Understanding CPU Usage with Web Inspector

Jonathan Davis, Web Technologies Evangelist
Content Blockers
Performance Best Practices
Performance Best Practices

Improve load time
Performance Best Practices

Improve load time

Optimize JavaScript
Performance Best Practices

Improve load time

Optimize JavaScript

Use CSS animations and transitions
CPU Usage Timeline
Recording CPU usage
Recording CPU usage
Finding power problems
Recording CPU usage
Finding power problems
Improving power efficiency
Smart Search Field: Show full website address

Accessibility:
- Never use font sizes smaller than [Dropdown]
- Press Tab to highlight each item on a webpage
  Option-Tab highlights each item.

Reading List: Save articles for offline reading automatically

Internet plug-ins: Stop plug-ins to save power

Style sheet: None Selected

Default encoding: Western (ISO Latin 1)

Proxies: Change Settings...
Smart Search Field: Show full website address

Accessibility:
- Never use font sizes smaller than [9]
- Press Tab to highlight each item on a webpage
  Option-Tab highlights each item.

Reading List: Save articles for offline reading automatically

Internet plug-ins:
- Stop plug-ins to save power

Style sheet: None Selected

Default encoding: Western (ISO Latin 1)

Proxies:
- Change Settings...

Show Develop menu in menu bar
Smart Search Field: □ Show full website address

Accessibility:
- □ Never use font sizes smaller than [9] □
- □ Press Tab to highlight each item on a webpage
  Option-Tab highlights each item.

Reading List:
- □ Save articles for offline reading automatically

Internet plug-ins:
- □ Stop plug-ins to save power

Style sheet: None Selected

Default encoding: Western (ISO Latin 1)

Proxies: Change Settings...

□ Show Develop menu in menu bar
A fast, open source web browser engine.

WebKit is the web browser engine used by Safari, Mail, App Store, and many other apps on macOS, iOS, and Linux. Get started contributing code, or reporting bugs.

Web developers can follow development, check feature status, download Safari Technology Preview to try out the latest web technologies, and report bugs.
A fast, open source web browser engine.

WebKit is the web browser engine used by Safari, Mail, App Store, and many other apps on macOS, iOS, and Linux. Get started contributing code, or reporting bugs.

Web developers can follow development, check feature status, download Safari Technology Preview to try out the latest web technologies, and report bugs.
A fast, open source web browser engine.

WebKit is the web browser engine used by Safari, Mail, App Store, and many other apps on macOS, iOS, and Linux. Get started contributing code, or reporting bugs.

Web developers can follow development, check feature status, download Safari Technology Preview to try out the latest web technologies, and report bugs.
WebKit is the web browser engine used by Safari, Mail, App Store, and many other apps on macOS, iOS, and Linux. Get started contributing code, or reporting bugs.

Web developers can follow development, check feature status, download Safari Technology Preview to try out the latest web technologies, and report bugs.
WebKit is the web browser engine used by Safari, Mail, App Store, and many other apps on macOS, iOS, and Linux. Get started contributing code, or reporting bugs.

Web developers can follow development, check feature status, download Safari Technology Preview to try out the latest web technologies, and report bugs.
WebKit Feature Status

Application Cache

Array.prototype.copyWithin 

Array.prototype.includes 

Supported

Supported

Deprecated
<table>
<thead>
<tr>
<th>Feature</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Cache</td>
<td>Deprecated</td>
</tr>
<tr>
<td>Array.prototype.copyWithin es5</td>
<td>Supported</td>
</tr>
<tr>
<td>Array.prototype.includes es7</td>
<td>Supported</td>
</tr>
</tbody>
</table>
function stageImage(element, src) {
  element.style.backgroundImage = 'url(' + src + ')';
  if (!element.parentElement.classList.contains('loaded'))
    element.parentElement.classList.add('loaded');
}

function loadImage(element) {
  var src = element.getAttribute('data-url');
  if (sessionStorage.getItem(src)) {
    setTimeout(function () { stageImage(element, src); }, 1);
  } else {
    var img = new Image();
    img.src = src;
    img.onload = function () {
      stageImage(element, src);
      try {
        sessionStorage.setItem(src, true);
      } catch (error) {
        return false; // private browsing
      }
      img = undefined;
    }
  }
}

function onMovement() {
  if (!updating)
    requestAnimationFrame(updateImages);
  updating = true;
}

function updateImages() {
  updating = false;
  for (var i = 0; i < imgs.length; i++) {
    if (InView(imgs[i]))
      loadImage(imgs[i]);
  }
}

function enableScrollableTables() {
  var tables = document.querySelectorAll('table');
  var tableCount = tables.length;
  for (var i = 0; i < tableCount; i++) {
    var scrollableDiv = document.createElement('div');
    var paddingDiv = document.createElement('div');
    scrollableDiv.classList.add('scrollable');
    paddingDiv.classList.add('scrollable-padding');
    scrollableDiv.appendChild(paddingDiv);
    tables[i].parentNode.insertBefore(scrollableDiv, tables[i]);
    paddingDiv.appendChild(tables[i]);
  }
  var imgs = document.querySelectorAll('div[data-url]');
function stageImage(element, src) {
  element.style.backgroundImage = 'url(' + src + ')';
  if (element.parentElement.classList.contains('loaded'))
    element.parentElement.classList.add('loaded');
}

function loadImage(element) {
  var src = element.getAttribute('data-url');

  if (sessionStorage.getItem(src)) {
    setTimeout(function () { stageImage(element, src); }, 1);
  } else {
    var img = new Image();
    img.src = src;
    img.onload = function() {
      stageImage(element, src);
      try {
        sessionStorage.setItem(src, true);
      } catch (error) {
        return false; // private browsing
      }
    }
    img = undefined;
  }
}

function onMovement() {
  if (!updating)
    requestAnimationFrame(updateImages);
  updating = true;
}

function updateImages() {
  updating = false;
  for (var i = 0; i < imgs.length; i++) {
    if (InView(imgs[i]))
      loadImage(imgs[i]);
  }
}

function enableScrollableTables() {
  var tables = document.querySelectorAll(':not(table)');
  var tableCount = tables.length;

  for (var i = 0; i < tableCount; i++) {
    var scrollableDiv = document.createElement('div');
    var paddingDiv = document.createElement('div');
    scrollableDiv.classList.add('scrollable');
    paddingDiv.classList.add('scrollable-padding');
    scrollableDiv.appendChild(paddingDiv);
    tables[i].parentNode.insertBefore(scrollableDiv, tables[i]);
    paddingDiv.appendChild(tables[i]);
  }
}

var imgs = document.querySelectorAll('div[data-url]');
function stageImage(element, src) {
  element.style.backgroundColor = 'url(' + src + ')';
  if (element.parentElement.classList.contains('loaded'))
    element.parentElement.classList.add('loaded');
}

function loadImage(element) {
  var src = element.getAttribute('data-url');

  if (sessionStorage.getItem(src)) {
    setTimeout(function () { stageImage(element, src); }, 1);
  } else {
    var img = new Image();
    img.src = src;
    img.onload = function() {
      stageImage(element, src);
      try {
        sessionStorage.setItem(src, true);
      } catch (error) {
        return false; // private browsing
      }
    }
  }
}

function onMovement() {
  if (!updating)
    requestAnimationFrame(updateImages);
  updating = true;
}

function updateImages() {
  updating = false;
  for (var i = 0; i < imgs.length; i++) {
    if (InView(imgs[i]))
      loadImage(imgs[i]);
  }
}

function enableScrollableTables() {
  var tables = document.querySelectorAll('.bodycopy > table');
  var tableCount = tables.length;

  for (var i = 0; i < tableCount; i++) {
    var scrollableDiv = document.createElement('div');
    var paddingDiv = document.createElement('div');
    scrollableDiv.classList.add('scrollable');
    paddingDiv.classList.add('scrollable-padding');
    scrollableDiv.appendChild(paddingDiv);
    tables[i].parentNode.insertBefore(scrollableDiv, tables[i]);
    paddingDiv.appendChild(tables[i]);
  }

  var imgs = document.querySelectorAll('div[data-url]');
function stageImage(element, src) {
  element.style.backgroundImage = 'url(' + src + ')';
  if (!element.parentElement.classList.contains('loaded'))
    element.parentElement.classList.add('loaded');
}

function loadImage(element) {
  var src = element.getAttribute('data-url');
  if (sessionStorage.getItem(src)) {
    setTimeout(function() { stageImage(element, src); }, 1);
  } else {
    var img = new Image();
    img.src = src;
    img.onload = function() {
      stageImage(element, src);
      try {
        sessionStorage.setItem(src, true);
      } catch (error) {
        return false; // private browsing
      }
    }
    img = undefined;
  }
}

function onMovement() {
  if (!updating)
    requestAnimationFrame(updateImages);
  updating = true;
}

function updateImages() {
  updating = false;
  for (var i = 0; i < imgs.length; i++) {
    if (InView(imgs[i]))
      loadImage(imgs[i]);
  }
}

function enableScrollableTables() {
  var tables = document.querySelectorAll('.bodycopy > table');
  var tableCount = tables.length;
  for (var i = 0; i < tableCount; i++) {
    var scrollableDiv = document.createElement('div');
    var paddingDiv = document.createElement('div');
    scrollableDiv.classList.add('scrollable');
    paddingDiv.classList.add('scrollable-padding');
    scrollableDiv.appendChild(paddingDiv);
    tables[i].parentNode.insertBefore(scrollableDiv, tables[i]);
    paddingDiv.appendChild(tables[i]);
  }
  var imgs = document.querySelectorAll('div[data-url]');
<table>
<thead>
<tr>
<th>Feature</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Cache</td>
<td>Deprecated</td>
</tr>
<tr>
<td>Array.prototype.copyWithin</td>
<td>Supported</td>
</tr>
<tr>
<td>Array.prototype.includes</td>
<td>Supported</td>
</tr>
<tr>
<td>Arrow Functions</td>
<td>Supported</td>
</tr>
<tr>
<td>Async Iteration</td>
<td>Supported</td>
</tr>
<tr>
<td>async/await</td>
<td>Supported</td>
</tr>
<tr>
<td>Battery Status API</td>
<td>Removed</td>
</tr>
<tr>
<td>Beacon API</td>
<td>Supported In Preview</td>
</tr>
<tr>
<td>Block Scoping</td>
<td>Supported</td>
</tr>
<tr>
<td>Case-insensitive attribute selectors</td>
<td>Supported</td>
</tr>
<tr>
<td>CSS Selectors Level 4</td>
<td>Supported</td>
</tr>
<tr>
<td>Classes</td>
<td>Supported</td>
</tr>
<tr>
<td>Conic Gradients</td>
<td>Supported In Preview</td>
</tr>
<tr>
<td>Feature</td>
<td>Status</td>
</tr>
<tr>
<td>------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Application Cache</td>
<td>Deprecated</td>
</tr>
<tr>
<td>Array.prototype.copyWithin</td>
<td>Supported</td>
</tr>
<tr>
<td>Array.prototype.includes</td>
<td>Supported</td>
</tr>
<tr>
<td>Arrow Functions</td>
<td>Supported</td>
</tr>
<tr>
<td>Async Iteration</td>
<td>Supported</td>
</tr>
<tr>
<td>async/await</td>
<td>Supported</td>
</tr>
<tr>
<td>Battery Status API</td>
<td>Removed</td>
</tr>
<tr>
<td>Beacon API</td>
<td>Supported In Preview</td>
</tr>
<tr>
<td>Block Scoping</td>
<td>Supported</td>
</tr>
<tr>
<td>Case-insensitive attribute selectors</td>
<td>Supported</td>
</tr>
<tr>
<td>CSS Selectors Level 4</td>
<td>Supported</td>
</tr>
<tr>
<td>Classes</td>
<td>Supported</td>
</tr>
<tr>
<td>Conic Gradients</td>
<td>Supported In Preview</td>
</tr>
</tbody>
</table>
var src = document.getElementById("data-url");

if (sessionStorage.getItem(src)) {
  setTimeout(function () { stageImage(element, src); }, 1);
} else {
  var img = new Image();
  img.src = src;
  img.onload = function () {
    stageImage(element, src);
    try {
      sessionStorage.setItem(src, true);
    } catch (error) {
      return false; // private browsing
    }
    img = undefined;
  }
}

function onMovement() {
  if (updating)
    requestAnimationFrame(updateImages);
  updating = true;
}

function updateImages() {
  updating = false;
  for (var i = 0; i < imgs.length; i++) {
    if (imgs[i])
      loadImage(imgs[i]);
  }
}

function enableScrollableTables() {
  var tables = document.querySelectorAll('table')
  var tableCount = tables.length;
  for (var i = 0; i < tableCount; i++) {
    var scrollTableDiv = document.createElement('div');
    var paddingDiv = document.createElement('div');
    scrollTableDiv.classList.add('scrollable');
    paddingDiv.classList.add('scrollable-padding');
    scrollTableDiv.appendChild(paddingDiv);
    tables[i].parentNode.insertBefore(scrollTableDiv, tables[i]);
    paddingDiv.appendChild(tables[i]);
  }
  var imgs = document.querySelectorAll('div[data-url]');
  document.addEventListener('scroll', onMovement);
  document.addEventListener('resize', onMovement);
  updateImages();
  enableScrollableTables();
}
```javascript
var src = document.getElementById('data-url');
if (sessionStorage.getItem(src)) {
  setTimeout(function() { stageImage(element, src); }, 1);
} else {
  var img = new Image();
  img.src = src;
  img.onload = function() {
    stageImage(element, src);
    try {
      sessionStorage.setItem(src, true);
    } catch (error) {
      return false; // private browsing
    }
    img = undefined;
  }
}

function onMovement() {
  if (updating) requestAnimationFrame(updateImages);
  updating = true;
}

function updateImages() {
  updating = false;
  for (var i = 0; i < imgs.length; i++) {
    if (inView(imgs[i])) loadImage(imgs[i]);
  }
}

function enableScrollableTables() {
  var tables = document.querySelectorAll('table');
  var tableCount = tables.length;
  for (var i = 0; i < tableCount; i++) {
    var scrollTableDiv = document.createElement('div');
    var paddingDiv = document.createElement('div');
    scrollTableDiv.classList.add('scrollable');
    paddingDiv.classList.add('padding');
    scrollTableDiv.appendChild(paddingDiv);
    paddingDiv.appendChild(tables[i]);
    tables[i].parentNode.insertBefore(scrollTableDiv, tables[i]);
  }
}

var imgs = document.querySelectorAll('div[data-url]');
document.addEventListener('scroll', onMovement);
document.addEventListener('resize', onMovement);
updateImages();
enableScrollableTables();
```
// Lazy Loading Images

var imgs = document.querySelectorAll('div[data-url]');
document.addEventListener('scroll', onMovement);
document.addEventListener('resize', onMovement);
// Lazy Loading Images with a conditional guard

var imgs = document.querySelectorAll('div[data-url]');
if (imgs.length > 0) {
    document.addEventListener('scroll', onMovement);
    document.addEventListener('resize', onMovement);
}
// Lazy Loading Images with a conditional guard

var imgs = document.querySelectorAll('div[data-url]');
if (imgs.length > 0) {
    document.addEventListener('scroll', onMovement);
    document.addEventListener('resize', onMovement);
}
WebKit Feature Status

145 features

Application Cache

Array.prototype.copyWithin

Array.prototype.includes

Arrow Functions

Async Functions

Deprecated

Supported
Introducing Custom Elements
A little over a year ago, we announced the introduction of skitt-based shadow DOM API, a lightweight mechanism to encapsulate a DOM tree by allowing the creation of a parallel DOM tree on an element called a "shadow tree" that replaces the rendering of the element without

Release Notes for Safari Technology Preview 18
Input events on by default, <link> style sheets loaded in Shadow DOM, Web Worker Debugging and more...
Read more >

Release Notes for Safari Technology Preview 17
Safari Technology Preview Release 17 is now available for download for macOS Sierra.
Read more >
MotionMark: A New Graphics Benchmark

Today, we are pleased to introduce MotionMark, a new graphics benchmark for web browsers.

Read more >

Release Notes for Safari Technology Preview 13

Safari Technology Preview Release 13 is now available for download for both macOS Sierra betas and OS X El Capitan 10.11.6.

Read more >

Release Notes for Safari Technology Preview 12

Safari Technology Preview Release 12 is now available for download for both macOS Sierra betas and OS X El Capitan.

Read more >
Variable Fonts on the Web
Variable fonts can expose any number of axes, each controlling a different aspect of the font.

Read more »

Release Notes for Safari Technology Preview 16
Safari Technology Preview Release 16 is now available for download for macOS Sierra.

Read more »

iOS 10 Link Preview API in WKWebView
Multi-Touch gestures have always been an integral part of the iPhone experience, but pinch and double tap were just the beginning of the story.

Read more »
Variable Fonts on the Web

Variable fonts can expose any number of axes, each controlling a different aspect of the font.

Read more »

Release Notes for Safari Technology Preview 16

Safari Technology Preview Release 16 is now available for download for macOS Sierra.

Read more »

iOS 10 Link Preview API in WKWebView

Multi-Touch gestures have always been an integral part of the iPhone experience, but pinch and double tap were just the beginning of the story.

Read more »
Power Saving Tips for Web Content
Power Saving Tips for Web Content

Investigate page interactions and idle
Power Saving Tips for Web Content

Investigate page interactions and idle

Reduce CPU usage to save power
Power Saving Tips for Web Content

Investigate page interactions and idle

Reduce CPU usage to save power

Use CSS animations and transitions
Power Saving Tips for Web Content

Investigate page interactions and idle
Reduce CPU usage to save power
Use CSS animations and transitions
Avoid scroll events, use IntersectionObserver
CPU Usage Timeline
Auditing Web Content with Web Inspector

Jonathan Davis, Web Technologies Evangelist
WebDriver
WebDriver
Audit
Using Audits
Using Audits

Working with Tests
Using Audits

Working with Tests

Creating Custom Audits
Smart Search Field: ☑ Show full website address

Accessibility:
- ☐ Never use font sizes smaller than 9
- ☐ Press Tab to highlight each item on a webpage
- Option-Tab highlights each item.

Reading List: ☑ Save articles for offline reading automatically

Internet plug-ins:
- ☑ Stop plug-ins to save power

Style sheet: None Selected

Default encoding: Western (ISO Latin 1)

Proxies: Change Settings...

☐ Show Develop menu in menu bar
Smart Search Field:  
- Show full website address

Accessibility:
- Never use font sizes smaller than 9
- Press Tab to highlight each item on a webpage
  Option-Tab highlights each item.

Reading List:  
- Save articles for offline reading automatically

Internet plug-ins:  
- Stop plug-ins to save power

Style sheet: None Selected

Default encoding: Western (ISO Latin 1)

Proxies: Change Settings...

Show Develop menu in menu bar
WebKit

A fast, open source web browser engine.

WebKit is the web browser engine used by Safari, Mail, App Store, and many other apps on macOS, iOS, and Linux. Get started contributing code, or reporting bugs.

Web developers can follow development, check feature status, download Safari Technology Preview to try out the latest web technologies, and report bugs.
No audit selected

Press `import` to import a test or result file
No audit selected

Press  Import to import a test or result file
No audit selected

Press Import to import a test or result file
No audit selected

Press Import to import a test or result file
No audit selected

Press 🔄 Import to import a test or result file
No audit selected

Press Import to import a test or result file.
No audit selected

Press `import` to import a test or result file
Demo Audit

These tests serve as a demonstration of the functionality and structure of audits.

Result Levels

These are all of the different test result levels.

- **level-pass**
  This is the result of a passing test with no data looks like.

- **level-warn**
  This is the result of a warning test with no data looks like.

- **level-fail**
  This is the result of a failing test with no data looks like.

- **level-error**
  This is the result of a test that threw an error with no data looks like.

- **level-unsupported**
  This is the result of an unsupported test with no data looks like.

Result Data

These are all of the different types of data that can be returned with the test result.

- **data-domNodes**
  This is an example of how DOM nodes are shown. It will pass with the `<body>` element.

  DOM Nodes:
  ```html
  <body class="home page-template-default page page-id-4118"/>
  ```

- **data-domAttributes**
  This is an example of how result DOM nodes are shown. It will pass with all elements with an id attribute.

  DOM Nodes:
  ```html
  <link rel="stylesheet" id="wp-block-library-css" href="https://webkit.org/wp-includes/css/dist-block-library/style.min.css?ver=5.1.1" type="text/css" media="all"/>
  ```
These are all of the different test result levels.

- **level-pass**
  This is what the result of a passing test with no data looks like.

- **level-warn**
  This is what the result of a warning test with no data looks like.

- **level-fail**
  This is what the result of a failing test with no data looks like.

- **level-error**
  This is what the result of a test that threw an error with no data looks like.

- **level-unsupported**
  This is what the result of an unsupported test with no data looks like.

### Result Data
These are all of the different types of data that can be returned with the test result.

- **data-domNodes**
  This is an example of how result DOM nodes are shown. It will pass with the `<body>` element.

  **DOM Nodes:**
  ```html
  <body class="home page-template-default page page-id-4118">...
  </body>
  ```

- **data-domAttributes**
  This is an example of how result DOM nodes are shown. It will pass with all elements with an id attribute.

  **DOM Nodes:**
  ```html
  <link rel="stylesheet" id="wp-block-library-css" href="https://webkit.org/wp/wp-includes/css/dist/block-library/styles.css"
  media="all">
  ```
  ```html
  .wp-block-highlighter wi("wi") style="blockquotes indentation-4" @media="all"
  ```
These are all of the different test result levels.

- **level-pass**
  This is what the result of a passing test with no data looks like.

- **level-warn**
  This is what the result of a warning test with no data looks like.

- **level-fail**
  This is what the result of a failing test with no data looks like.

- **level-error**
  This is what the result of a test that threw an error with no data looks like.

- **level-unsupported**
  This is what the result of an unsupported test with no data looks like.

**Result Data**
These are all of the different types of data that can be returned with the test result.

- **data-domNodes**
  This is an example of how result DOM nodes are shown. It will pass with the `<body>` element.
  
  DOM Nodes:
  ```
  1 <body class="home page-template-default page page-id-4118">...
  ```

- **data-domAttributes**
  This is an example of how result DOM nodes are shown. It will pass with all elements with an id attribute.
  
  DOM Nodes:
  ```
  1 <link rel="stylesheet" id="wp-block-library-css" href="https://webkit.org/wp-includes/css/dist/block-library/styles.css?media=all">
  ```
Editing audits

Press **Done** to stop editing
Editing audits

Press Done to stop editing
Editing audits

Press Done to stop editing
Editing audits

Press Done to stop editing
Demo Audit

Result Levels
- level-pass
- level-warn
- level-fail
- level-error
- level-unsupported

Result Data
- data-domNodes
- data-domAttributes
- data-errors

Accessibility

Editing audits

Press Done to stop editing
Editing audits

Press Done to stop editing
Editing audits

Press Done to stop editing

Audit version: 2
Editing audits

Press Done to stop editing
Demo Audit

These tests serve as a demonstration of the functionality and structure of audits.

Result Levels
These are all of the different test result levels.

- **level-pass**: This is what the result of a passing test with no data looks like.
  5/20/2019, 8:34:50 PM  4.00ms

- **level-warn**: This is what the result of a warning test with no data looks like.
  5/20/2019, 8:34:50 PM  2.00ms

- **level-fail**: This is what the result of a failing test with no data looks like.
  5/20/2019, 8:34:50 PM  2.00ms

- **level-error**: This is what the result of a test that threw an error with no data looks like.
  5/20/2019, 8:34:50 PM  2.00ms

- **level-unsupported**: This is what the result of an unsupported test with no data looks like.
  5/20/2019, 8:34:50 PM  15.00ms

Result Data
These are all of the different types of data that can be returned with the test result.

- **data-domNodes**: This is an example of how result DOM nodes are shown. It will pass with the `<body>` element.
  5/20/2019, 8:34:50 PM  15.00ms

  DOM Nodes:
  1. `<body class="home page-template-default page-id-4118"><\/body>`

- **data-domAttributes**: This is an example of how result DOM nodes are shown. It will pass with all elements with an id attribute.
  5/20/2019, 8:34:50 PM  3.00ms

  DOM Nodes:
  1. `<link rel="stylesheet" id="wp-block-library-css" href="https://webkit.org/wp-includes/css/dist-block-library/style.min.css?ver=5.1.1" type="text/css" media="all"`
No audit selected

Press Import to import a test or result file
No audit selected

Press import to import a test or result file
No audit selected

Press Import to import a test or result file

Audit version: 2
Demo Audit

These tests serve as a demonstration of the functionality and structure of audits.

Result Levels

These are all of the different test result levels.

- **level-pass**
  This is what the result of a passing test with no data looks like.

- **level-warn**
  This is what the result of a warning test with no data looks like.

- **level-fail**
  This is what the result of a failing test with no data looks like.

- **level-error**
  This is what the result of a test that threw an error with no data looks like.

- **level-unsupported**
  This is what the result of an unsupported test with no data looks like.

Result Data

These are all of the different types of data that can be returned with the test result.

- **data-domNodes**
  This is an example of how result DOM nodes are shown. It will pass with the `<body>` element.

  DOM Nodes:
  ```html
  <body class="home page-template-default page page-id-4118">...
  </body>
  ```

- **data-domAttributes**
  This is an example of how result DOM nodes are shown. It will pass with all elements with an id attribute.

  DOM Nodes:
  ```html
  <link rel="stylesheet" id="wp-block-library-css" href="https://webkit.org/wp/wp-includes/css/dist/block-library-style.min.css?ver=5.11" type="text/css" media="all">
  ```
Passed | Warning | Failed | Error | Unsupported
Demo Audit

These tests serve as a demonstration of the functionality and structure of audits.

Result Levels

These are all of the different test result levels.

- **level-pass**
  This is what the result of a passing test with no data looks like.

- **level-warn**
  This is what the result of a warning test with no data looks like.

- **level-fail**
  This is what the result of a failing test with no data looks like.

- **level-error**
  This is what the result of a test that threw an error with no data looks like.

- **level-unsupported**
  This is what the result of an unsupported test with no data looks like.

Result Data

These are all of the different types of data that can be returned with the test result.

- **data-domNodes**
  This is an example of how result DOM nodes are shown. It will pass with the `<body>` element.

  **DOM Nodes:**
  ```html
  <body class="home page-template-default page page-id-4118">...
  ```
Demo Audit

These tests serve as a demonstration of the functionality and structure of audits.

Result Levels

These are all of the different test result levels.

- **level-pass**
  This is what the result of a passing test with no data looks like.

- **level-warn**
  This is what the result of a warning test with no data looks like.

- **level-fail**
  This is what the result of a failing test with no data looks like.

- **level-error**
  This is what the result of a test that threw an error with no data looks like.

- **level-unsupported**
  This is what the result of an unsupported test with no data looks like.

Result Data

These are all of the different types of data that can be returned with the test result.

- **data-domNodes**
  This is an example of how result DOM nodes are shown. It will pass with the `<body>` element.

DOM Nodes:

```
<body class="home page-template-default page page-id-4118"...></body>
```
Demo Audit

These tests serve as a demonstration of the functionality and structure of audits.

No Result

Press [Start] to start running the audit
Demo Audit

These tests serve as a demonstration of the functionality and structure of audits.
Accessibility

Diagnoses common accessibility problems affecting screen readers and other assistive technology.

No Result

Press [Start] to start running the audit.
Accessibility

Diagnoses common accessibility problems affecting screen readers and other assistive technology.

No Result

Press [Start] to start running the audit
Accessibility

Diagnoses common accessibility problems affecting screen readers and other assistive technology.

No Result

Press Start to start running the audit.
Accessibility
Diagnoses common accessibility problems affecting screen readers and other assistive technology.

- **testMenuRoleForRequiredChildren**: Ensure that element of role "menu" and "menubar" have required owned elements in accordance with WAI-ARIA.
  - DOM Nodes:
    ```html
    <ul id="menu-main-menu" class="menu" role="menubar">
    </ul>
    ```
  - 5/30/2019, 10:31:27 AM
  - 6.00ms

- **testGridRoleForRequiredChildren**: Ensure that elements of role "grid" have required owned elements in accordance with WAI-ARIA.
  - 5/30/2019, 10:31:27 AM
  - 2.00ms

- **testForAriaLabelledBySpelling**: Ensure that "aria-labelledby" is spelled correctly.
  - 5/30/2019, 10:31:27 AM
  - 2.00ms

- **testForMultipleBanners**: Ensure that only one banner is used on the page.
  - 5/30/2019, 10:31:27 AM
  - 26.00ms

- **testForLinkLabels**: Ensure that links have accessible labels for assistive technology.
  - 5/30/2019, 10:31:27 AM
  - 19.00ms

- **testRowGroupRoleForRequiredChildren**: Ensure that element of role "rowgroup" have required owned elements in accordance with WAI-ARIA.
  - 5/30/2019, 10:31:27 AM
  - 3.00ms

- **testTableRoleForRequiredChildren**: Ensure that elements of role "table" have required owned elements in accordance with WAI-ARIA.
  - 5/30/2019, 10:31:27 AM
  - 24.00ms

- **testForMultipleLiveRegions**: Ensure that only one live region is used on the page.
  - 5/30/2019, 10:31:27 AM
  - 20.00ms

- **testListBoxRoleForRequiredChildren**: Ensure that elements of role "listbox" have required owned elements in accordance with WAI-ARIA.
  - 5/30/2019, 10:31:27 AM
  - 2.00ms

- **testImageLabels**: Ensure that elements of role "img" have accessible labels for assistive technology.
  - 5/30/2019, 10:31:27 AM
  - 20.00ms
Accessibility

Diagnoses common accessibility problems affecting screen readers and other assistive technology.

- **testMenuRoleForRequiredChildren**
  - Ensure that element of role "menu" and "menubar" have required owned elements in accordance with WAIA-ARIA.
  - DOM Nodes:
    - `<ol id="main-menu" class="menu" role="menubar">`</ol>
  - 5/30/2018, 10:31:27 AM
  - 6.00ms

- **testGridRoleForRequiredChildren**
  - Ensure that elements of role "grid" have required owned elements in accordance with WAIA-ARIA.
  - 5/30/2018, 10:31:27 AM
  - 2.00ms

- **testForAriaLabelledBySpelling**
  - Ensure that "aria-labelledby" is spelled correctly.
  - 5/30/2018, 10:31:27 AM
  - 2.00ms

- **testForMultipleBanners**
  - Ensure that only one banner is used on the page.
  - 5/30/2018, 10:31:27 AM
  - 26.00ms

- **testForLinkLabels**
  - Ensure that links have accessible labels for assistive technology.
  - 5/30/2018, 10:31:27 AM
  - 19.00ms

- **testRowGroupRoleForRequiredChildren**
  - Ensure that element of role "rowgroup" have required owned elements in accordance with WAIA-ARIA.
  - 5/30/2018, 10:31:27 AM
  - 3.00ms

- **testTableRoleForRequiredChildren**
  - Ensure that elements of role "table" have required owned elements in accordance with WAIA-ARIA.
  - 5/30/2018, 10:31:27 AM
  - 24.00ms

- **testForMultipleLiveRegions**
  - Ensure that only one live region is used on the page.
  - 5/30/2018, 10:31:27 AM
  - 20.00ms

- **testListBoxRoleForRequiredChildren**
  - Ensure that elements of role "listbox" have required owned elements in accordance with WAIA-ARIA.
  - 5/30/2018, 10:31:27 AM
  - 2.00ms

- **testImageLabels**
  - Ensure that elements of role "img" have accessible labels for assistive technology.
  - 5/30/2018, 10:31:27 AM
  - 20.00ms
WebKit

A fast, open source web browser engine.

WebKit is the web browser engine used by Safari, Mail, App Store, and many other apps on macOS, iOS, and Linux. Get started contributing code, or reporting bugs.

Web developers can follow development, check feature status, download Safari Technology Preview to try out the latest web technologies, and report bugs.

Accessibility

Diagnoses common accessibility problems affecting screen readers and other assistive technology.

testMenuRoleForRequiredChildren

Ensure that elements of role "menu" and "menubar" have required owned elements in accordance with WAI-ARIA.

DOQ Nodes:

5/30/2018, 10:49:54 AM 5.00ms

5/30/2018, 10:49:54 AM 2.00ms

5/30/2018, 10:49:54 AM 2.00ms

5/30/2018, 10:49:54 AM 5.00ms
A fast, open source web browser engine.

WebKit is the web browser engine used by Safari, Mail, App Store, and many other apps on macOS, iOS, and Linux. Get started contributing code, or reporting bugs.

Web developers can follow development, check feature status, download Safari Technology Preview to try out the latest web technologies, and report bugs.

Accessibility

Diagnoses common accessibility problems affecting screen readers and other assistive technology.

testMenuRoleForRequiredChildren

Ensure that element of role "menu" and "menubar" have required owned elements in accordance with WA-ARIA.

DOQ Nodes:

5/30/2018, 10:49:54 AM  5.00ms

<ol>
  <li>
    <button class="menu-item menu-item-has-children">
      <a role="menuitem" href="#">
        <span class="menu-item-title">Item 1</span>
        <span class="menu-item-caret">▼</span>
      </a>
    </button>
  </li>
  <li>
    <button class="menu-item menu-item-has-children">
      <a role="menuitem" href="#">
        <span class="menu-item-title">Item 2</span>
        <span class="menu-item-caret">▼</span>
      </a>
    </button>
  </li>
  <li>
    <button class="menu-item menu-item-has-children">
      <a role="menuitem" href="#">
        <span class="menu-item-title">Item 3</span>
        <span class="menu-item-caret">▼</span>
      </a>
    </button>
  </li>
</ol>

5/30/2018, 10:49:54 AM  2.00ms

testGridRoleForRequiredChildren

Ensure that elements of role "grid" have required owned elements in accordance with WA-ARIA.

5/30/2018, 10:49:54 AM  2.00ms

testForAriaLabelledBySpelling

Ensures that "aria-labeledby" is spelled correctly.
Accessibility
Diagnoses common accessibility problems affecting screen readers and other assistive technology.
Accessibility

Diagnoses common accessibility problems affecting screen readers and other assistive technology.

- **testMenuRoleForRequiredChildren**
  Ensure that element of role "menu" and "menubar" have required owned elements in accordance with WAI-ARIA.
  
  5/30/2019, 10:51:04 AM 46.0ms

- **testGridRoleForRequiredChildren**
  Ensure that elements of role "grid" have required owned elements in accordance with WAI-ARIA.
  
  5/30/2019, 10:51:04 AM 2.00ms

- **testForAriaLabelledBySpelling**
  Ensure that "aria-labelledby" is spelled correctly.
  
  5/30/2019, 10:51:04 AM 3.00ms

- **testForMultipleBanners**
  Ensure that only one banner is used on the page.
  
  5/30/2019, 10:51:04 AM 27.0ms

- **testForLinkLabels**
  Ensure that links have accessible labels for assistive technology.
  
  5/30/2019, 10:51:04 AM 31.0ms

- **testRowGroupRoleForRequiredChildren**
  Ensure that element of role "rowgroup" have required owned elements in accordance with WAI-ARIA.
  
  5/30/2019, 10:51:04 AM 30.0ms

- **testTableRoleForRequiredChildren**
  Ensure that elements of role "table" have required owned elements in accordance with WAI-ARIA.
  
  5/30/2019, 10:51:04 AM 2.00ms
Accessibility
Diagnoses common accessibility problems affecting screen readers and other assistive technology.

- **testMenuRoleForRequiredChildren**
  Ensure that element of role "menu" and "menubar" have required owned elements in accordance with WAI-ARIA.

- **testGridRoleForRequiredChildren**
  Ensure that elements of role "grid" have required owned elements in accordance with WAI-ARIA.

- **testForAriaLabelledBySpelling**
  Ensure that "aria-labelledby" is spelled correctly.

- **testForMultipleBanners**
  Ensure that only one banner is used on the page.

- **testForLinkLabels**
  Ensure that links have accessible labels for assistive technology.

- **testRowGroupRoleForRequiredChildren**
  Ensure that element of role "rowgroup" have required owned elements in accordance with WAI-ARIA.

- **testTableRoleForRequiredChildren**
  Ensure that elements of role "table" have required owned elements in accordance with WAI-ARIA.
Accessibility

Diagnoses common accessibility problems affecting screen readers and other assistive technology.

- **testMenuRoleForRequiredChildren**
  Ensure that element of role "menu" and "menubar" have required owned elements in accordance with WAI-ARIA.

- **testGridRoleForRequiredChildren**
  Ensure that elements of role "grid" have required owned elements in accordance with WAI-ARIA.

- **testForAriaLabelledBySpelling**
  Ensure that "aria-labelledby" is spelled correctly.

- **testForMultipleBanners**
  Ensure that only one banner is used on the page.

- **testForLinkLabels**
  Ensure that links have accessible labels for assistive technology.

- **testRowGroupRoleForRequiredChildren**
  Ensure that element of role "rowgroup" have required owned elements in accordance with WAI-ARIA.

- **testTableRoleForRequiredChildren**
  Ensure that elements of role "table" have required owned elements in accordance with WAI-ARIA.
Accessibility

Diagnoses common accessibility problems affecting screen readers and other assistive technology.

- **testMenuRoleForRequiredChildren**
  Ensure that element of role "menu" and "menubar" have required owned elements in accordance with WAI-ARIA.

- **testGridRoleForRequiredChildren**
  Ensure that elements of role "grid" have required owned elements in accordance with WAI-ARIA.

- **testForAriaLabelledBySpelling**
  Ensure that "aria-labelledby" is spelled correctly.

- **testForMultipleBanners**
  Ensure that only one banner is used on the page.

- **testForLinkLabels**
  Ensure that links have accessible labels for assistive technology.

- **testRowGroupRoleForRequiredChildren**
  Ensure that element of role "rowgroup" have required owned elements in accordance with WAI-ARIA.

- **testTableRoleForRequiredChildren**
  Ensure that elements of role "table" have required owned elements in accordance with WAI-ARIA.
// Example Audit JSON Format

{
  "type": "test-group",
  "name": "Test Group Name",
  "description": "Description of the test group",
  "tests": [
    {
      "type": "test-case",
      "name": "testName",
      "description": "Your description of the test.",
      "supports": 1,
      "test": "function () { return {level: 'pass'}; }"
    },
    ...
  ]
}
// Example Audit JSON Format

{
    "type": "test-group",
    "name": "Test Group Name",
    "description": "Description of the test group",
    "tests": [
        {
            "type": "test-case",
            "name": "testName",
            "description": "Your description of the test."
        },
        ...
    ]
}
Creating Web Inspector Audits

This post is a followup to the Audits in Web Inspector post, and explains the capabilities of and how to write an Audit.

Test Case Format

The actual test that is run in the page is just a string/file JavaScript function. async functions are allowed, as well as non-async functions that return a Promise. The only requirement is that the return value of the function (or Promise) conforms to the following rules:

- Returning true/false will translate to a Pass/Fail result with no additional data.
- Returning a string value will translate to the corresponding result (e.g. "pass" is a Pass result)

Accessibility

Diagnoses common accessibility problems affecting screen readers and other assistive technology.

No Result

Press [Start] to start running the audit
Creating Web Inspector Audits

This post is a followup to the Audits in Web Inspector post, and explains the capabilities of and how to write an Audit.

Test Case Format

The actual test that is run in the page is just a string. A JavaScript function. async functions are allowed, as well as non-async functions that return a Promise. The only requirement is that the return value of the function (or Promise) conforms to the following rules:

- Returning true/false will translate to a Pass/Fail result with no additional data.
- Returning a string value will translate to the corresponding result (e.g. “pass” is a Pass result)

Accessibility

Diagnoses common accessibility problems affecting screen readers and other assistive technology.

No Result

Press [Start] to start running the audit
Summary
Summary

Use the Audit tab to check your web content
Summary

Use the Audit tab to check your web content

Develop custom audits for your development work
Summary

Use the Audit tab to check your web content

Develop custom audits for your development work

File bug reports and share feedback