('error-debugging-info').classList.toggle(HIDDEN_CLASS);%0A}%0A%0Afunction setupEvents() {%0A  var overridable = loadTimeData.getBoolean('overridable');%0A  var interstitialType = loadTimeData.getString('type');%0A  var ssl = interstitialType == 'SSL';%0A  var captivePortal = interstitialType == 'CAPTIVE_PORTAL';%0A  var badClock = ssl && loadTimeData.getBoolean('bad_clock');%0A  var hidePrimaryButton = loadTimeData.getBoolean('hide_primary_button');%0A%0A  if (ssl) {%0A    $('body').classList.add(badClock ? 'bad-clock' : 'ssl');%0A    $('error-code').textContent = loadTimeData.getString('errorCode');%0A    $('error-code').classList.remove(HIDDEN_CLASS);%0A  } else if (captivePortal) {%0A    $('body').classList.add('captive-portal');%0A  } else {%0A    $('body').classList.add('safe-browsing');%0A  }%0A%0A  $('icon').classList.add('icon');%0A%0A  if (hidePrimaryButton) {%0A    $('primary-button').classList.add(HIDDEN_CLASS);%0A  } else {%0A    $('primary-button').addEventListener('click', function() {%0A      switch (interstitialType) {%0A        case 'CAPTIVE_PORTAL':%0A          sendCommand(SecurityInterstitialCommandId.CMD_OPEN_LOGIN);%0A          break;%0A%0A        case 'SSL':%0A          if (badClock)%0A            sendCommand(SecurityInterstitialCommandId.CMD_OPEN_DATE_SETTINGS);%0A          else if (overridable)%0A            sendCommand(SecurityInterstitialCommandId.CMD_DONT_PROCEED);%0A          else%0A            sendCommand(SecurityInterstitialCommandId.CMD_RELOAD);%0A          break;%0A%0A        case 'SAFEBROWSING':%0A          sendCommand(SecurityInterstitialCommandId.CMD_DONT_PROCEED);%0A          break;%0A%0A        default:%0A          throw 'Invalid interstitial type';%0A      }%0A    });%0A  }%0A%0A  if (overridable) {%0A    // Captive portal page isn't overridable.%0A    $('proceed-link').addEventListener('click', function(event) {%0A      sendCommand(SecurityInterstitialCommandId.CMD_PROCEED);%0A    });%0A  } else if (!ssl) {%0A    $('final-paragraph').classList.add(HIDDEN_CLASS);%0A  }%0A%0A  if (ssl && overridable) {%0A    $('proceed-link').classList.add('small-link');%0A  }%0A%0A  if ($('diagnostic-link')) {%0A    $('diagnostic-link').addEventListener('click', function(event) {%0A      sendCommand(SecurityInterstitialCommandId.CMD_OPEN_DIAGNOSTIC);%0A    });%0A  }%0A%0A  if ($('learn-more-link')) {%0A    $('learn-more-link').addEventListener('click', function(event) {%0A      sendCommand(SecurityInterstitialCommandId.CMD_OPEN_HELP_CENTER);%0A    });%0A  }%0A%0A  if (captivePortal) {%0A    // Captive portal page doesn't have details button.%0A    $('details-button').classList.add('hidden');%0A  } else {%0A    $('details-button').addEventListener('click', function(event) {%0A      var hiddenDetails = $('details').classList.toggle(HIDDEN_CLASS);%0A%0A      if (mobileNav) {%0A        // Details appear over the main content on small screens.%0A        $('main-content').classList.toggle(HIDDEN_CLASS, !hiddenDetails);%0A      } else {%0A        $('main-content').classList.remove(HIDDEN_CLASS);%0A      }%0A%0A      $('details-button').innerText = hiddenDetails ?%0A          loadTimeData.getString('openDetails') :%0A          loadTimeData.getString('closeDetails');%0A      if (!expandedDetails) {%0A        // Record a histogram entry only the first time that details is opened.%0A        sendCommand(SecurityInterstitialCommandId.CMD_SHOW_MORE_SECTION);%0A        expandedDetails = true;%0A      }%0A    });%0A  }%0A%0A  if ($('report-error-link')) {%0A    $('report-error-link').addEventListener('click', function(event) {%0A      sendCommand(SecurityInterstitialCommandId.CMD_REPORT_PHISHING_ERROR);%0A    });%0A  }%0A%0A  preventDefaultOnPoundLinkClicks();%0A  setupExtendedReportingCheckbox();%0A  setupSSLDebuggingInfo();%0A  document.addEventListener('keypress', handleKeypress);%0A}%0A%0Adocument.addEventListener('DOMContentLoaded', setupEvents);%0A</script>%0A</head>%0A<body id="body">%0A  <div class="interstitial-wrapper">%0A    <div id="main-content">%0A      <div class="icon" id="icon"></div>%0A      <div id="main-message">%0A        <h1>%E6%82%A8%E7%9A%84%E8%BF%9E%E6%8E%A5%E4%B8%8D%E6%98%AF%E7%A7%81%E5%AF%86%E8%BF%9E%E6%8E%A5</h1>%0A        <p>%E6%94%BB%E5%87%BB%E8%80%85%E5%8F%AF%E8%83%BD%E4%BC%9A%E8%AF%95%E5%9B%BE%E4%BB%8E <strong>101.231.204.80</strong> %E7%AA%83%E5%8F%96%E6%82%A8%E7%9A%84%E4%BF%A1%E6%81%AF%EF%BC%88%E4%BE%8B%E5%A6%82%EF%BC%9A%E5%AF%86%E7%A0%81%E3%80%81%E9%80%9A%E8%AE%AF%E5%86%85%E5%AE%B9%E6%88%96%E4%BF%A1%E7%94%A8%E5%8D%A1%E4%BF%A1%E6%81%AF%EF%BC%89%E3%80%82<a href="%23" id="learn-more-link">%E4%BA%86%E8%A7%A3%E8%AF%A6%E6%83%85</a></p>%0A        <div id="debugging">%0A          <div id="error-code" class="error-code"></div>%0A          <div id="error-debugging-info" class="hidden"></div>%0A        </div>%0A      </div>%0A      <div id="extended-reporting-opt-in" class="hidden">%0A        <label>%0A          <div class="checkboxes">%0A            <input type="checkbox" id="opt-in-checkbox">%0A            <span class="checkbox"></span>%0A          </div>%0A          <span id="opt-in-label"></span>%0A        </label>%0A      </div>%0A    </div>%0A    <div class="nav-wrapper">%0A      <button id="primary-button">%E8%BF%94%E5%9B%9E%E5%AE%89%E5%85%A8%E8%BF%9E%E6%8E%A5</button>%0A      <button id="details-button" class="small-link">%0A        %E9%AB%98%E7%BA%A7%0A      </button>%0A    </div>%0A    <div id="details" class="hidden">%0A      <p>%E6%AD%A4%E6%9C%8D%E5%8A%A1%E5%99%A8%E6%97%A0%E6%B3%95%E8%AF%81%E5%AE%9E%E5%AE%83%E5%B0%B1%E6%98%AF <strong>101.231.204.80</strong> - %E5%AE%83%E7%9A%84%E5%AE%89%E5%85%A8%E8%AF%81%E4%B9%A6%E6%B2%A1%E6%9C%89%E6%8C%87%E5%AE%9A%E4%B8%BB%E9%A2%98%E5%A4%87%E7%94%A8%E5%90%8D%E7%A7%B0%E3%80%82%E8%BF%99%E5%8F%AF%E8%83%BD%E6%98%AF%E5%9B%A0%E4%B8%BA%E6%9F%90%E9%A1%B9%E9%85%8D%E7%BD%AE%E6%9C%89%E8%AF%AF%E6%88%96%E6%9F%90%E4%B8%AA%E6%94%BB%E5%87%BB%E8%80%85%E6%8B%A6%E6%88%AA%E4%BA%86%E6%82%A8%E7%9A%84%E8%BF%9E%E6%8E%A5%E3%80%82</p>%0A      <p id="final-paragraph"><a href="%23" id="proceed-link">%E7%BB%A7%E7%BB%AD%E5%89%8D%E5%BE%80101.231.204.80%EF%BC%88%E4%B8%8D%E5%AE%89%E5%85%A8%EF%BC%89</a></p>%0A    </div>%0A  </div>%0A</body>%0A</html>%0A<style>/* Copyright 2014 The Chromium Authors. All rights reserved.%0A * Use of this source code is governed by a BSD-style license that can be%0A * found in the LICENSE file. */%0A%0A/* This file is dynamically processed by a C++ data source handler to fill in%0A * some per-platform/locale styles that dramatically alter the page. This is%0A * done to reduce flicker, as JS may not run before the page is rendered.%0A *%0A * There are two ways to include this stylesheet:%0A * 1. via its chrome://resources/ URL in HTML, i.e.:%0A *%0A *   <link rel="stylesheet" href="chrome://resources/css/text_defaults.css">%0A *%0A * 2. via the webui::AppendWebUICSSTextDefaults() method to directly append it%0A * to an HTML string.%0A * Otherwise its placeholders won't be expanded. */%0A%0Ahtml {%0A  direction: ltr;%0A}%0A%0Abody {%0A  font-family: 'Segoe UI',Arial,'Microsoft Yahei',sans-serif;%0A  font-size: 75%;%0A}%0A%0Abutton {%0A  font-family: 'Segoe UI',Arial,'Microsoft Yahei',sans-serif;%0A}%0A</style><script>// Copyright (c) 2012 The Chromium Authors. All rights reserved.%0A// Use of this source code is governed by a BSD-style license that can be%0A// found in the LICENSE file.%0A%0A/**%0A * @fileoverview This file defines a singleton which provides access to all data%0A * that is available as soon as the page's resources are loaded (before DOM%0A * content has finished loading). This data includes both localized strings and%0A * any data that is important to have ready from a very early stage (e.g. things%0A * that must be displayed right away).%0A *%0A * Note that loadTimeData is not guaranteed to be consistent between page%0A * refreshes (https://crbug.com/740629) and should not contain values that might%0A * change if the page is re-opened later.%0A */%0A%0A/**%0A * @typedef {{%0A *   substitutions: (Array<string>|undefined),%0A *   attrs: (Object<function(Node, string):boolean>|undefined),%0A *   tags: (Array<string>|undefined),%0A * }}%0A */%0Avar SanitizeInnerHtmlOpts;%0A%0A/** @type {!LoadTimeData} */ var loadTimeData;%0A%0A// Expose this type globally as a temporary work around until%0A// https://github.com/google/closure-compiler/issues/544 is fixed.%0A/** @constructor */%0Afunction LoadTimeData(){}%0A%0A(function() {%0A  'use strict';%0A%0A  LoadTimeData.prototype = {%0A    /**%0A     * Sets the backing object.%0A     *%0A     * Note that there is no getter for |data_| to discourage abuse of the form:%0A     *%0A     *     var value = loadTimeData.data()['key'];%0A     *%0A     * @param {Object} value The de-serialized page data.%0A     */%0A    set data(value) {%0A      expect(!this.data_, 'Re-setting data.');%0A      this.data_ = value;%0A    },%0A%0A    /**%0A     * Returns a JsEvalContext for |data_|.%0A     * @returns {JsEvalContext}%0A     */%0A    createJsEvalContext: function() {%0A      return new JsEvalContext(this.data_);%0A    },%0A%0A    /**%0A     * @param {string} id An ID of a value that might exist.%0A     * @return {boolean} True if |id| is a key in the dictionary.%0A     */%0A    valueExists: function(id) {%0A      return id in this.data_;%0A    },%0A%0A    /**%0A     * Fetches a value, expecting that it exists.%0A     * @param {string} id The key that identifies the desired value.%0A     * @return {*} The corresponding value.%0A     */%0A    getValue: function(id) {%0A      expect(this.data_, 'No data. Did you remember to include strings.js?');%0A      var value = this.data_[id];%0A      expect(typeof value != 'undefined', 'Could not find value for ' + id);%0A      return value;%0A    },%0A%0A    /**%0A     * As above, but also makes sure that the value is a string.%0A     * @param {string} id The key that identifies the desired string.%0A     * @return {string} The corresponding string value.%0A     */%0A    getString: function(id) {%0A      var value = this.getValue(id);%0A      expectIsType(id, value, 'string');%0A      return /** @type {string} */ (value);%0A    },%0A%0A    /**%0A     * Returns a formatted localized string where $1 to $9 are replaced by the%0A     * second to the tenth argument.%0A     * @param {string} id The ID of the string we want.%0A     * @param {...(string|number)} var_args The extra values to include in the%0A     *     formatted output.%0A     * @return {string} The formatted string.%0A     */%0A    getStringF: function(id, var_args) {%0A      var value = this.getString(id);%0A      if (!value)%0A        return '';%0A%0A      var args = Array.prototype.slice.call(arguments);%0A      args[0] = value;%0A      return this.substituteString.apply(this, args);%0A    },%0A%0A    /**%0A     * Make a string safe for use with with Polymer bindings that are%0A     * inner-h-t-m-l (or other innerHTML use).%0A     * @param {string} rawString The unsanitized string.%0A     * @param {SanitizeInnerHtmlOpts=} opts Optional additional allowed tags and%0A     *     attributes.%0A     * @return {string}%0A     */%0A    sanitizeInnerHtml: function(rawString, opts) {%0A      opts = opts || {};%0A      return parseHtmlSubset('<b>' + rawString + '</b>', opts.tags, opts.attrs)%0A          .firstChild.innerHTML;%0A    },%0A%0A    /**%0A     * Returns a formatted localized string where $1 to $9 are replaced by the%0A     * second to the tenth argument. Any standalone $ signs must be escaped as%0A     * $$.%0A     * @param {string} label The label to substitute through.%0A     *     This is not an resource ID.%0A     * @param {...(string|number)} var_args The extra values to include in the%0A     *     formatted output.%0A     * @return {string} The formatted string.%0A     */%0A    substituteString: function(label, var_args) {%0A      var varArgs = arguments;%0A      return label.replace(/\\$(.|$|\\n)/g, function(m) {%0A        assert(m.match(/\\$[$1-9]/), 'Unescaped $ found in localized string.');%0A        return m == '$$' ? '$' : varArgs[m[1]];%0A      });%0A    },%0A%0A    /**%0A     * Returns a formatted string where $1 to $9 are replaced by the second to%0A     * tenth argument, split apart into a list of pieces describing how the%0A     * substitution was performed. Any standalone $ signs must be escaped as $$.%0A     * @param {string} label A localized string to substitute through.%0A     *     This is not an resource ID.%0A     * @param {...(string|number)} var_args The extra values to include in the%0A     *     formatted output.%0A     * @return {!Array<!{value: string, arg: (null|string)}>} The formatted%0A     *     string pieces.%0A     */%0A    getSubstitutedStringPieces: function(label, var_args) {%0A      var varArgs = arguments;%0A      // Split the string by separately matching all occurrences of $1-9 and of%0A      // non $1-9 pieces.%0A      var pieces = (label.match(/(\\$[1-9])|(([^$]|\\$([^1-9]|$))+)/g) ||%0A                    []).map(function(p) {%0A        // Pieces that are not $1-9 should be returned after replacing $$%0A        // with $.%0A        if (!p.match(/^\\$[1-9]$/)) {%0A          assert(%0A              (p.match(/\\$/g) || []).length % 2 == 0,%0A              'Unescaped $ found in localized string.');%0A          return {value: p.replace(/\\$\\$/g, '$'), arg: null};%0A        }%0A%0A        // Otherwise, return the substitution value.%0A        return {value: varArgs[p[1]], arg: p};%0A      });%0A%0A      return pieces;%0A    },%0A%0A    /**%0A     * As above, but also makes sure that the value is a boolean.%0A     * @param {string} id The key that identifies the desired boolean.%0A     * @return {boolean} The corresponding boolean value.%0A     */%0A    getBoolean: function(id) {%0A      var value = this.getValue(id);%0A      expectIsType(id, value, 'boolean');%0A      return /** @type {boolean} */ (value);%0A    },%0A%0A    /**%0A     * As above, but also makes sure that the value is an integer.%0A     * @param {string} id The key that identifies the desired number.%0A     * @return {number} The corresponding number value.%0A     */%0A    getInteger: function(id) {%0A      var value = this.getValue(id);%0A      expectIsType(id, value, 'number');%0A      expect(value == Math.floor(value), 'Number isn\\'t integer: ' + value);%0A      return /** @type {number} */ (value);%0A    },%0A%0A    /**%0A     * Override values in loadTimeData with the values found in |replacements|.%0A     * @param {Object} replacements The dictionary object of keys to replace.%0A     */%0A    overrideValues: function(replacements) {%0A      expect(%0A          typeof replacements == 'object',%0A          'Replacements must be a dictionary object.');%0A      for (var key in replacements) {%0A        this.data_[key] = replacements[key];%0A      }%0A    }%0A  };%0A%0A  /**%0A   * Checks condition, displays error message if expectation fails.%0A   * @param {*} condition The condition to check for truthiness.%0A   * @param {string} message The message to display if the check fails.%0A   */%0A  function expect(condition, message) {%0A    if (!condition) {%0A      console.error(%0A          'Unexpected condition on ' + document.location.href + ': ' + message);%0A    }%0A  }%0A%0A  /**%0A   * Checks that the given value has the given type.%0A   * @param {string} id The id of the value (only used for error message).%0A   * @param {*} value The value to check the type on.%0A   * @param {string} type The type we expect |value| to be.%0A   */%0A  function expectIsType(id, value, type) {%0A    expect(%0A        typeof value == type, '[' + value + '] (' + id + ') is not a ' + type);%0A  }%0A%0A  expect(!loadTimeData, 'should only include this file once');%0A  loadTimeData = new LoadTimeData;%0A})();%0A</script><script>loadTimeData.data = {"bad_clock":false,"boxchecked":false,"closeDetails":"%E9%9A%90%E8%97%8F%E8%AF%A6%E6%83%85","currentDate":"2018%E5%B9%B46%E6%9C%8822%E6%97%A5","displaycheckbox":true,"errorCode":"net::ERR_CERT_COMMON_NAME_INVALID","expirationDate":"2020%E5%B9%B44%E6%9C%885%E6%97%A5","explanationParagraph":"%E6%AD%A4%E6%9C%8D%E5%8A%A1%E5%99%A8%E6%97%A0%E6%B3%95%E8%AF%81%E5%AE%9E%E5%AE%83%E5%B0%B1%E6%98%AF \\u003Cstrong>101.231.204.80\\u003C/strong> - %E5%AE%83%E7%9A%84%E5%AE%89%E5%85%A8%E8%AF%81%E4%B9%A6%E6%B2%A1%E6%9C%89%E6%8C%87%E5%AE%9A%E4%B8%BB%E9%A2%98%E5%A4%87%E7%94%A8%E5%90%8D%E7%A7%B0%E3%80%82%E8%BF%99%E5%8F%AF%E8%83%BD%E6%98%AF%E5%9B%A0%E4%B8%BA%E6%9F%90%E9%A1%B9%E9%85%8D%E7%BD%AE%E6%9C%89%E8%AF%AF%E6%88%96%E6%9F%90%E4%B8%AA%E6%94%BB%E5%87%BB%E8%80%85%E6%8B%A6%E6%88%AA%E4%BA%86%E6%82%A8%E7%9A%84%E8%BF%9E%E6%8E%A5%E3%80%82","finalParagraph":"\\u003Ca href=\\"%23\\" id=\\"proceed-link\\">%E7%BB%A7%E7%BB%AD%E5%89%8D%E5%BE%80101.231.204.80%EF%BC%88%E4%B8%8D%E5%AE%89%E5%85%A8%EF%BC%89\\u003C/a>","fontfamily":"'Segoe UI',Arial,'Microsoft Yahei',sans-serif","fontsize":"75%","heading":"%E6%82%A8%E7%9A%84%E8%BF%9E%E6%8E%A5%E4%B8%8D%E6%98%AF%E7%A7%81%E5%AF%86%E8%BF%9E%E6%8E%A5","hide_primary_button":false,"issuer":"101.231.204.80","language":"zh","openDetails":"%E9%AB%98%E7%BA%A7","optInLink":"%E8%87%AA%E5%8A%A8%E5%90%91 Google %E5%8F%91%E9%80%81%E4%B8%80%E4%BA%9B\\u003Ca href=\\"%23\\" id=\\"whitepaper-link\\">%E7%B3%BB%E7%BB%9F%E4%BF%A1%E6%81%AF%E5%92%8C%E7%BD%91%E9%A1%B5%E5%86%85%E5%AE%B9\\u003C/a>%EF%BC%8C%E4%BB%A5%E5%B8%AE%E5%8A%A9%E6%A3%80%E6%B5%8B%E5%8D%B1%E9%99%A9%E5%BA%94%E7%94%A8%E5%92%8C%E7%BD%91%E7%AB%99%E3%80%82\\u003Ca id=\\"privacy-link\\" href=\\"%23\\" onclick=\\"sendCommand(10); return false;\\" onmousedown=\\"return false;\\">%E9%9A%90%E7%A7%81%E6%9D%83%E6%94%BF%E7%AD%96\\u003C/a>","overridable":true,"pem":"-----BEGIN CERTIFICATE-----\\nMIIDYTCCAkmgAwIBAgIRAMu1LWMSMhA7yPYnL4/rwZ0wDQYJKoZIhvcNAQELBQAw\\nbDELMAkGA1UEBhMCQ04xEDAOBgNVBAgTB2JlaWppbmcxEDAOBgNVBAcTB2JlaWpp\\nbmcxDTALBgNVBAoTBGNmY2ExETAPBgNVBAsTCExvY2FsIFJBMRcwFQYDVQQDEw4x\\nMDEuMjMxLjIwNC44MDAeFw0xODA0MDYxMDI5MDFaFw0yMDA0MDUxMDI5MDFaMGwx\\nCzAJBgNVBAYTAkNOMRAwDgYDVQQIEwdiZWlqaW5nMRAwDgYDVQQHEwdiZWlqaW5n\\nMQ0wCwYDVQQKEwRjZmNhMREwDwYDVQQLEwhMb2NhbCBSQTEXMBUGA1UEAxMOMTAx\\nLjIzMS4yMDQuODAwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDtc1bq\\nDL9aVhZKrUkzfm0aCfiIb1+E/23F+Hrh4T5ocz+gEPYyixKf/6nQNg4ySymWRKcu\\nUSfI0T1/3kRRosFbDuRduxpI14m7RlW1dpXOAzwggPGX2Ub+AfbIbPAoD6jT0dg4\\nx5CH6pgJi6kHf5xEDG1yuEVN94D/S37E21zxMn2OHYOFK/bRxb/j9UUWuUUQXGf4\\n8P3xBt20qlaXz+pSw5kDlPg8PLmeq8bfj45iGry51ZGKwunG/fabJzGQHYWqvsxZ\\nVwAm/t5HX4/PYMYzlUtDrLfN89lUrRQaht206tQgCzFchbLvAt9fhzomuVypUBwb\\nqtPhGA8fjAYhukivAgMBAAEwDQYJKoZIhvcNAQELBQADggEBAOhdzJFO+9usfCa5\\nDTbghltAnayq5BRBidCd1xPIDoxhcH00uOs5C8kvUV7VnOoEqm+SvFst6XPGZNTD\\n1b7zBu5jHofXIvPaGQEv9VDzImrDPcd1nnMc0NhKlKvr6dDIwONlRgqnSbBDUzc2\\nXgbSntuOkWI8q3uVLknMrUdfJHMl5aHloZd/BWLymvNCTrIHRFASJ2uBAQCOTl96\\nR4MTxq/uC2ZhdjHUat+g805yzrSUxrYClbZw2tt9hGB8kqFKUUZSPaKTdPTZ0oe8\\nT4Khz1rQFye2rgKxPvLl2aRP31GeopBOtCi9hNGzZCJX3yht8LHbx0Ls9ZKRmnu8\\ng9HqK3I=\\n-----END CERTIFICATE-----\\n","primaryButtonText":"%E8%BF%94%E5%9B%9E%E5%AE%89%E5%85%A8%E8%BF%9E%E6%8E%A5","primaryParagraph":"%E6%94%BB%E5%87%BB%E8%80%85%E5%8F%AF%E8%83%BD%E4%BC%9A%E8%AF%95%E5%9B%BE%E4%BB%8E \\u003Cstrong>101.231.204.80\\u003C/strong> %E7%AA%83%E5%8F%96%E6%82%A8%E7%9A%84%E4%BF%A1%E6%81%AF%EF%BC%88%E4%BE%8B%E5%A6%82%EF%BC%9A%E5%AF%86%E7%A0%81%E3%80%81%E9%80%9A%E8%AE%AF%E5%86%85%E5%AE%B9%E6%88%96%E4%BF%A1%E7%94%A8%E5%8D%A1%E4%BF%A1%E6%81%AF%EF%BC%89%E3%80%82\\u003Ca href=\\"%23\\" id=\\"learn-more-link\\">%E4%BA%86%E8%A7%A3%E8%AF%A6%E6%83%85\\u003C/a>","subject":"101.231.204.80","tabTitle":"%E9%9A%90%E7%A7%81%E8%AE%BE%E7%BD%AE%E9%94%99%E8%AF%AF","textdirection":"ltr","type":"SSL"};</script><script>// Copyright (c) 2012 The Chromium Authors. All rights reserved.%0A// Use of this source code is governed by a BSD-style license that can be%0A// found in the LICENSE file.%0A%0A// Note: vulcanize sometimes disables GRIT processing. If you're importing i18n%0A// stuff with <link rel="import">, you should probably be using%0A// html/i18n_template.html instead of this file.%0A%0A// // Copyright (c) 2012 The Chromium Authors. All rights reserved.%0A// Use of this source code is governed by a BSD-style license that can be%0A// found in the LICENSE file.%0A%0A/** @typedef {Document|DocumentFragment|Element} */%0Avar ProcessingRoot;%0A%0A/**%0A * @fileoverview This is a simple template engine inspired by JsTemplates%0A * optimized for i18n.%0A *%0A * It currently supports three handlers:%0A *%0A *   * i18n-content which sets the textContent of the element.%0A *%0A *     <span i18n-content="myContent"></span>%0A *%0A *   * i18n-options which generates <option> elements for a <select>.%0A *%0A *     <select i18n-options="myOptionList"></select>%0A *%0A *   * i18n-values is a list of attribute-value or property-value pairs.%0A *     Properties are prefixed with a '.' and can contain nested properties.%0A *%0A *     <span i18n-values="title:myTitle;.style.fontSize:fontSize"></span>%0A *%0A * This file is a copy of i18n_template.js, with minor tweaks to support using%0A * load_time_data.js. It should replace i18n_template.js eventually.%0A */%0A%0Avar i18nTemplate = (function() {%0A  /**%0A   * This provides the handlers for the templating engine. The key is used as%0A   * the attribute name and the value is the function that gets called for every%0A   * single node that has this attribute.%0A   * @type {!Object}%0A   */%0A  var handlers = {%0A    /**%0A     * This handler sets the textContent of the element.%0A     * @param {!HTMLElement} element The node to modify.%0A     * @param {string} key The name of the value in |data|.%0A     * @param {!LoadTimeData} data The data source to draw from.%0A     * @param {!Set<ProcessingRoot>} visited%0A     */%0A    'i18n-content': function(element, key, data, visited) {%0A      element.textContent = data.getString(key);%0A    },%0A%0A    /**%0A     * This handler adds options to a <select> element.%0A     * @param {!HTMLElement} select The node to modify.%0A     * @param {string} key The name of the value in |data|. It should%0A     *     identify an array of values to initialize an <option>. Each value,%0A     *     if a pair, represents [content, value]. Otherwise, it should be a%0A     *     content string with no value.%0A     * @param {!LoadTimeData} data The data source to draw from.%0A     * @param {!Set<ProcessingRoot>} visited%0A     */%0A    'i18n-options': function(select, key, data, visited) {%0A      var options = data.getValue(key);%0A      options.forEach(function(optionData) {%0A        var option = typeof optionData == 'string' ?%0A            new Option(optionData) :%0A            new Option(optionData[1], optionData[0]);%0A        select.appendChild(option);%0A      });%0A    },%0A%0A    /**%0A     * This is used to set HTML attributes and DOM properties. The syntax is:%0A     *   attributename:key;%0A     *   .domProperty:key;%0A     *   .nested.dom.property:key%0A     * @param {!HTMLElement} element The node to modify.%0A     * @param {string} attributeAndKeys The path of the attribute to modify%0A     *     followed by a colon, and the name of the value in |data|.%0A     *     Multiple attribute/key pairs may be separated by semicolons.%0A     * @param {!LoadTimeData} data The data source to draw from.%0A     * @param {!Set<ProcessingRoot>} visited%0A     */%0A    'i18n-values': function(element, attributeAndKeys, data, visited) {%0A      var parts = attributeAndKeys.replace(/\\s/g, '').split(/;/);%0A      parts.forEach(function(part) {%0A        if (!part)%0A          return;%0A%0A        var attributeAndKeyPair = part.match(/^([^:]+):(.+)$/);%0A        if (!attributeAndKeyPair)%0A          throw new Error('malformed i18n-values: ' + attributeAndKeys);%0A%0A        var propName = attributeAndKeyPair[1];%0A        var propExpr = attributeAndKeyPair[2];%0A%0A        var value = data.getValue(propExpr);%0A%0A        // Allow a property of the form '.foo.bar' to assign a value into%0A        // element.foo.bar.%0A        if (propName[0] == '.') {%0A          var path = propName.slice(1).split('.');%0A          var targetObject = element;%0A          while (targetObject && path.length > 1) {%0A            targetObject = targetObject[path.shift()];%0A          }%0A          if (targetObject) {%0A            targetObject[path] = value;%0A            // In case we set innerHTML (ignoring others) we need to recursively%0A            // check the content.%0A            if (path == 'innerHTML') {%0A              for (var i = 0; i < element.children.length; ++i) {%0A                processWithoutCycles(element.children[i], data, visited, false);%0A              }%0A            }%0A          }%0A        } else {%0A          element.setAttribute(propName, /** @type {string} */ (value));%0A        }%0A      });%0A    }%0A  };%0A%0A  var prefixes = [''];%0A%0A  // Only look through shadow DOM when it's supported. As of April 2015, iOS%0A  // Chrome doesn't support shadow DOM.%0A  if (Element.prototype.createShadowRoot)%0A    prefixes.push('* /deep/ ');%0A%0A  var attributeNames = Object.keys(handlers);%0A  var selector = prefixes%0A                     .map(function(prefix) {%0A                       return prefix + '[' +%0A                           attributeNames.join('], ' + prefix + '[') + ']';%0A                     })%0A                     .join(', ');%0A%0A  /**%0A   * Processes a DOM tree using a |data| source to populate template values.%0A   * @param {!ProcessingRoot} root The root of the DOM tree to process.%0A   * @param {!LoadTimeData} data The data to draw from.%0A   */%0A  function process(root, data) {%0A    processWithoutCycles(root, data, new Set(), true);%0A  }%0A%0A  /**%0A   * Internal process() method that stops cycles while processing.%0A   * @param {!ProcessingRoot} root%0A   * @param {!LoadTimeData} data%0A   * @param {!Set<ProcessingRoot>} visited Already visited roots.%0A   * @param {boolean} mark Whether nodes should be marked processed.%0A   */%0A  function processWithoutCycles(root, data, visited, mark) {%0A    if (visited.has(root)) {%0A      // Found a cycle. Stop it.%0A      return;%0A    }%0A%0A    // Mark the node as visited before recursing.%0A    visited.add(root);%0A%0A    var importLinks = root.querySelectorAll('link[rel=import]');%0A    for (var i = 0; i < importLinks.length; ++i) {%0A      var importLink = /** @type {!HTMLLinkElement} */ (importLinks[i]);%0A      if (!importLink.import) {%0A        // Happens when a <link rel=import> is inside a <template>.%0A        // TODO(dbeam): should we log an error if we detect that here?%0A        continue;%0A      }%0A      processWithoutCycles(importLink.import, data, visited, mark);%0A    }%0A%0A    var templates = root.querySelectorAll('template');%0A    for (var i = 0; i < templates.length; ++i) {%0A      var template = /** @type {HTMLTemplateElement} */ (templates[i]);%0A      if (!template.content)%0A        continue;%0A      processWithoutCycles(template.content, data, visited, mark);%0A    }%0A%0A    var isElement = root instanceof Element;%0A    if (isElement && root.webkitMatchesSelector(selector))%0A      processElement(/** @type {!Element} */ (root), data, visited);%0A%0A    var elements = root.querySelectorAll(selector);%0A    for (var i = 0; i < elements.length; ++i) {%0A      processElement(elements[i], data, visited);%0A    }%0A%0A    if (mark) {%0A      var processed = isElement ? [root] : root.children;%0A      if (processed) {%0A        for (var i = 0; i < processed.length; ++i) {%0A          processed[i].setAttribute('i18n-processed', '');%0A        }%0A      }%0A    }%0A  }%0A%0A  /**%0A   * Run through various [i18n-*] attributes and populate.%0A   * @param {!Element} element%0A   * @param {!LoadTimeData} data%0A   * @param {!Set<ProcessingRoot>} visited%0A   */%0A  function processElement(element, data, visited) {%0A    for (var i = 0; i < attributeNames.length; i++) {%0A      var name = attributeNames[i];%0A      var attribute = element.getAttribute(name);%0A      if (attribute != null)%0A        handlers[name](element, attribute, data, visited);%0A    }%0A  }%0A%0A  return {process: process};%0A}());%0A%0A// // Copyright 2017 The Chromium Authors. All rights reserved.%0A// Use of this source code is governed by a BSD-style license that ca)”。

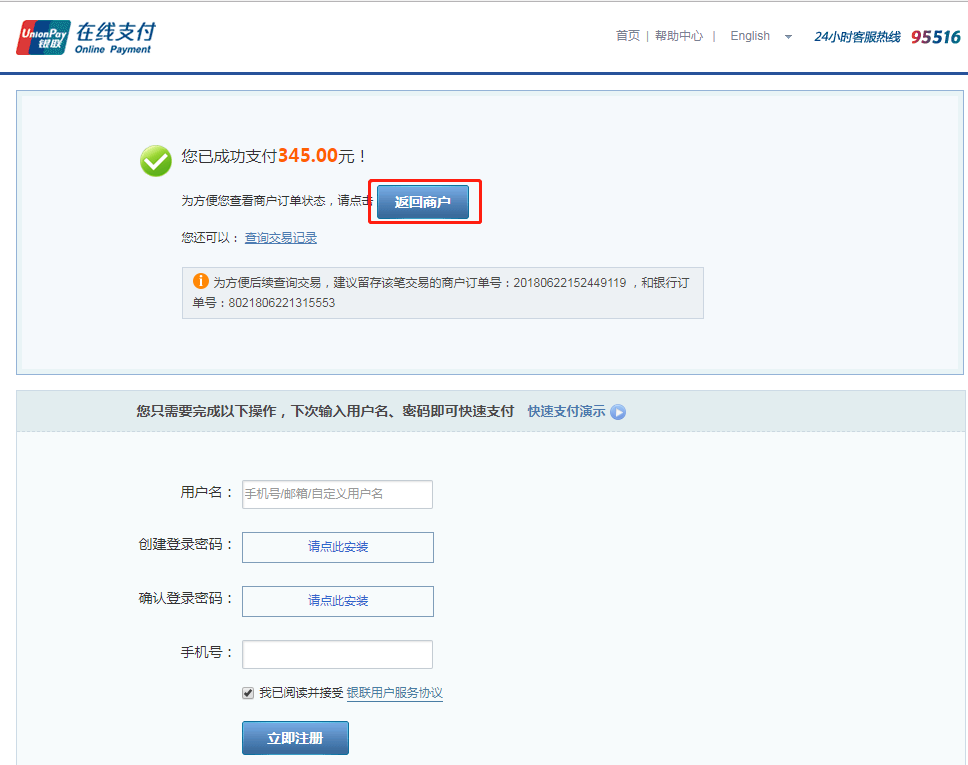




选择“中国农业银行”，再点击“到网上银行支付”。



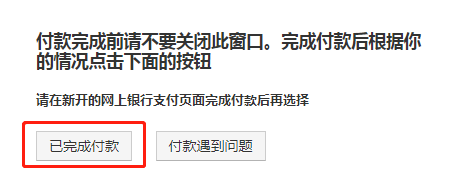
点击“返回商户”按钮。



显示当前支付单已经正常支付。



点击“已完成付款”按钮。



点击“已完成付款”按钮后，则显示该考生的缴费结果。

点击“关闭”按钮，则回到缴费页面，显示该考生分配的考位信息以及考试科目。

