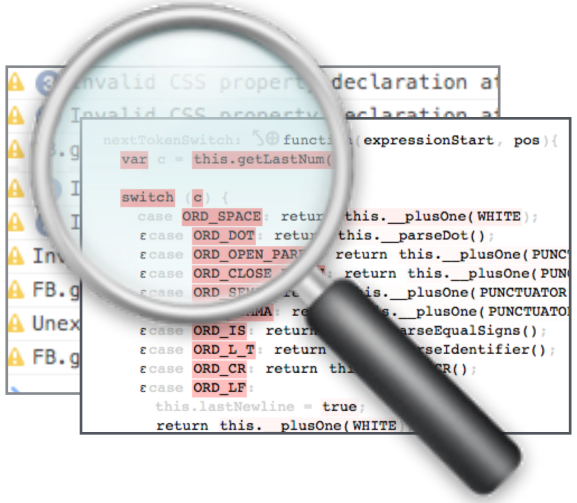
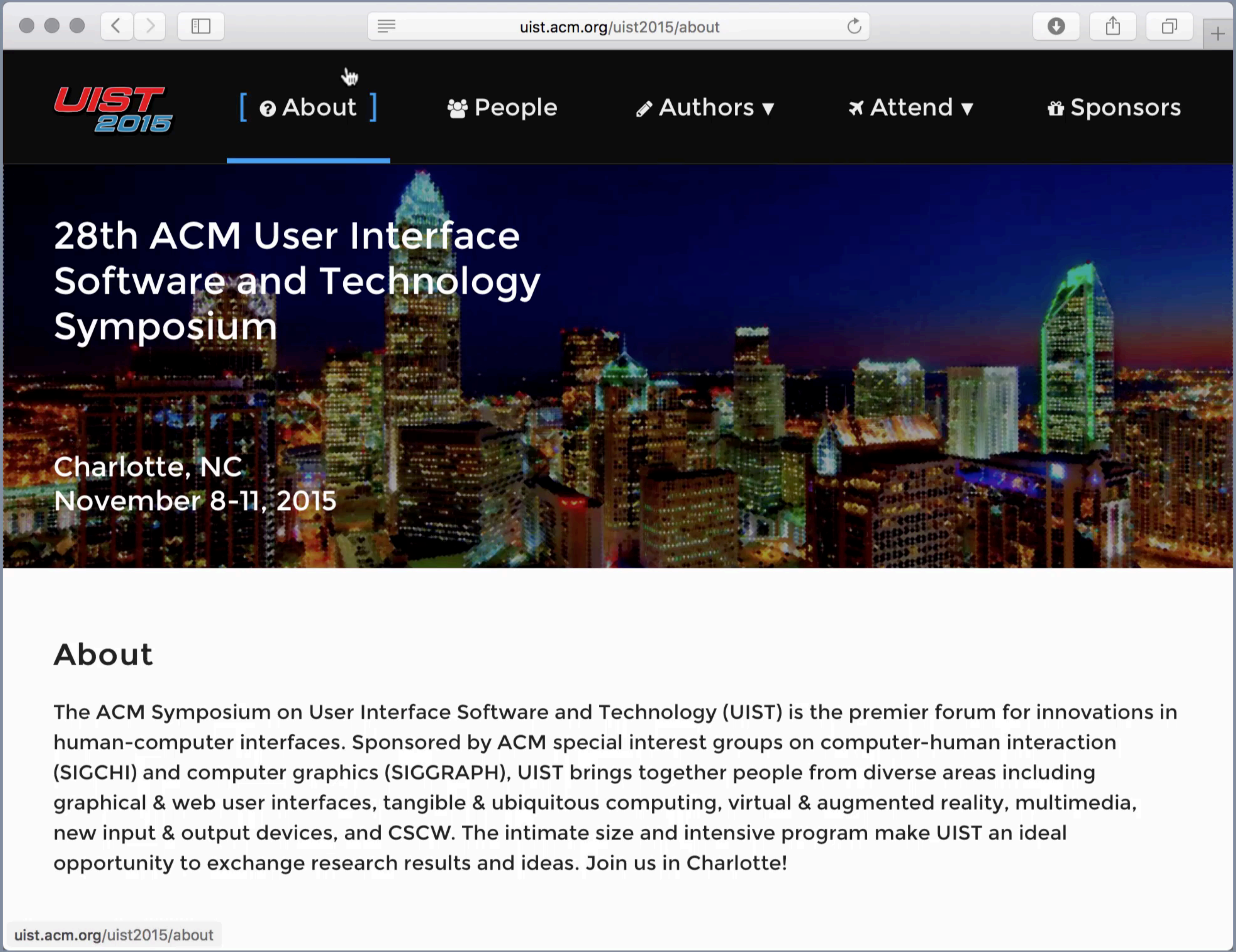


Explaining Visual Changes in Web Interfaces

Brian Burg, Andrew J Ko, Michael Ernst

University of Washington



28th ACM User Interface Software and Technology Symposium

Charlotte, NC
November 8-11, 2015

About

The ACM Symposium on User Interface Software and Technology (UIST) is the premier forum for innovations in human-computer interfaces. Sponsored by ACM special interest groups on computer-human interaction (SIGCHI) and computer graphics (SIGGRAPH), UIST brings together people from diverse areas including graphical & web user interfaces, tangible & ubiquitous computing, virtual & augmented reality, multimedia, new input & output devices, and CSCW. The intimate size and intensive program make UIST an ideal opportunity to exchange research results and ideas. Join us in Charlotte!

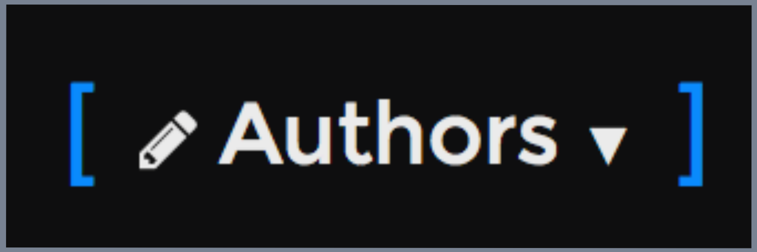
Feature Location

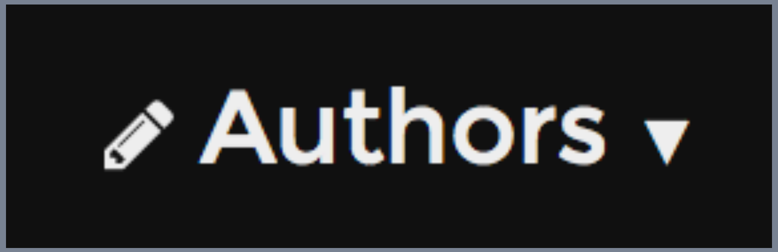
__Feature location__ is the activity of identifying an initial location in the source code that implements functionality in a software system.

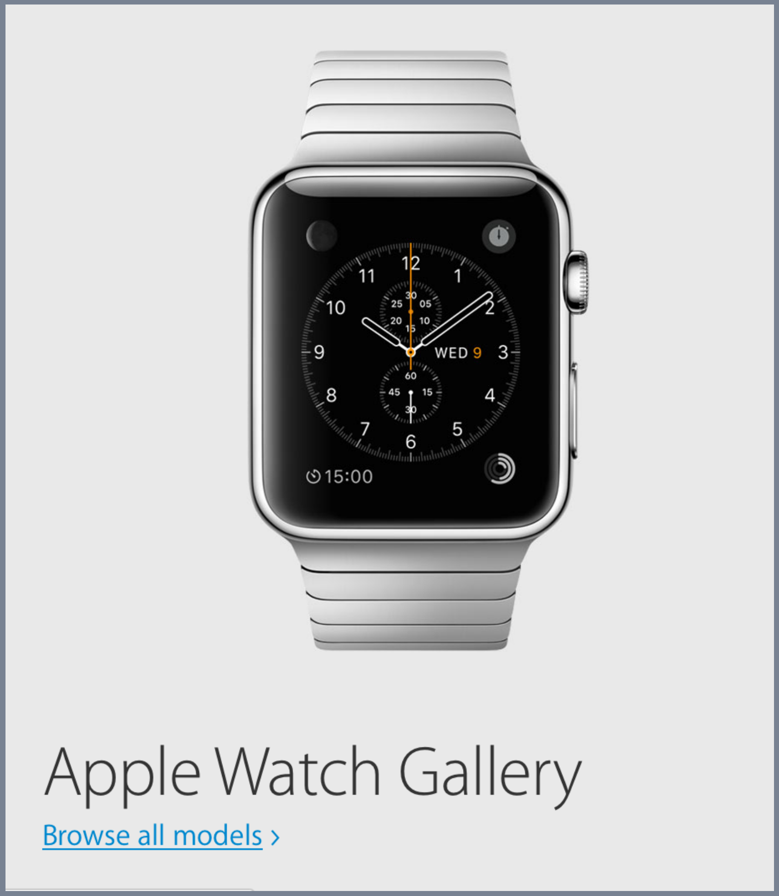
Dit, B., Revelle, M., Gethers, M., and Poshyvanyk, D., "Feature Location in Source Code: A Taxonomy and Survey", Journal of Software: Evolution and Process (JSEP), vol. 25, no. 1, January 2013, pp. 53–95

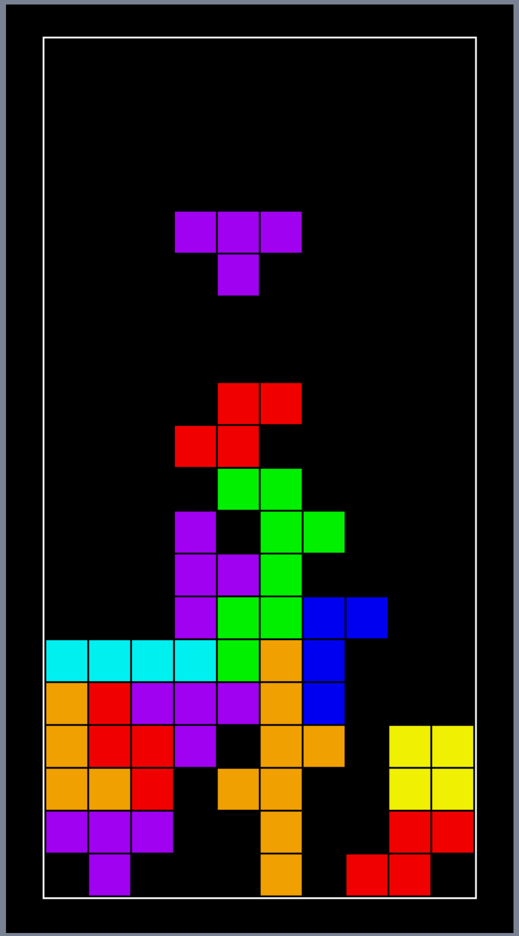
Feature Location

for Interactive Web Content







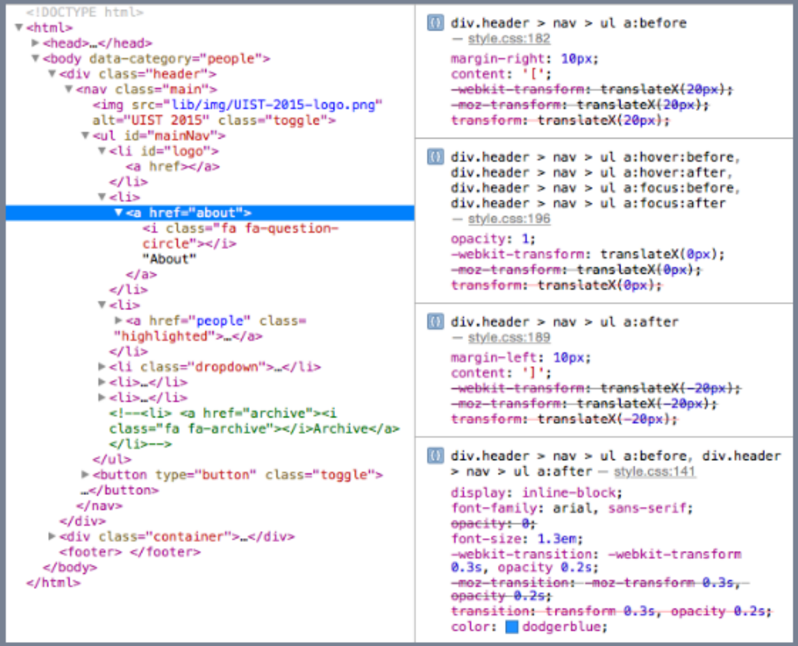


How can I interact with it?

Visual Output

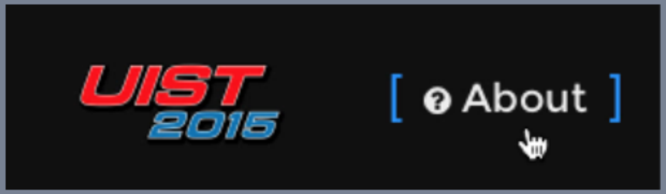
😄

How is the visual effect achieved?



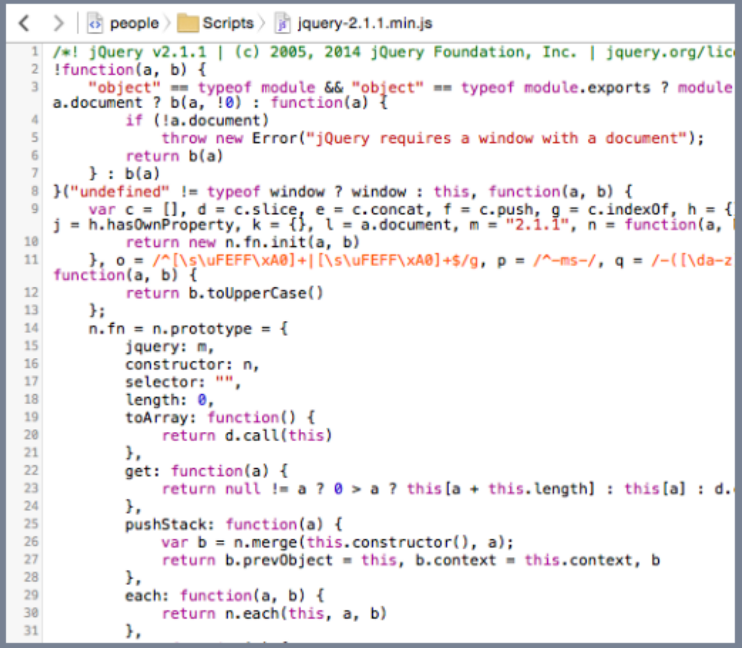
```
<!DOCTYPE html>
▼<html>
  ▶<head>…</head>
  ▼<body data-category="people">
    ▼<div class="header">
      ▼<nav class="main">
          <img src="lib/img/UIST-2015-logo.png"
          alt="UIST 2015" class="toggle">
        ▼<ul id="mainNav">
          ▼<li id="logo">
              <a href></a>
            </li>
          ▼<li>
            ▼<a href="about">
                <i class="fa fa-question-
                circle"></i>
                "About"
              </a>
            </li>
          ▼<li>
            ▶<a href="people" class=
            "highlighted">…</a>
            </li>
          ▶<li class="dropdown">…</li>
          ▶<li>…</li>
          ▶<li>…</li>
            <!--<li> <a href="archive"><i
            class="fa fa-archive"></i>Archive</a>
            </li>-->
          </ul>
        ▶<button type="button" class="toggle">
        …</button>
        </nav>
      </div>
    ▶<div class="container">…</div>
      <footer> </footer>
    </body>
</html>
```

```
div.header > nav > ul a:before
— style.css:182
margin-right: 10px;
content: '[';
~~-webkit-transform: translateX(20px);~~
~~-moz-transform: translateX(20px);~~
transform: ~~translateX(20px);~~

div.header > nav > ul a:hover:before,
div.header > nav > ul a:hover:after,
div.header > nav > ul a:focus:before,
div.header > nav > ul a:focus:after
— style.css:196
opacity: 1;
-webkit-transform: translateX(0px);
~~-moz-transform: translateX(0px);~~
transform: ~~translateX(0px);~~

div.header > nav > ul a:after
— style.css:189
margin-left: 10px;
content: ']';
~~-webkit-transform: translateX(-20px);~~
~~-moz-transform: translateX(-20px);~~
transform: ~~translateX(-20px);~~

div.header > nav > ul a:before, div.header
> nav > ul a:after — style.css:141
display: inline-block;
font-family: arial, sans-serif;
~~opacity: 0;~~
font-size: 1.3em;
-webkit-transition: -webkit-transform
0.3s, opacity 0.2s;
~~-moz-transition: -moz-transform 0.3s,
opacity 0.2s;~~
transition: ~~transform 0.3s, opacity 0.2s;~~
color: 🟦 dodgerblue;
```



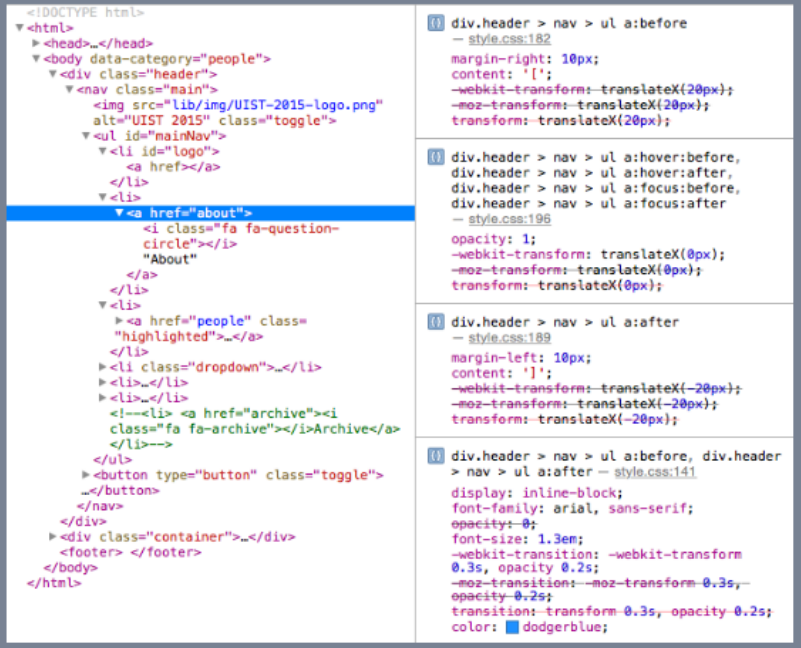
DOM & CSS ← Visual Output

What code is ultimately responsible?



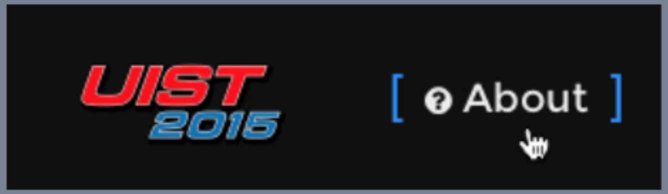
people › Scripts › jquery-2.1.1.min.js

```
/*! jQuery v2.1.1 | (c) 2005, 2014 jQuery Foundation, Inc. | jquery.org/lic
!function(a, b) {
    "object" == typeof module && "object" == typeof module.exports ? module
a.document ? b(a, !0) : function(a) {
        if (!a.document)
            throw new Error("jQuery requires a window with a document");
        return b(a)
    } : b(a)
}("undefined" != typeof window ? window : this, function(a, b) {
    var c = [], d = c.slice, e = c.concat, f = c.push, g = c.indexOf, h = {
j = h.hasOwnProperty, k = {}, l = a.document, m = "2.1.1", n = function(a,
        return new n.fn.init(a, b)
    }, o = /^[\s\uFEFF\xA0]+|[\s\uFEFF\xA0]+$/g, p = /^-ms-/, q = /-([\da-z
function(a, b) {
        return b.toUpperCase()
    };
    n.fn = n.prototype = {
        jquery: m,
        constructor: n,
        selector: "",
        length: 0,
        toArray: function() {
            return d.call(this)
        },
        get: function(a) {
            return null != a ? 0 > a ? this[a + this.length] : this[a] : d.
        },
        pushStack: function(a) {
            var b = n.merge(this.constructor(), a);
            return b.prevObject = this, b.context = this.context, b
        },
        each: function(a, b) {
            return n.each(this, a, b)
        },
```



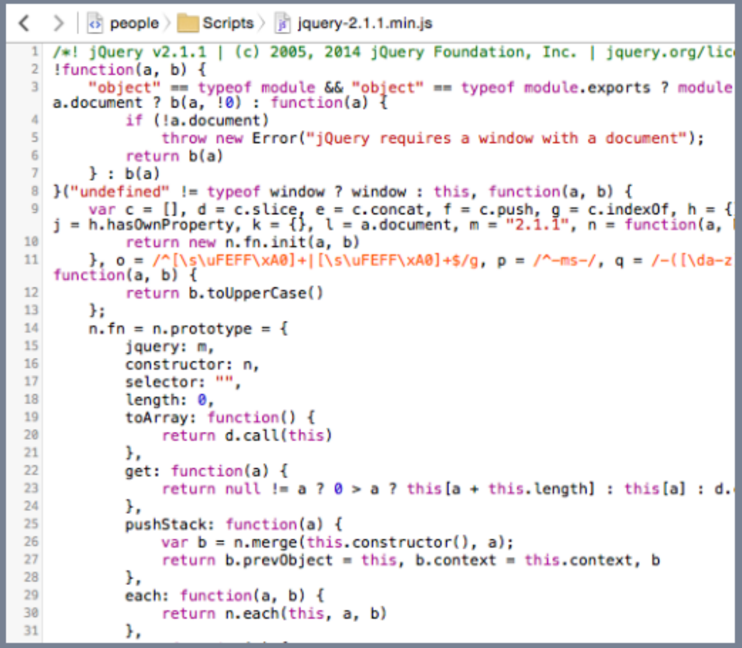
```
<!DOCTYPE html>
▼<html>
  ▶<head>…</head>
  ▼<body data-category="people">
    ▼<div class="header">
      ▼<nav class="main">
          <img src="lib/img/UIST-2015-logo.png"
          alt="UIST 2015" class="toggle">
        ▼<ul id="mainNav">
          ▼<li id="logo">
              <a href></a>
            </li>
          ▼<li>
            ▼<a href="about">
                <i class="fa fa-question-
                circle"></i>
                "About"
              </a>
            </li>
          ▼<li>
            ▶<a href="people" class=
            "highlighted">…</a>
            </li>
          ▶<li class="dropdown">…</li>
          ▶<li>…</li>
          ▶<li>…</li>
            <!--<li> <a href="archive"><i
            class="fa fa-archive"></i>Archive</a>
            </li>-->
          </ul>
        ▶<button type="button" class="toggle">
        …</button>
        </nav>
      </div>
    ▶<div class="container">…</div>
      <footer> </footer>
    </body>
</html>
```

```
div.header > nav > ul a:before
  — style.css:182
  margin-right: 10px;
  content: '[';
  ~~-webkit-transform: translateX(20px);~~
  ~~-moz-transform: translateX(20px);~~
  ~~transform: translateX(20px);~~

div.header > nav > ul a:hover:before,
div.header > nav > ul a:hover:after,
div.header > nav > ul a:focus:before,
div.header > nav > ul a:focus:after
  — style.css:196
  opacity: 1;
  -webkit-transform: translateX(0px);
  ~~-moz-transform: translateX(0px);~~
  ~~transform: translateX(0px);~~

div.header > nav > ul a:after
  — style.css:189
  margin-left: 10px;
  content: ']';
  ~~-webkit-transform: translateX(-20px);~~
  ~~-moz-transform: translateX(-20px);~~
  ~~transform: translateX(-20px);~~

div.header > nav > ul a:before, div.header
> nav > ul a:after — style.css:141
  display: inline-block;
  font-family: arial, sans-serif;
  ~~opacity: 0;~~
  font-size: 1.3em;
  -webkit-transition: -webkit-transform
  0.3s, opacity 0.2s;
  ~~-moz-transition: -moz-transform 0.3s,
  opacity 0.2s;~~
  ~~transition: transform 0.3s, opacity 0.2s;~~
  color: 🟦 dodgerblue;
```

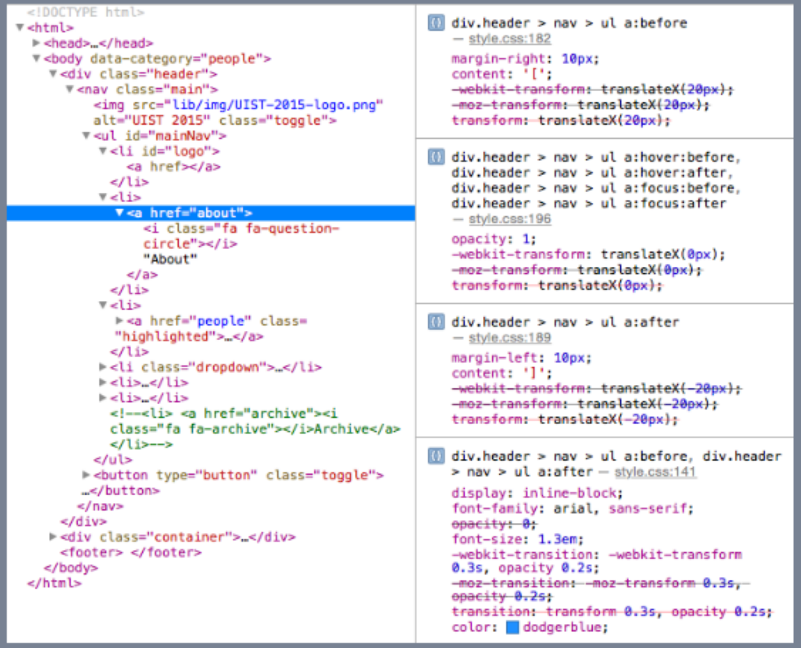


JavaScript ← **DOM & CSS** ← **Visual Output**

😦 🤔 😄

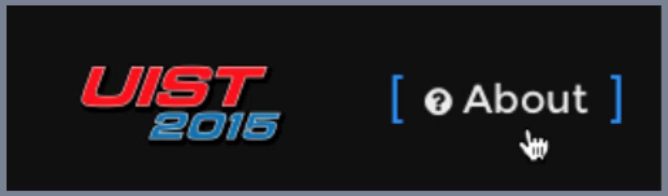


```
/*! jQuery v2.1.1 | (c) 2005, 2014 jQuery Foundation, Inc. | jquery.org/lic
!function(a, b) {
    "object" == typeof module && "object" == typeof module.exports ? module
a.document ? b(a, !0) : function(a) {
        if (!a.document)
            throw new Error("jQuery requires a window with a document");
        return b(a)
    } : b(a)
}("undefined" != typeof window ? window : this, function(a, b) {
    var c = [], d = c.slice, e = c.concat, f = c.push, g = c.indexOf, h = {
j = h.hasOwnProperty, k = {}, l = a.document, m = "2.1.1", n = function(a,
        return new n.fn.init(a, b)
    }, o = /^[\s\uFEFF\xA0]+|[\s\uFEFF\xA0]+$/g, p = /^-ms-/, q = /-([\da-z
function(a, b) {
        return b.toUpperCase()
    };
    n.fn = n.prototype = {
        jquery: m,
        constructor: n,
        selector: "",
        length: 0,
        toArray: function() {
            return d.call(this)
        },
        get: function(a) {
            return null != a ? 0 > a ? this[a + this.length] : this[a] : d.
        },
        pushStack: function(a) {
            var b = n.merge(this.constructor(), a);
            return b.prevObject = this, b.context = this.context, b
        },
        each: function(a, b) {
            return n.each(this, a, b)
        },
```



```
<!DOCTYPE html>
▼<html>
  ▶<head>…</head>
  ▼<body data-category="people">
    ▼<div class="header">
      ▼<nav class="main">
          <img src="lib/img/UIST-2015-logo.png"
          alt="UIST 2015" class="toggle">
        ▼<ul id="mainNav">
          ▼<li id="logo">
              <a href></a>
            </li>
          ▼<li>
            ▼<a href="about">
                <i class="fa fa-question-
                circle"></i>
                "About"
              </a>
            </li>
          ▼<li>
            ▶<a href="people" class=
            "highlighted">…</a>
            </li>
          ▶<li class="dropdown">…</li>
          ▶<li>…</li>
          ▶<li>…</li>
            <!--<li> <a href="archive"><i
            class="fa fa-archive"></i>Archive</a>
            </li>-->
          </ul>
        ▶<button type="button" class="toggle">
        …</button>
        </nav>
      </div>
    ▶<div class="container">…</div>
      <footer> </footer>
    </body>
</html>
```

```
div.header > nav > ul a:before
— style.css:182
margin-right: 10px;
content: '[';
-webkit-transform: translateX(20px);
-moz-transform: translateX(20px);
transform: translateX(20px);

div.header > nav > ul a:hover:before,
div.header > nav > ul a:hover:after,
div.header > nav > ul a:focus:before,
div.header > nav > ul a:focus:after
— style.css:196
opacity: 1;
-webkit-transform: translateX(0px);
-moz-transform: translateX(0px);
transform: translateX(0px);

div.header > nav > ul a:after
— style.css:189
margin-left: 10px;
content: ']';
-webkit-transform: translateX(-20px);
-moz-transform: translateX(-20px);
transform: translateX(-20px);

div.header > nav > ul a:before, div.header
> nav > ul a:after — style.css:141
display: inline-block;
font-family: arial, sans-serif;
opacity: 0;
font-size: 1.3em;
-webkit-transition: -webkit-transform
0.3s, opacity 0.2s;
-moz-transition: -moz-transform 0.3s,
opacity 0.2s;
transition: transform 0.3s, opacity 0.2s;
color: ■ dodgerblue;
```



JavaScript ← **DOM & CSS** ← **Visual Output**

No Links Between Source and State

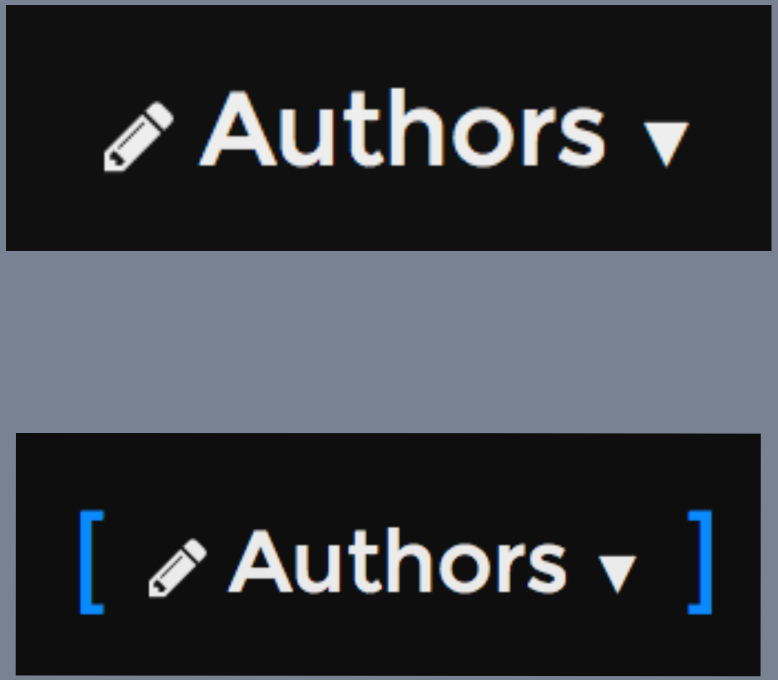
Scattered State

No Output History

Scry

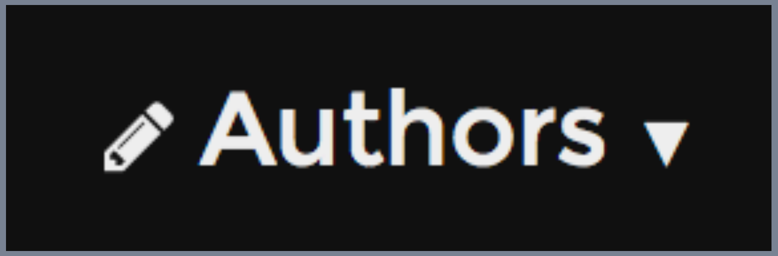
Staged, Interactive Feature Location

Staged, Interactive Feature Location

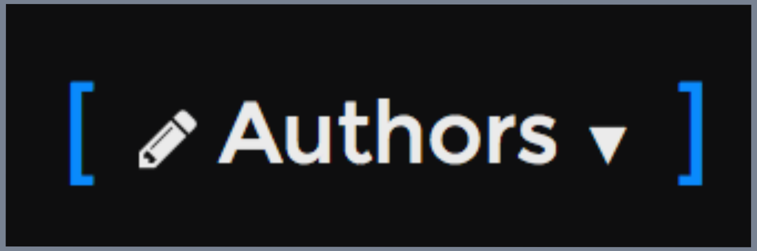


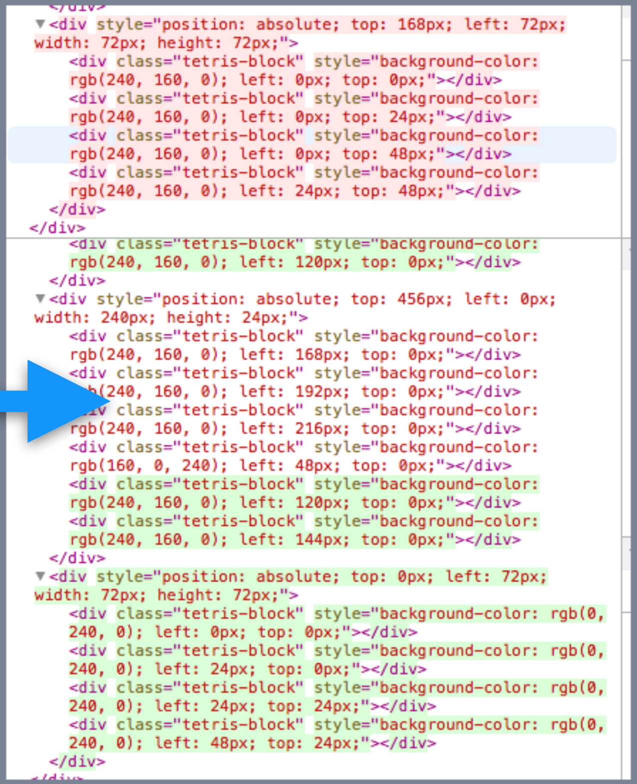
Output Examples

Staged, Interactive Feature Location



Δ(*pre*, *post*)

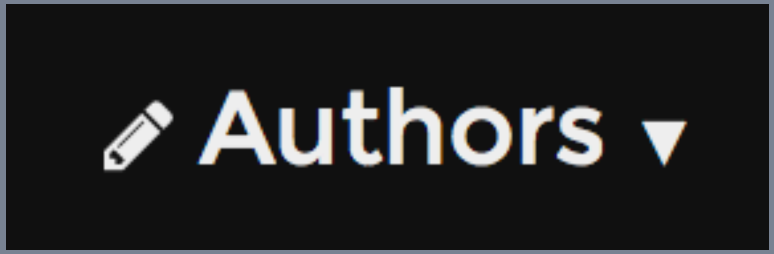




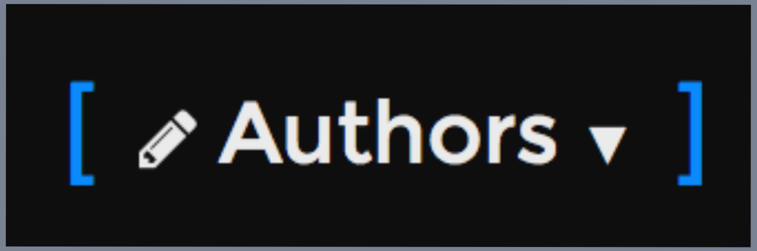
```
  </div>
▼<div style="position: absolute; top: 168px; left: 72px;
width: 72px; height: 72px;">
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 0px; top: 0px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 0px; top: 24px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 0px; top: 48px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 24px; top: 48px;"></div>
  </div>
</div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 120px; top: 0px;"></div>
  </div>
▼<div style="position: absolute; top: 456px; left: 0px;
width: 240px; height: 24px;">
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 168px; top: 0px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 192px; top: 0px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 216px; top: 0px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(160, 0, 240); left: 48px; top: 0px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 120px; top: 0px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 144px; top: 0px;"></div>
  </div>
▼<div style="position: absolute; top: 0px; left: 72px;
width: 72px; height: 72px;">
    <div class="tetris-block" style="background-color: rgb(0,
    240, 0); left: 0px; top: 0px;"></div>
    <div class="tetris-block" style="background-color: rgb(0,
    240, 0); left: 24px; top: 0px;"></div>
    <div class="tetris-block" style="background-color: rgb(0,
    240, 0); left: 24px; top: 24px;"></div>
    <div class="tetris-block" style="background-color: rgb(0,
    240, 0); left: 48px; top: 24px;"></div>
  </div>
</div>
```

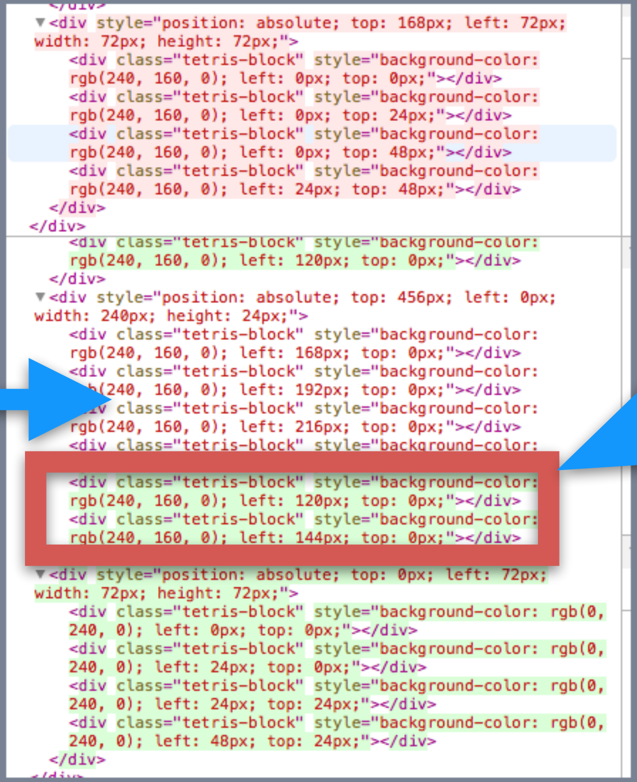
Output Examples **State Differences**

Staged, Interactive Feature Location



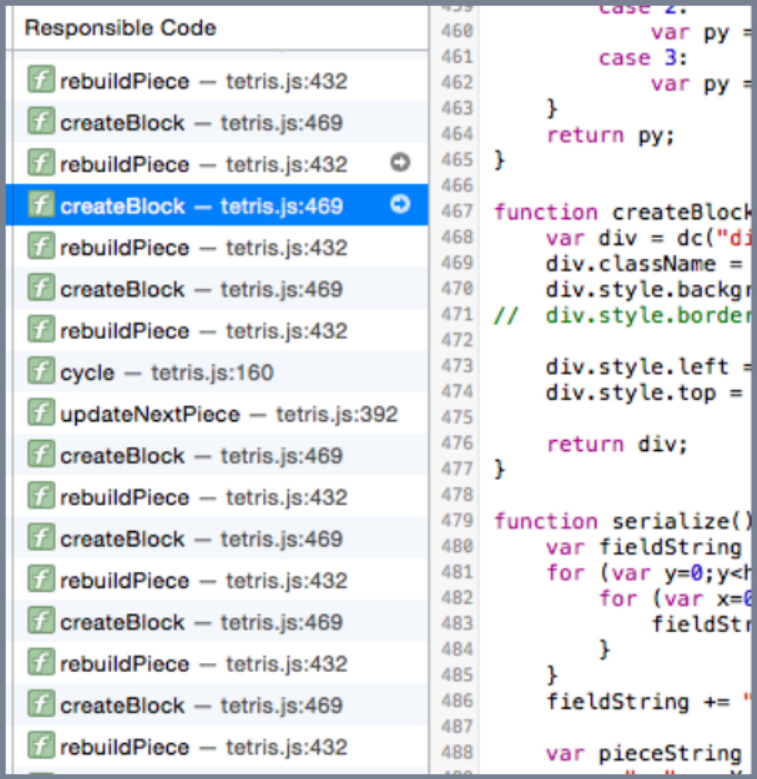
Δ(*pre*, *post*)





```
  </div>
▼<div style="position: absolute; top: 168px; left: 72px;
width: 72px; height: 72px;">
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 0px; top: 0px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 0px; top: 24px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 0px; top: 48px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 24px; top: 48px;"></div>
  </div>
</div>
```

```
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 120px; top: 0px;"></div>
  </div>
▼<div style="position: absolute; top: 456px; left: 0px;
width: 240px; height: 24px;">
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 168px; top: 0px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 192px; top: 0px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 216px; top: 0px;"></div>
    <div class="tetris-block" style="background-color:

    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 120px; top: 0px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 144px; top: 0px;"></div>
▼<div style="position: absolute; top: 0px; left: 72px;
width: 72px; height: 72px;">
    <div class="tetris-block" style="background-color: rgb(0,
    240, 0); left: 0px; top: 0px;"></div>
    <div class="tetris-block" style="background-color: rgb(0,
    240, 0); left: 24px; top: 0px;"></div>
    <div class="tetris-block" style="background-color: rgb(0,
    240, 0); left: 24px; top: 24px;"></div>
    <div class="tetris-block" style="background-color: rgb(0,
    240, 0); left: 48px; top: 24px;"></div>
  </div>
</div>
```

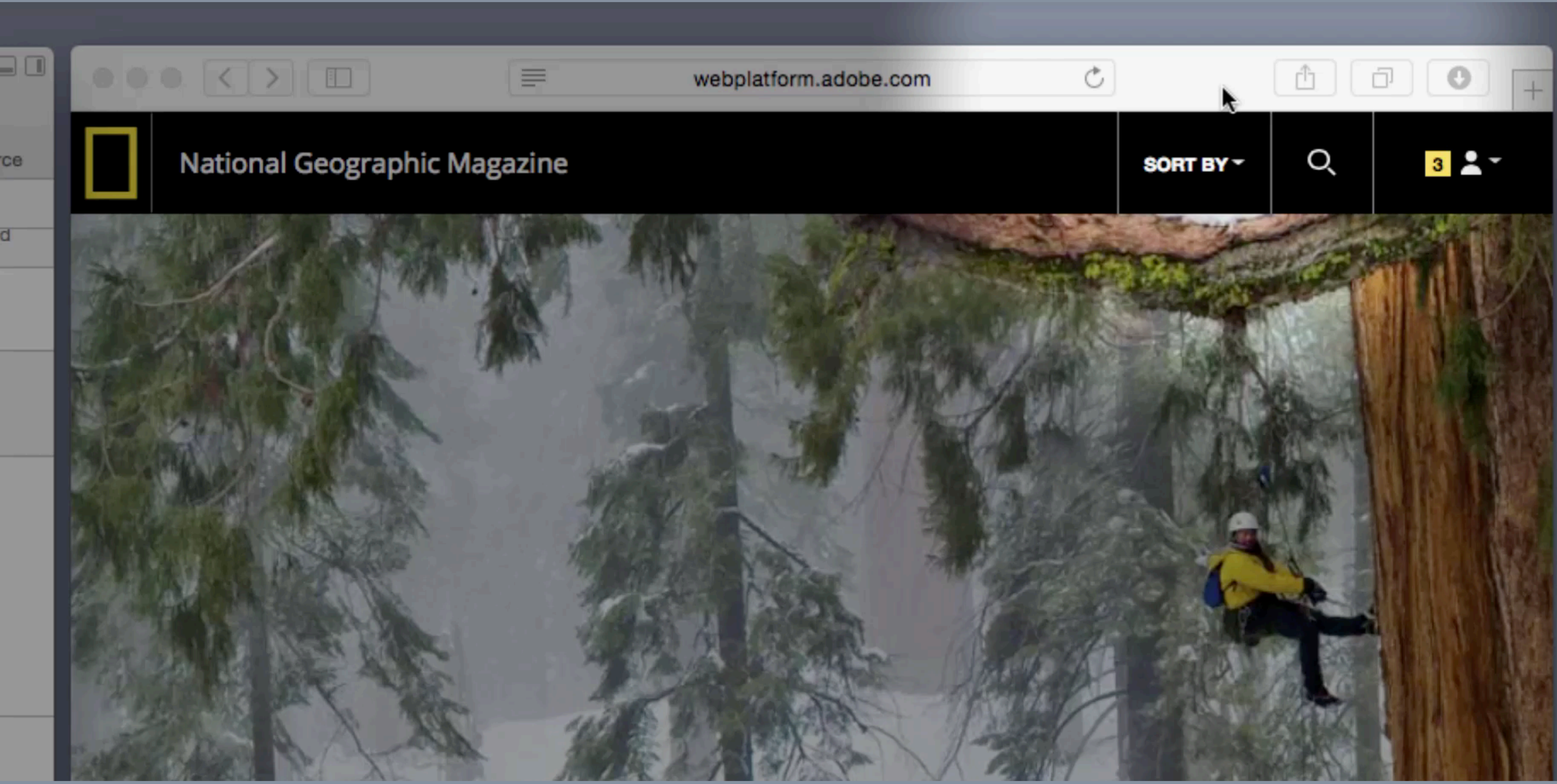


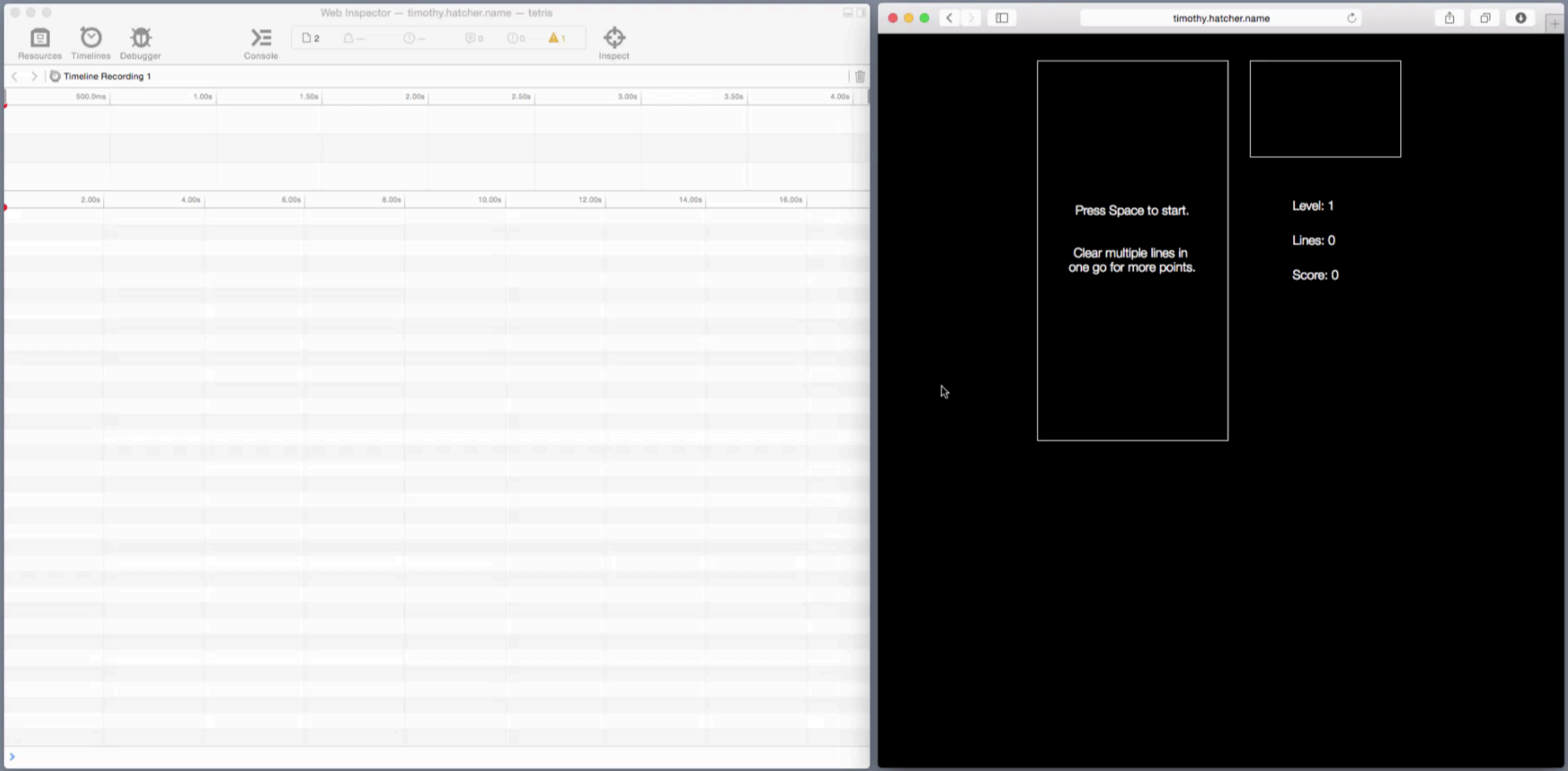
Responsible Code

```
rebuildPiece — tetris.js:432
createBlock — tetris.js:469
rebuildPiece — tetris.js:432
createBlock — tetris.js:469
rebuildPiece — tetris.js:432
createBlock — tetris.js:469
rebuildPiece — tetris.js:432
cycle — tetris.js:160
updateNextPiece — tetris.js:392
createBlock — tetris.js:469
rebuildPiece — tetris.js:432
createBlock — tetris.js:469
rebuildPiece — tetris.js:432
createBlock — tetris.js:469
rebuildPiece — tetris.js:432
createBlock — tetris.js:469
rebuildPiece — tetris.js:432
```

```
        case 2:
            var py =
        case 3:
            var py =
    }
    return py;
}

function createBlock
    var div = dc("di
    div.className =
    div.style.backgr
//  div.style.border

    div.style.left =
    div.style.top =

    return div;
}

function serialize()
    var fieldString
    for (var y=0;y<h
        for (var x=0
            fieldStr
        }
    }
    fieldString += "

    var pieceString
```

Output Examples **State Differences** **JavaScript Mutations**


webplatform.adobe.com
National Geographic Magazine
SORT BY
3



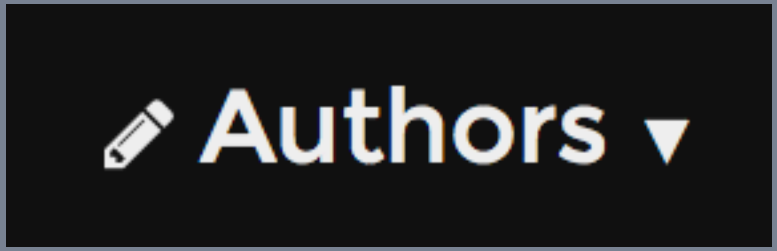
Press Space to start.

Clear multiple lines in one go for more points.

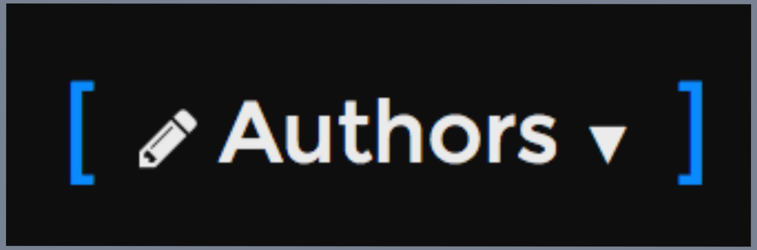
Level: 1

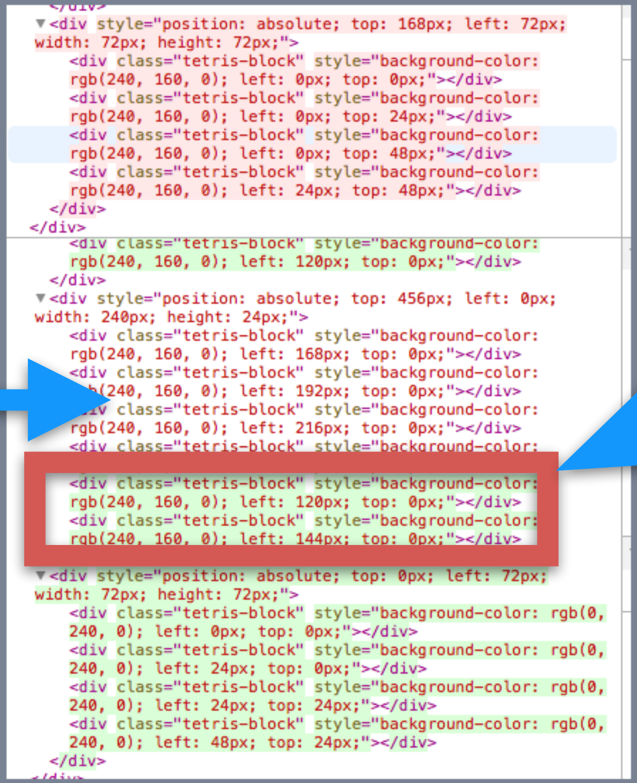
Lines: 0

Score: 0



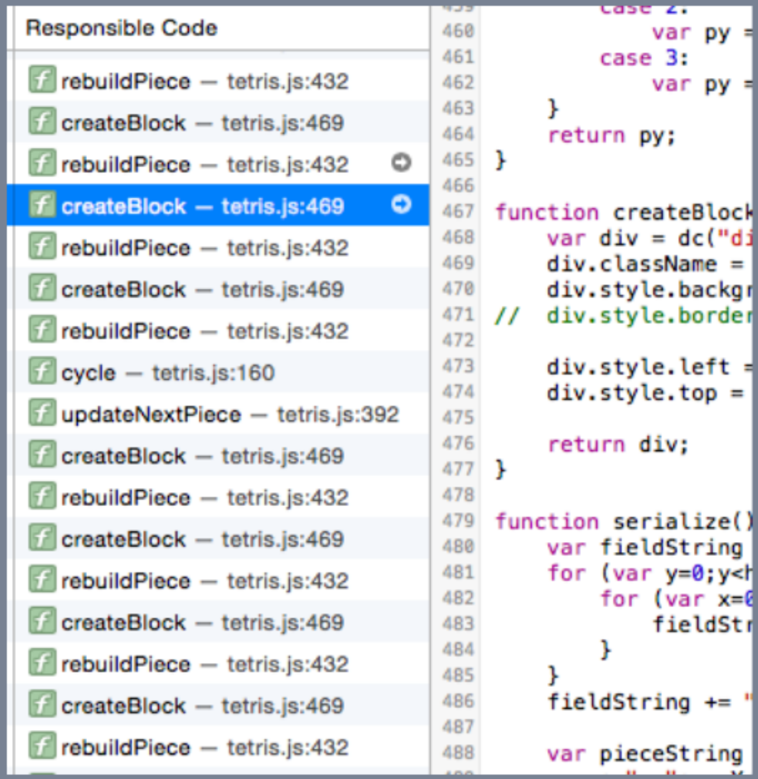
Δ(pre, post)





```
▼<div style="position: absolute; top: 168px; left: 72px;
width: 72px; height: 72px;">
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 0px; top: 0px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 0px; top: 24px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 0px; top: 48px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 24px; top: 48px;"></div>
  </div>
</div>
```

```
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 120px; top: 0px;"></div>
  </div>
▼<div style="position: absolute; top: 456px; left: 0px;
width: 240px; height: 24px;">
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 168px; top: 0px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 192px; top: 0px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 216px; top: 0px;"></div>
    <div class="tetris-block" style="background-color:
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 120px; top: 0px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 144px; top: 0px;"></div>
▼<div style="position: absolute; top: 0px; left: 72px;
width: 72px; height: 72px;">
    <div class="tetris-block" style="background-color: rgb(0,
    240, 0); left: 0px; top: 0px;"></div>
    <div class="tetris-block" style="background-color: rgb(0,
    240, 0); left: 24px; top: 0px;"></div>
    <div class="tetris-block" style="background-color: rgb(0,
    240, 0); left: 24px; top: 24px;"></div>
    <div class="tetris-block" style="background-color: rgb(0,
    240, 0); left: 48px; top: 24px;"></div>
  </div>
```



Responsible Code

```
rebuildPiece — tetris.js:432
createBlock — tetris.js:469
rebuildPiece — tetris.js:432
createBlock — tetris.js:469
rebuildPiece — tetris.js:432
createBlock — tetris.js:469
rebuildPiece — tetris.js:432
cycle — tetris.js:160
updateNextPiece — tetris.js:392
createBlock — tetris.js:469
rebuildPiece — tetris.js:432
createBlock — tetris.js:469
rebuildPiece — tetris.js:432
createBlock — tetris.js:469
rebuildPiece — tetris.js:432
createBlock — tetris.js:469
rebuildPiece — tetris.js:432
```

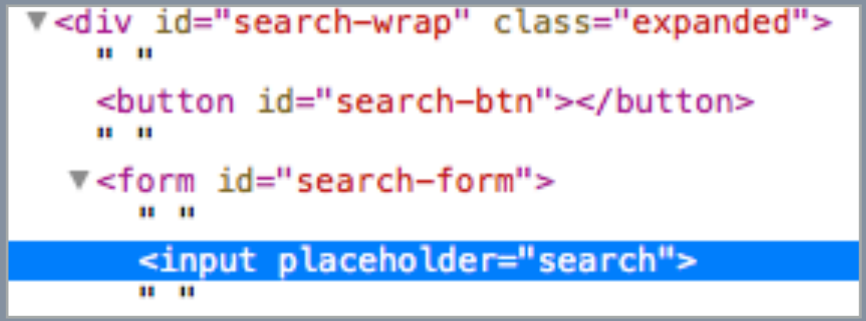
```
                var py =
            case 3:
                var py =
        }
        return py;
}

function createBlock
    var div = dc("di
    div.className =
    div.style.backgr
//  div.style.border

    div.style.left =
    div.style.top =

    return div;
}

function serialize()
    var fieldString
    for (var y=0;y<h
        for (var x=0
            fieldStr
        }
    }
    fieldString += "

    var pieceString
```

Output Examples

State Differences

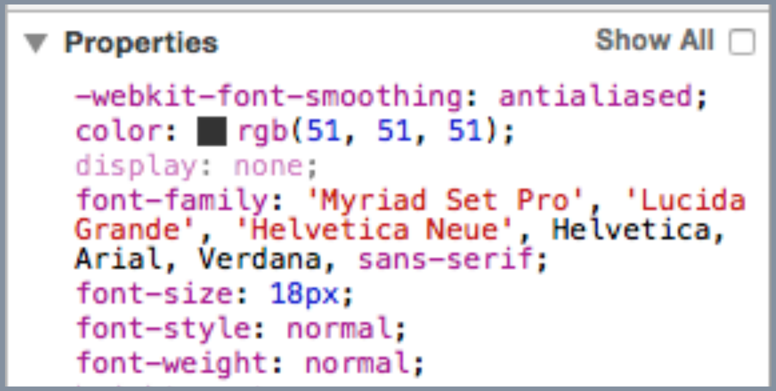
JavaScript Mutations

What determines visual appearance?



```
▼<div id="search-wrap" class="expanded">
   " "
   <button id="search-btn"></button>
   " "
  ▼<form id="search-form">
     " "
     <input placeholder="search">
     " "
```

DOM Tree Structure



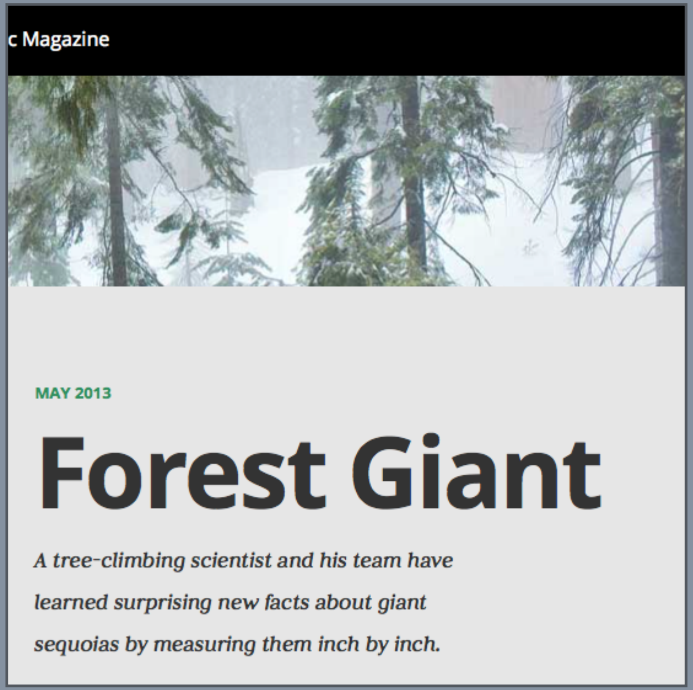
▼ Properties Show All □

```
-webkit-font-smoothing: antialiased;
color: ■rgb(51, 51, 51);
display: none;
font-family: 'Myriad Set Pro', 'Lucida
Grande', 'Helvetica Neue', Helvetica,
Arial, Verdana, sans-serif;
font-size: 18px;
font-style: normal;
font-weight: normal;
```

CSS Style Properties

Color, layout mode, visual styling,
text rendering, handling of children

Rendering Engine



c Magazine

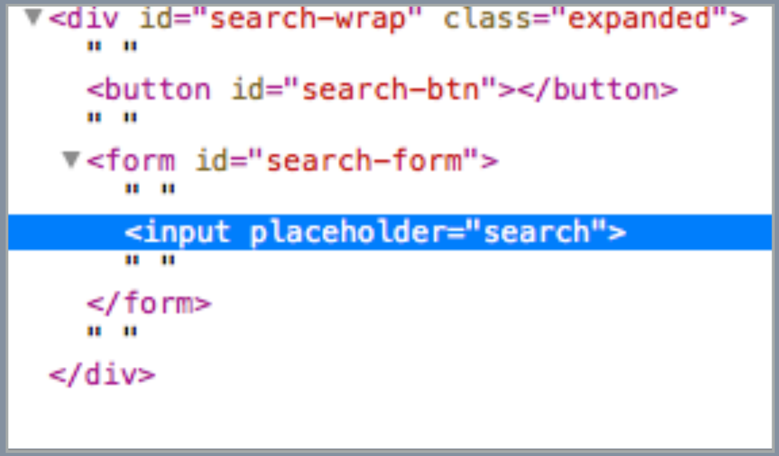
MAY 2013

Forest Giant

A tree-climbing scientist and his team have learned surprising new facts about giant sequoias by measuring them inch by inch.

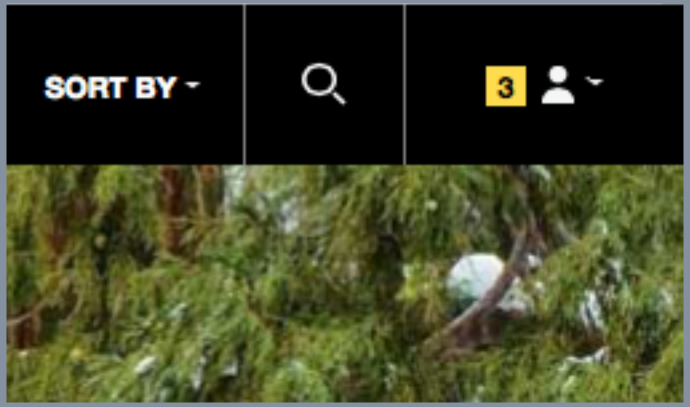
Visual State

Target Element Snapshots

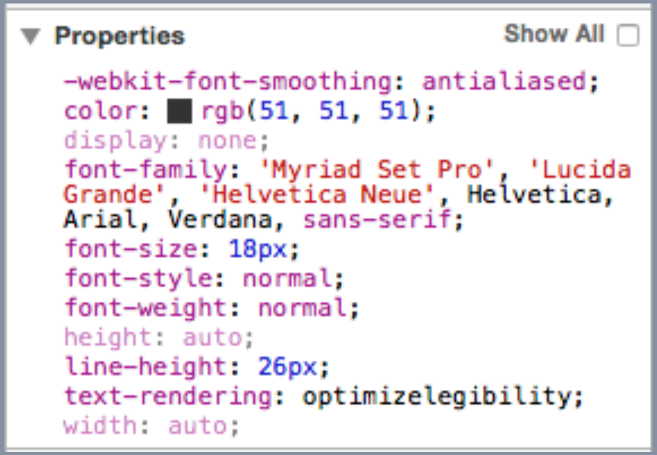


```
▼<div id="search-wrap" class="expanded">
   " "
   <button id="search-btn"></button>
   " "
  ▼<form id="search-form">
     " "
     <input placeholder="search">
     " "
   </form>
   " "
 </div>
```

DOM Subtree

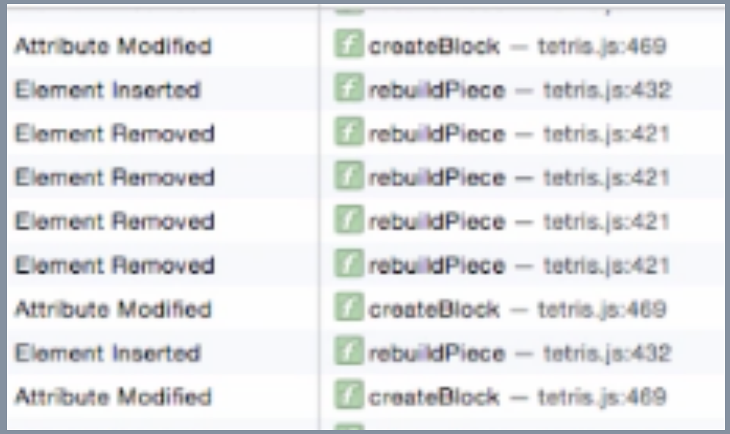


Element Screenshots



```
▼ Properties                                    Show All □
    -webkit-font-smoothing: antialiased;
    color: ■ rgb(51, 51, 51);
    display: none;
    font-family: 'Myriad Set Pro', 'Lucida
    Grande', 'Helvetica Neue', Helvetica,
    Arial, Verdana, sans-serif;
    font-size: 18px;
    font-style: normal;
    font-weight: normal;
    height: auto;
    line-height: 26px;
    text-rendering: optimizelegibility;
    width: auto;
```

Computed Styles & Related Rules

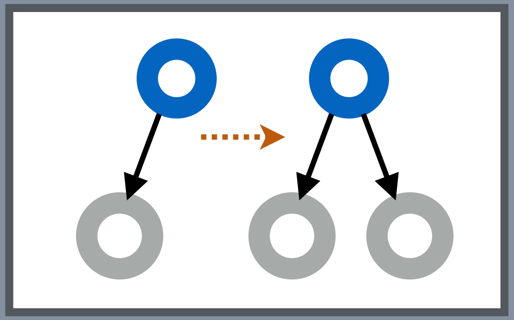


```
Attribute Modified      createBlock — tetris.js:469
Element Inserted        rebuildPiece — tetris.js:432
Element Removed         rebuildPiece — tetris.js:421
Element Removed         rebuildPiece — tetris.js:421
Element Removed         rebuildPiece — tetris.js:421
Element Removed         rebuildPiece — tetris.js:421
Attribute Modified      createBlock — tetris.js:469
Element Inserted        rebuildPiece — tetris.js:432
Attribute Modified      createBlock — tetris.js:469
```

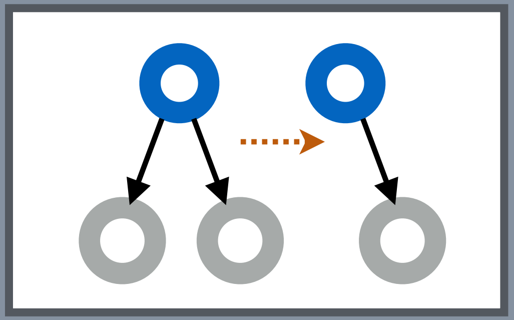
Mutation Operations

Why does visual appearance change?

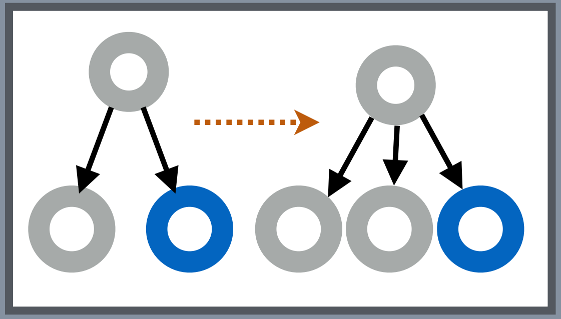
DOM Tree Mutations



Insertion



Deletion



Ordinal Change

```
class="active"
textContent = …
:hover, :focus
```

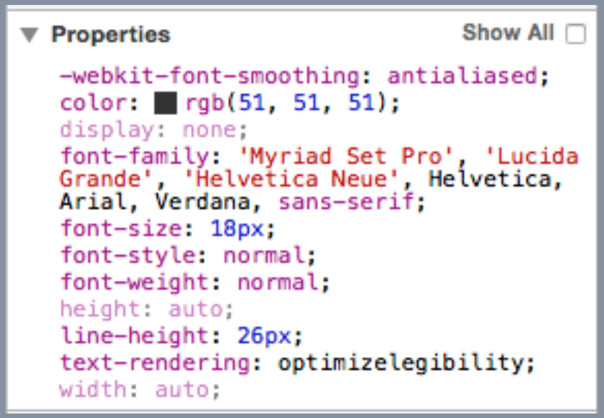
Attributes & States

Style Property Changes

Rule Changes

Inline Styles

Animations

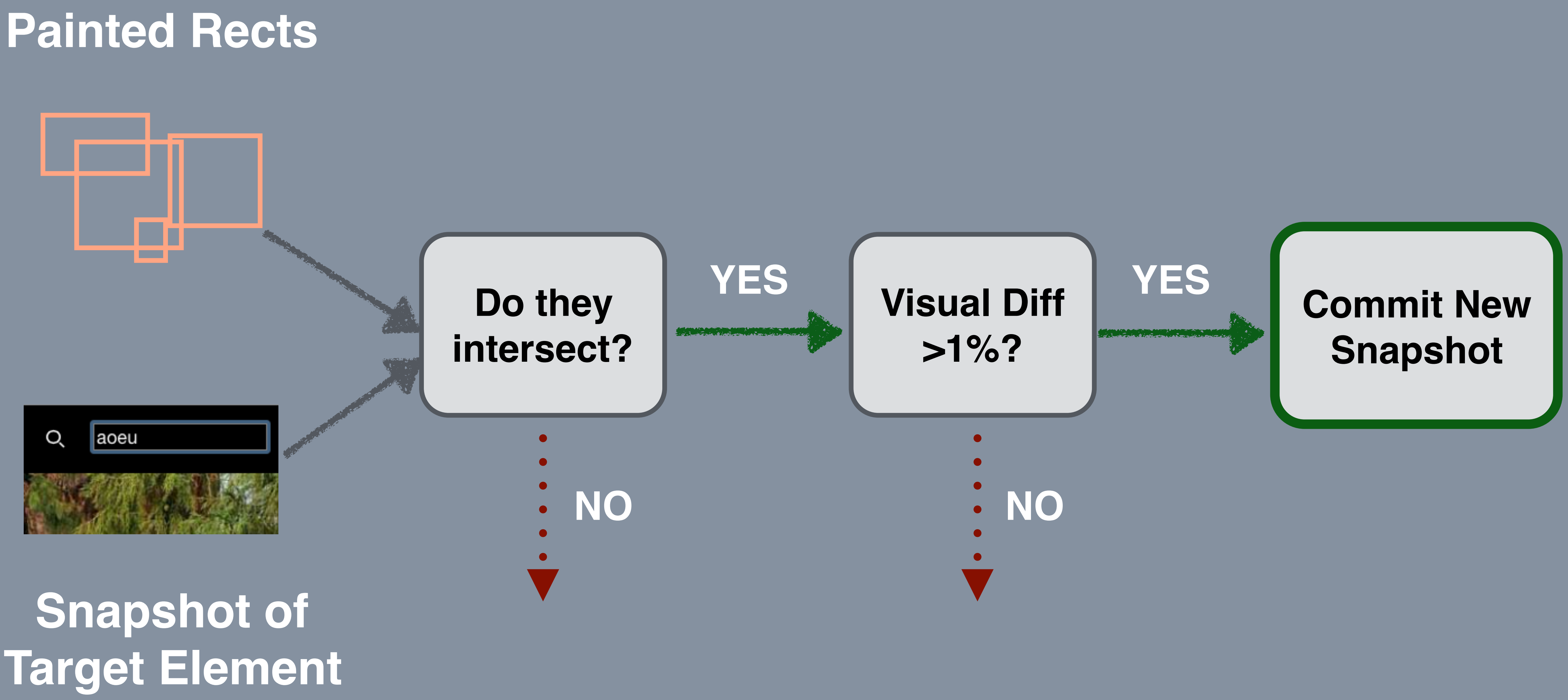


▼ Properties

Show All □

```
-webkit-font-smoothing: antialiased;
color: ■ rgb(51, 51, 51);
display: none;
font-family: 'Myriad Set Pro', 'Lucida
Grande', 'Helvetica Neue', Helvetica,
Arial, Verdana, sans-serif;
font-size: 18px;
font-style: normal;
font-weight: normal;
height: auto;
line-height: 26px;
text-rendering: optimizelegibility;
width: auto;
```

Detecting changes in appearance



Painted Rects

Snapshot of Target Element

Do they intersect?

YES

NO

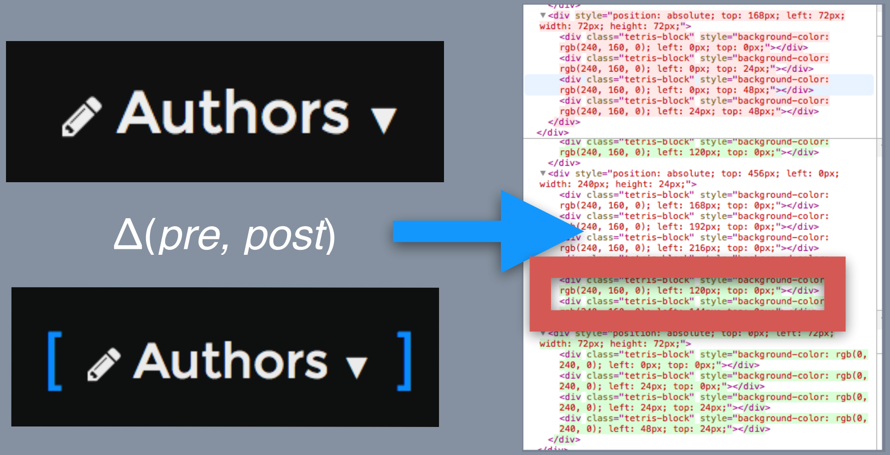
Visual Diff >1%?

YES

NO

Commit New Snapshot

Comparing State Snapshots



```
▾<div style="position: absolute; top: 168px; left: 72px;
width: 72px; height: 72px;">
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 0px; top: 0px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 0px; top: 24px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 0px; top: 48px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 24px; top: 48px;"></div>
  </div>
</div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 120px; top: 0px;"></div>
  </div>
▾<div style="position: absolute; top: 456px; left: 0px;
width: 240px; height: 24px;">
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 168px; top: 0px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 192px; top: 0px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 216px; top: 0px;"></div>
    <div class="tetris-block" style="background-color
    rgb(240, 160, 0); left: 120px; top: 0px;"></div>
    <div class="tetris-block" style="background-color
▾<div style="position: absolute; top: 0px; left: 72px;
width: 72px; height: 72px;">
    <div class="tetris-block" style="background-color: rgb(0,
    240, 0); left: 0px; top: 0px;"></div>
    <div class="tetris-block" style="background-color: rgb(0,
    240, 0); left: 24px; top: 0px;"></div>
    <div class="tetris-block" style="background-color: rgb(0,
    240, 0); left: 24px; top: 24px;"></div>
    <div class="tetris-block" style="background-color: rgb(0,
    240, 0); left: 48px; top: 24px;"></div>
  </div>
```

Per-element change summaries

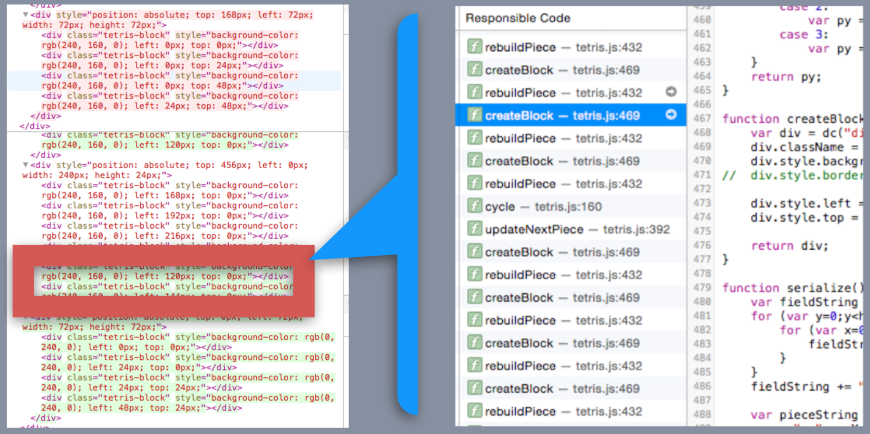
Structure: Insertion, Deletion, Attributes, …

Styles: Added, Removed, Value Change

Relies on stable DOM element identity

Doesn't work well when view state is split from DOM

Change-Relevant Operation Slicing



```
▼<div style="position: absolute; top: 168px; left: 72px;
width: 72px; height: 72px;">
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 0px; top: 0px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 0px; top: 24px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 0px; top: 48px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 24px; top: 48px;"></div>
  </div>
</div>
```

```
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 120px; top: 0px;"></div>
  </div>
▼<div style="position: absolute; top: 456px; left: 0px;
width: 240px; height: 24px;">
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 168px; top: 0px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 192px; top: 0px;"></div>
    <div class="tetris-block" style="background-color:
    rgb(240, 160, 0); left: 216px; top: 0px;"></div>
```

```
    rgb(240, 160, 0); left: 120px; top: 0px;"></div>
    <div class="tetris-block" style="background-color
```

```
width: 72px; height: 72px;">
    <div class="tetris-block" style="background-color: rgb(0,
    240, 0); left: 0px; top: 0px;"></div>
    <div class="tetris-block" style="background-color: rgb(0,
    240, 0); left: 24px; top: 0px;"></div>
    <div class="tetris-block" style="background-color: rgb(0,
    240, 0); left: 24px; top: 24px;"></div>
    <div class="tetris-block" style="background-color: rgb(0,
    240, 0); left: 48px; top: 24px;"></div>
  </div>
```

Responsible Code

- rebuildPiece — tetris.js:432
- createBlock — tetris.js:469
- rebuildPiece — tetris.js:432
- createBlock — tetris.js:469
- rebuildPiece — tetris.js:432
- createBlock — tetris.js:469
- rebuildPiece — tetris.js:432
- cycle — tetris.js:160
- updateNextPiece — tetris.js:392
- createBlock — tetris.js:469
- rebuildPiece — tetris.js:432
- createBlock — tetris.js:469
- rebuildPiece — tetris.js:432
- createBlock — tetris.js:469
- rebuildPiece — tetris.js:432
- createBlock — tetris.js:469
- rebuildPiece — tetris.js:432

```
            var py =
        case 3:
            var py =
    }
    return py;
}

function createBlock
    var div = dc("di
    div.className =
    div.style.backgr
//  div.style.border

    div.style.left =
    div.style.top =

    return div;
}

function serialize()
    var fieldString
    for (var y=0;y<h
        for (var x=0
            fieldStr
        }
    }
    fieldString += "

    var pieceString
```

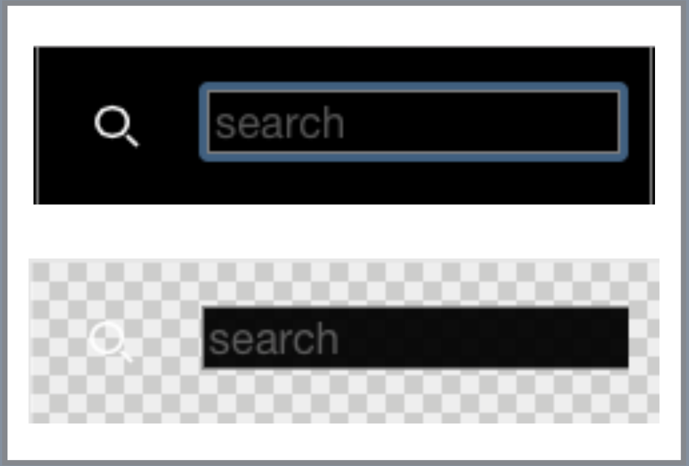
1. Instrument and record mutation operations.

2. Build a dependency graph for operations between the pre-state and post-state.

3. Based on change summary, find an *equivalent mutation operation* to explain the change.

4. Return *equivalent operation + dependencies*

Technical Challenges



Visual containment, stacking

Software vs hardware rendering

Unstable DOM element identities

React

Megamorphic call sites in library code

jQuery
write less, do more.

Pruning ineffective styles and attributes

Summary

Feature Location via Visual States

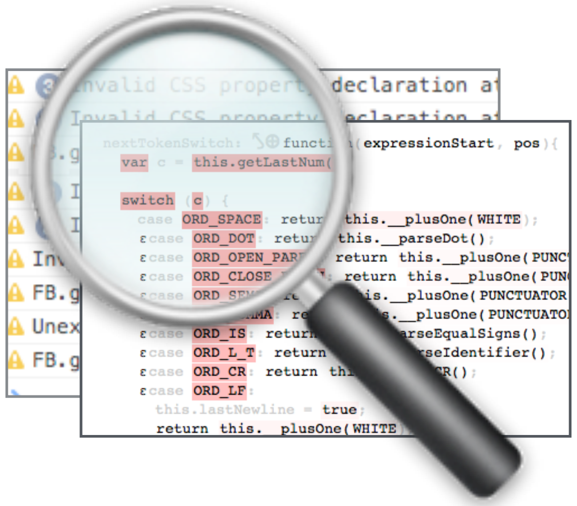
States can be automatically captured when drawing.

Juxtapose Captured Inputs and Outputs

State and output snapshots help explain each other.

Diff Markers Filter Relevant Operations

Slicing algorithms can show responsible operations.



Explaining Visual Changes in Web Interfaces

Brian Burg, Andrew J Ko, Michael Ernst

University of Washington