

**Are You Smarter  
Than A Web Browser?**





```
<html>
<head>
  <link rel="stylesheet" href="main.css" />
  <script>
    const file = window.innerWidth < 1000 ?
      'cats-mobile.css' :
      'cats-desktop.css';
    document.write(`<link rel="stylesheet" href="${file}">`);
  </script>
</head>
<body>
  <h1>Best site on the interwebz!</h1>
</body>
</html>
```



```
<html>
<head>
  <link rel="stylesheet" href="main.css" />
  <script>
    const file = window.innerWidth < 1000 ?
      'cats-mobile.css' :
      'cats-desktop.css';
    document.write(`<link rel="stylesheet" href="${file}">`);
  </script>
</head>
<body>
  <h1>Best site on the interwebz!</h1>
</body>
</html>
```



```
<html>
<head>
  <link rel="stylesheet" href="main.css" />
  <script>
    const file = window.innerWidth < 1000 ?
      'cats-mobile.css' :
      'cats-desktop.css';
    document.write(`<link rel="stylesheet" href="${file}">`);
  </script>
</head>
<body>
  <h1>Best site on the interwebz!</h1>
</body>
</html>
```







```
<html>
<head>
  <link rel="stylesheet" href="main.css" />
  <link rel="stylesheet" href="cats.css" />
</head>
<body>
  <h1>Best site on the interwebz!</h1>
</body>
</html>
```

.....



• • • • • • • • •

Best site on the interwebz!



sequentially

F12 DOM Explorer Console Debugger Network UI Responsiveness Profiler Memory Edge ? X

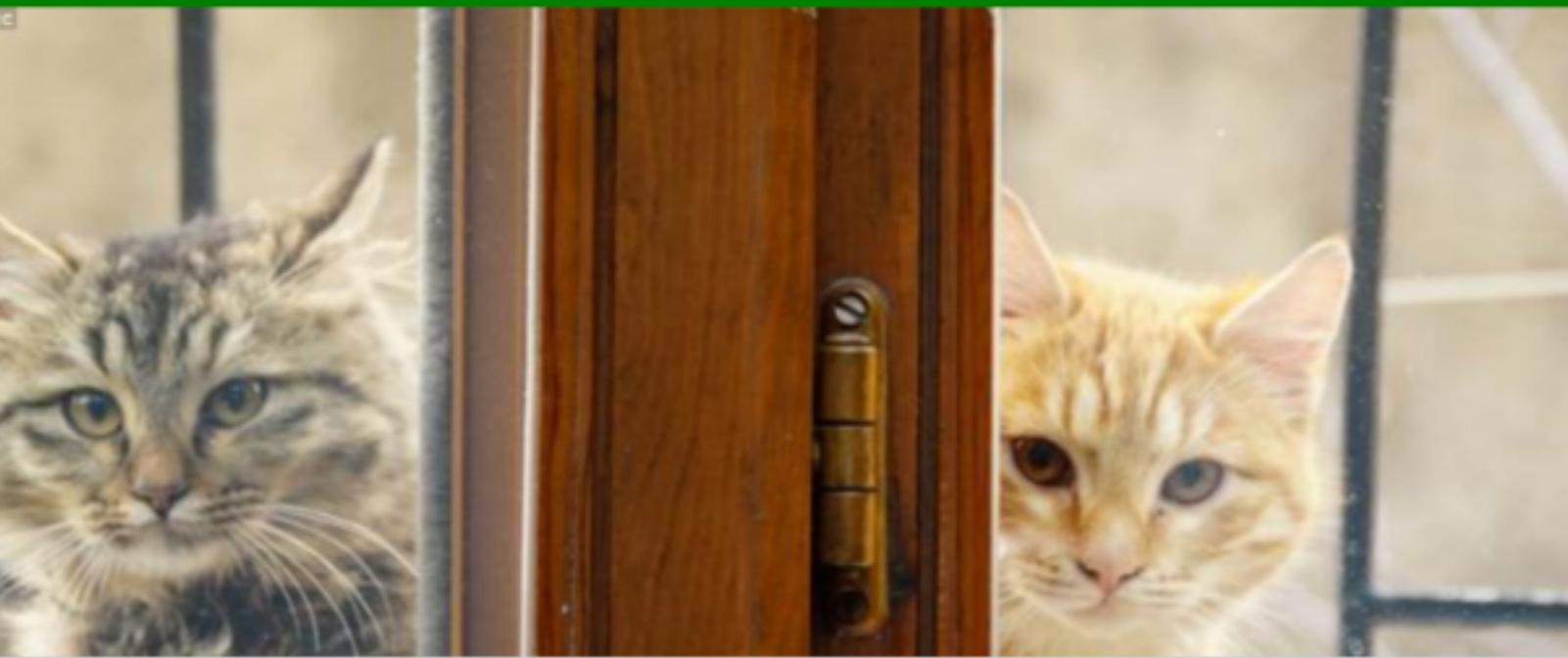
SUMMARY DETAILS

URL	Proto...	M...	Re...	Type	Rec...	Tak...	Initiator	Timings
https://smarter.yona...	HTTPS	GET	200	text/html	0.7...	171...	refresh	<div style="width: 10px; background-color: orange;"></div>
https://polyfill.io/v3/...	HTTPS	GET	200	text/java...	144...	79 ...	<script>	<div style="width: 10px; background-color: orange;"></div>
/waterfall/static/mai...	HTTPS	GET	200	text/css	332...	2.1...	<link r...	<div style="width: 10px; background-color: orange;"></div>
/waterfall/static/cats....	HTTPS	GET	200	text/css	338...	2.1...	<link r...	<div style="width: 10px; background-color: orange;"></div>
https://loremflickr.co...	HTTPS	GET	302	text/html	0.7...	421...	<img>	<div style="width: 10px; background-color: orange;"></div>
https://loremflickr.co...	HTTPS	GET	200	image/jp...	22....	344...	<img>	<div style="width: 10px; background-color: orange;"></div>

Items: 6 Sent: 2.21 KB (2,262 bytes) Received: 169.66 KB (173,728 bytes)

1:26 PM

Best site on the interwebz!

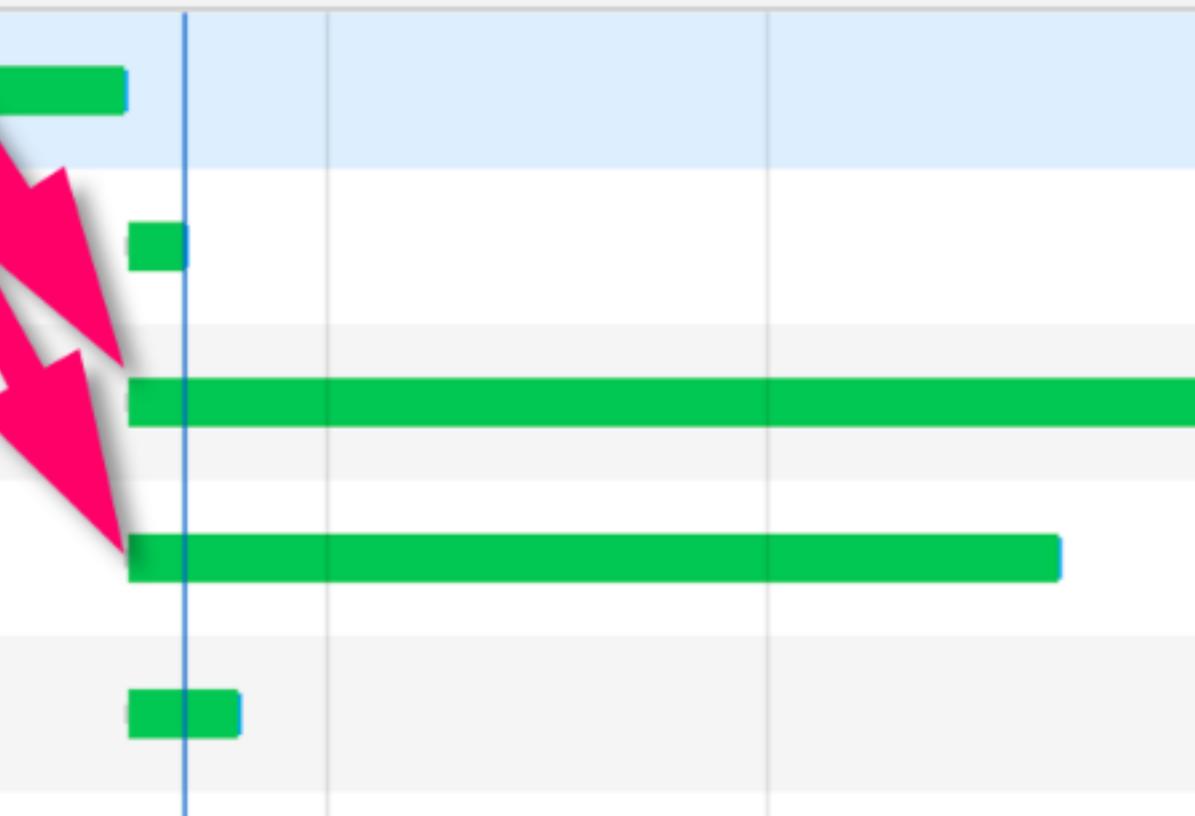
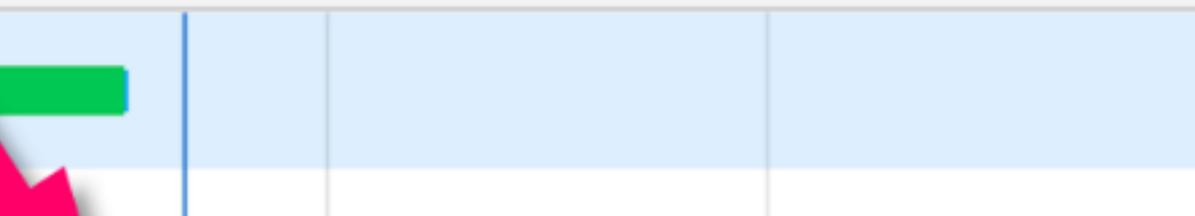
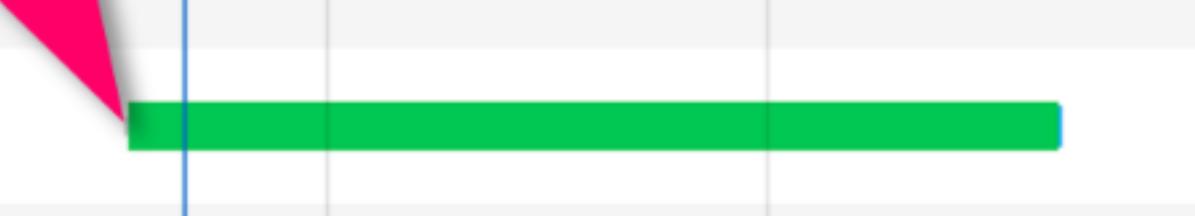
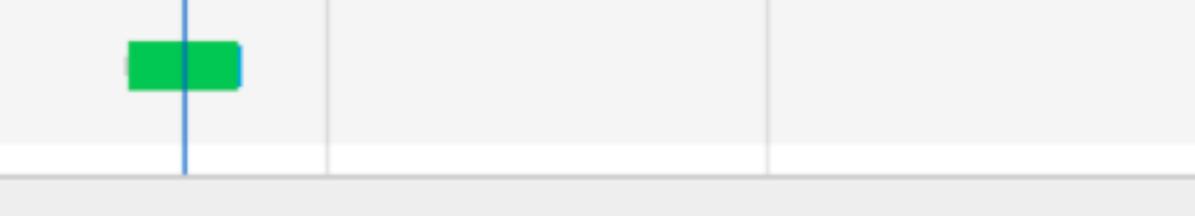


parallel

Elements Console Sources Network Performance Memory Application Security Audit

View: Group by frame Preserve log Disable cache Offline No throttling

Filter Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other

Name	Status	Type	Initiator	Size	Time	Waterfall
static/ /waterfall	200 OK	document	Other	796 B 490 B	503 ms 502 ms	
polyfill.js polyfill.io/v3	200	script	(index) Parser	240 B 224 B	136 ms 136 ms	
main.css?slow=2000 /waterfall/static	200 OK	stylesheet	(index) Parser	332 B 30 B	2.51 s 2.51 s	
cats.css?slow=2000 /waterfall/static	200 OK	stylesheet	(index) Parser	338 B 36 B	2.15 s 2.15 s	
480?random=1 loremflickr.com/640	302	text/html	(index) Parser	108 B 0 B	260 ms 259 ms	

6 requests | 43.1 KB transferred | 41.9 KB resources | Finish: 3.02 s | DOMContentLoaded: 648 ms | Load: 3.02 s

# HTMLPreloadScanner



```
<html>
<head>
  <link rel="stylesheet" href="main.css" />
  <script>
    const file = window.innerWidth < 1000 ?
      'cats-mobile.css' :
      'cats-desktop.css';
    document.write(`<link rel="stylesheet" href="${file}">`);
  </script>
</head>
<body>
  <h1>Best site on the interwebz!</h1>
</body>
</html>
```



```
<html>
<head>
  <link rel="stylesheet" href="main.css" />
  <link rel="stylesheet" href="cats-mobile.css"
        media="screen and (max-width: 1000px)">
  <link rel="stylesheet" href="cats-desktop.css"
        media="screen and (min-width: 1000px)">
</head>
<body>
  <h1>Best site on the interwebz!</h1>
</body>
</html>
```



```
<html>
<head>
  <link rel="stylesheet" href="main.css" />
  <link rel="stylesheet" href="cats-mobile.css"
        media="screen and (max-width: 1000px)">
  <link rel="stylesheet" href="cats-desktop.css"
        media="screen and (min-width: 1000px)">
</head>
<body>
  <h1>Best site on the interwebz!</h1>
</body>
</html>
```



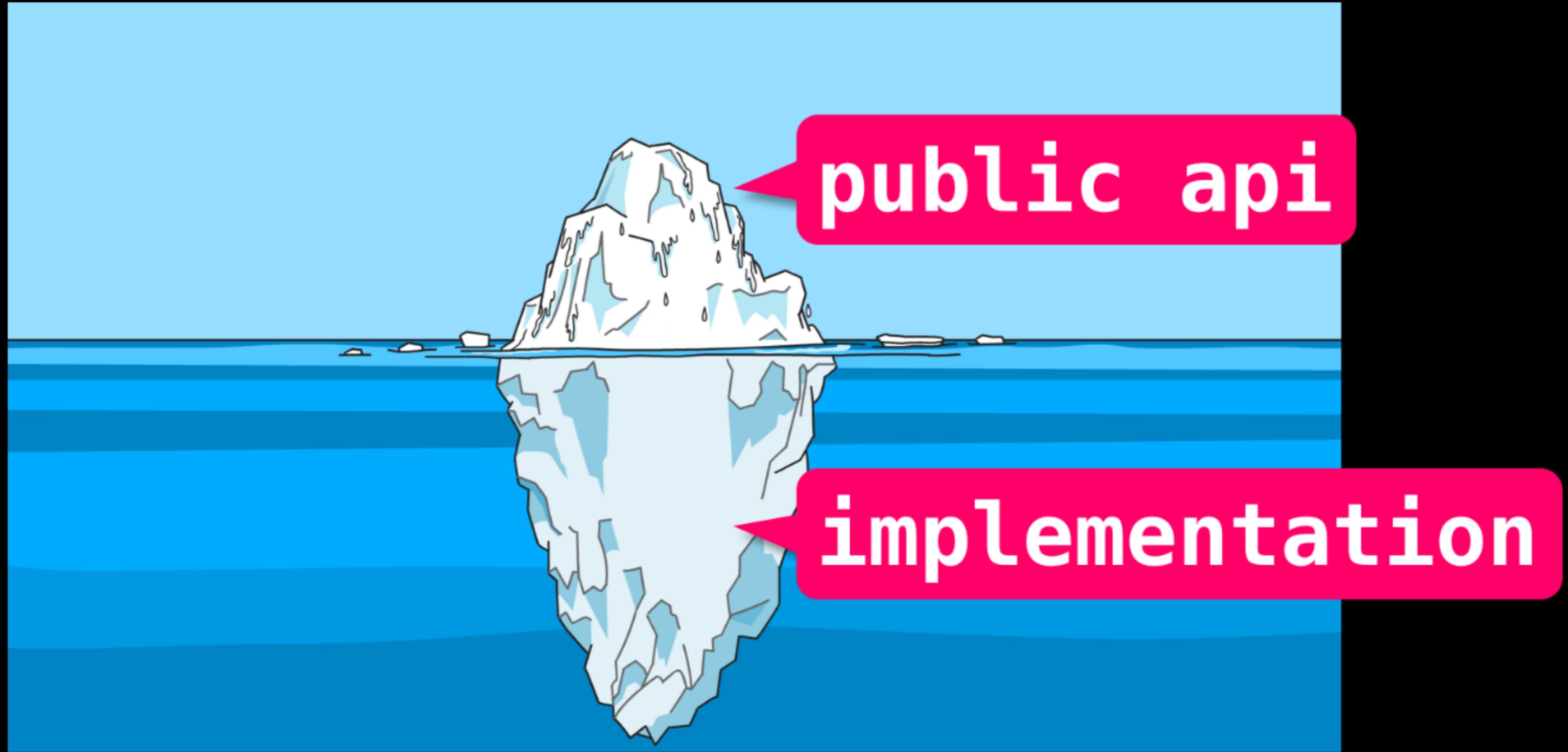
```
<html>
<head>
  <link rel="stylesheet" href="main.css" />
  <link rel="stylesheet" href="cats-mobile.css"
        media="screen and (max-width: 1000px)">
  <link rel="stylesheet" href="cats-desktop.css"
        media="screen and (min-width: 1000px)">
</head>
<body>
  <h1>Best site on the interwebz!</h1>
</body>
</html>
```





```
<link rel="preload" href="cats.css" as="style" />
```









# Lazy Loading Images

```















```



# Lazy Loading Images

```
document.addEventListener('DOMContentLoaded', () => {
  document
    .getElementById('container')
    .addEventListener('wheel', onViewportMove);

  onViewportMove();

  function onViewportMove() {
    const imgs = Array.from(
      document.querySelectorAll('.lazy-load')
    );
    imgs
      .filter(isInViewport)
      .forEach(img => {
        const src = img.getAttribute('data-src');
        img.setAttribute('src', src);
        img.classList.remove('lazy-load');
        img.classList.add('lazy-loaded');
      });
  }

  function isInViewport(element) {
    const rect = element.getBoundingClientRect();
    return (
      rect.bottom > 0 &&
      rect.top < window.innerHeight
    );
  }
});
```



# Lazy Loading Images

```
document.addEventListener('DOMContentLoaded', () => {
  document
    .getElementById('container')
    .addEventListener('wheel', onViewportMove);

  onViewportMove();

  function onViewportMove() {
    const imgs = Array.from(
      document.querySelectorAll('.lazy-load')
    );
    imgs
      .filter(isInViewport)
      .forEach(img => {
        const src = img.getAttribute('data-src');
        img.setAttribute('src', src);
        img.classList.remove('lazy-load');
        img.classList.add('lazy-loaded');
      });
    .....
  }
});
```

# Lazy Loading Images

```
document.addEventListener('DOMContentLoaded', () => {
  document
    .getElementById('container')
    .addEventListener('wheel', onViewportMove);

  onViewportMove();

  function onViewportMove() {
    const imgs = Array.from(
      document.querySelectorAll('.lazy-load')
    );
    imgs
      .filter(isInViewport)
      .forEach(img => {
        const src = img.getAttribute('data-src');
        img.setAttribute('src', src);
        img.classList.remove('lazy-load');
        img.classList.add('lazy-loaded');
      });
    .....
  }
});
```

# Lazy Loading Images

```
onViewportMove();

function onViewportMove() {
  const imgs = Array.from(
    document.querySelectorAll('.lazy-load')
  );
  imgs
    .filter(isInViewport)
    .forEach(img => {
      const src = img.getAttribute('data-src');
      img.setAttribute('src', src);
      img.classList.remove('lazy-load');
      img.classList.add('lazy-loaded');
    });
}

function isInViewport(element) {
  const rect = element.getBoundingClientRect():
  .....
}
```

# Lazy Loading Images

```
document.addEventListener('DOMContentLoaded', () => {
  document
    .getElementById('container')
    .addEventListener('wheel', onViewportMove);

  onViewportMove();

  function onViewportMove() {
    const imgs = Array.from(
      document.querySelectorAll('.lazy-load')
    );
    imgs
      .filter(isInViewport)
      .forEach(img => {
        const src = img.getAttribute('data-src');
        img.setAttribute('src', src);
        img.classList.remove('lazy-load');
        img.classList.add('lazy-loaded');
      });
    .....
  }
});
```

# Lazy Loading Images

```
document.addEventListener('wheel', onViewportMove);

onViewportMove();

function onViewportMove() {
  const imgs = Array.from(
    document.querySelectorAll('.lazy-load')
  );
  imgs
    .filter(isInViewport)
    .forEach(img => {
      const src = img.getAttribute('data-src');
      img.setAttribute('src', src);
      img.classList.remove('lazy-load');
      img.classList.add('lazy-loaded');
    });
}

function isInViewport(element) {
  const rect = element.getBoundingClientRect();
  return (
    rect.top <= window.innerHeight / 2
    && rect.left <= window.innerWidth / 2
    && rect.bottom >= 0
    && rect.right >= 0
  );
}
```

# Lazy Loading Images

```
  imgs
    .filter(isInViewport)
    .forEach(img => {
      const src = img.getAttribute('data-src');
      img.setAttribute('src', src);
      img.classList.remove('lazy-load');
      img.classList.add('lazy-loaded');
    });
}

function isInViewport(element) {
  const rect = element.getBoundingClientRect();
  return (
    rect.bottom > 0 &&
    rect.top < window.innerHeight
  );
}
});
```

.....

# Lazy Loading Images

```
onviewportMove();  
  
function onViewportMove() {  
  const imgs = Array.from(  
    document.querySelectorAll('.lazy-load')  
  );  
  imgs  
    .filter(isInViewport)  
    .forEach(img => {  
      const src = img.getAttribute('data-src');  
      img.setAttribute('src', src);  
      img.classList.remove('lazy-load');  
      img.classList.add('lazy-loaded');  
    });  
}  
  
function isInViewport(element) {  
  const rect = element.getBoundingClientRect();  
  return (  
    ....  
  )
```

# Lazy Loading Images

```
const imgs = Array.from(  
  document.querySelectorAll('.lazy-load')  
);  
imgs  
  .filter(isInViewport)  
  .forEach(img => {  
    const src = img.getAttribute('data-src');  
    img.setAttribute('src', src);  
    img.classList.remove('lazy-load');  
    img.classList.add('lazy-loaded');  
  });  
  
function isInViewport(element) {  
  const rect = element.getBoundingClientRect();  
  return (  
    rect.bottom > 0 &&  
    rect.top < window.innerHeight  
  );  
}
```

.....



# Lazy Loading Images

```
document.addEventListener('DOMContentLoaded', () => {
  document
    .getElementById('container')
-   .addEventListener('wheel', onViewportMove);
+   .addEventListener('wheel', debounce(onViewportMove, 50));

  onViewportMove();
  ....
```



# Lazy Loading Images

```
document.addEventListener('DOMContentLoaded', () => {  
  document  
    .getElementById('container')  
-   .addEventListener('wheel', onViewportMove);  
+   .addEventListener('wheel', onViewportMove, {  
+     passive: true  
+   });  
  
  onViewportMove();  
  ....
```



**Addy Osmani**

@addyoosmani

Engineering Manager working on  
@GoogleChrome • Husband & Dad • Make  
the web fast ⚡ • Lead for speed  
@ChromiumDev

⌚ Mountain View, CA

🔗 addyoosmani.com

📅 Joined April 2009

**Addy Osmani**

@addyoosmani

[Follow](#)

▼

Native <img> lazy-loading is coming to the web! [<img](https://bit.ly/loading-attrib...)  
loading=lazy> defers offscreen images until the user scrolls near them. Shipping in Chrome ~75 [<img>](https://bit.ly/loading-i2s)

## NATIVE IMAGE LAZY-LOADING

```
<!-- Lazy-load an offscreen image when a users scroll near it -->


<!-- Load an image right away instead of lazy-loading -->


<!-- Lazy-load images in <picture>. <img> drives loading -->
<picture>
  <source media="(min-width: 40em)"
         srcset="big.jpg 1x, big-hd.jpg 2x">
  <source srcset="small.jpg 1x, small-hd.jpg 2x">
  
</picture>
```

11:31 PM - 6 Apr 2019

6,146 Retweets 15,315 Likes



188

6.1K

15K



**Best site on the interwebz!**



# Overriding Touch Events

```
document.addEventListener('DOMContentLoaded', () => {
  document
    .getElementById('container')
    .addEventListener('touchmove', onTouchMove);

  function onTouchMove(e) {
    if(!e.target.classList.contains('vid')) {
      return;
    }
    e.preventDefault();

    const rect = e.target.getBoundingClientRect();
    const currentY = e.changedTouches[0].clientY;
    const offsetY = currentY - rect.top;
    const ratio = offsetY / rect.height;

    const currentTime = ratio * e.target.duration;
    e.target.currentTime = currentTime;
  }
});
```

# Overriding Touch Events

```
document.addEventListener('DOMContentLoaded', () => {
  document
    .getElementById('container')
    .addEventListener('touchmove', onTouchMove);

  function onTouchMove(e) {
    if(!e.target.classList.contains('vid')) {
      return;
    }
    e.preventDefault();

    const rect = e.target.getBoundingClientRect();
    const currentY = e.changedTouches[0].clientY;
    const offsetY = currentY - rect.top;
    const ratio = offsetY / rect.height;

    const currentTime = ratio * e.target.duration;
    e.target.currentTime = currentTime;
  }
}
```

# Overriding Touch Events

```
.getFullYearByTimeContainer,
.addEventListener('touchmove', onTouchMove);

function onTouchMove(e) {
  if(!e.target.classList.contains('vid')) {
    return;
  }
  e.preventDefault();

  const rect = e.target.getBoundingClientRect();
  const currentY = e.changedTouches[0].clientY;
  const offsetY = currentY - rect.top;
  const ratio = offsetY / rect.height;

  const currentTime = ratio * e.target.duration;
  e.target.currentTime = currentTime;
}

});
```

# Overriding Touch Events

```
document.addEventListener('DOMContentLoaded', () => {
  document
    .getElementById('container')
    .addEventListener('touchmove', onTouchMove);

  function onTouchMove(e) {
    if(!e.target.classList.contains('vid')) {
      return;
    }
    e.preventDefault();

    const rect = e.target.getBoundingClientRect();
    const currentY = e.changedTouches[0].clientY;
    const offsetY = currentY - rect.top;
    const ratio = offsetY / rect.height;

    const currentTime = ratio * e.target.duration;
    e.target.currentTime = currentTime;
  }
}
```

# Overriding Touch Events

```
document.addEventListener('DOMContentLoaded', () => {
  document
    .getElementById('container')
    .addEventListener('touchmove', onTouchMove);

  function onTouchMove(e) {
    if(!e.target.classList.contains('vid')) {
      return;
    }
    e.preventDefault();

    const rect = e.target.getBoundingClientRect();
    const currentY = e.changedTouches[0].clientY;
    const offsetY = currentY - rect.top;
    const ratio = offsetY / rect.height;

    const currentTime = ratio * e.target.duration;
    e.target.currentTime = currentTime;
  }
}
```

# Overriding Touch Events

```
.getFullYearByTimeContainer,
.addEventListener('touchmove', onTouchMove);

function onTouchMove(e) {
  if(!e.target.classList.contains('vid')) {
    return;
  }
  e.preventDefault();

  const rect = e.target.getBoundingClientRect();
  const currentY = e.changedTouches[0].clientY;
  const offsetY = currentY - rect.top;
  const ratio = offsetY / rect.height;

  const currentTime = ratio * e.target.duration;
  e.target.currentTime = currentTime;
}

});
```

# Overriding Touch Events

```
.getFullYearByTimeContainer,
.addEventListener('touchmove', onTouchMove);

function onTouchMove(e) {
  if(!e.target.classList.contains('vid')) {
    return;
  }
  e.preventDefault();

  const rect = e.target.getBoundingClientRect();
  const currentY = e.changedTouches[0].clientY;
  const offsetY = currentY - rect.top;
  const ratio = offsetY / rect.height;

  const currentTime = ratio * e.target.duration;
  e.target.currentTime = currentTime;
}

});
```



# Overriding Touch Events

```
document.addEventListener('DOMContentLoaded', () => {
-  document
-    .getElementById('container')
-    .addEventListener('touchmove', onTouchMove);
+  const vids = Array.from(document.querySelectorAll('.vid'));
+  vids.forEach(vid => {
+    vid.addEventListener('touchmove', onTouchMove);
+  });

  function onTouchMove(e) {
-    if(!e.target.classList.contains('vid')) {
-      return;
-    }
-    e.preventDefault();

  const rect = e.target.getBoundingClientRect();
```



# Overriding Touch Events

```
document.addEventListener('DOMContentLoaded', () => {
-  document
-    .getElementById('container')
-    .addEventListener('touchmove', onTouchMove);
+  const vids = Array.from(document.querySelectorAll('.vid'));
+  vids.forEach(vid => {
+    vid.addEventListener('touchmove', onTouchMove);
+  });

  function onTouchMove(e) {
-    if(!e.target.classList.contains('vid')) {
-      return;
-    }
-    e.preventDefault();

  const rect = e.target.getBoundingClientRect();
```



# Overriding Touch Events

```
document.addEventListener('DOMContentLoaded', () => {
-  document
-    .getElementById('container')
-    .addEventListener('touchmove', onTouchMove);
+  const vids = Array.from(document.querySelectorAll('.vid'));
+  vids.forEach(vid => {
+    vid.addEventListener('touchmove', onTouchMove);
+  });

  function onTouchMove(e) {
-    if(!e.target.classList.contains('vid')) {
-      return;
-    }
-    e.preventDefault();

  const rect = e.target.getBoundingClientRect();
```

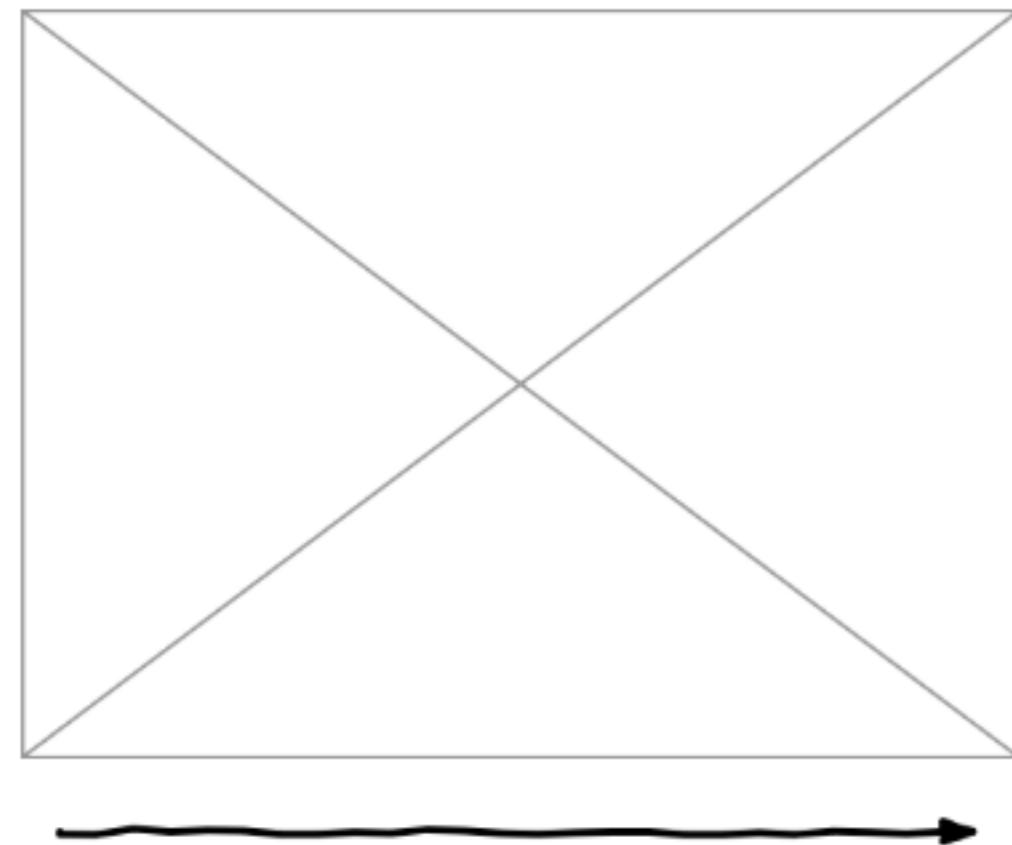






https://www.example.com

Best site on the interwebz!



# Animations

```
document.addEventListener('DOMContentLoaded', () => {
  const img = document.getElementById('img');
  let isHeadingForward = true;
  let startDate = new Date();
  const duration = 1000;
  const maxLeft = 130;

  step();

  function step() {
    const currentLoc = toNumber(img.style.left);
    const endDate = new Date();
    const ratio = (endDate - startDate) / duration;
    let nextLoc = maxLeft * ratio;
    nextLoc = isHeadingForward ? nextLoc : maxLeft - nextLoc;
    img.style.left = toPx(nextLoc);
    if(isHeadingForward && currentLoc >= maxLeft) {
      isHeadingForward = false;
      startDate = new Date();
    }
    if(!isHeadingForward && currentLoc <= 0) {
      isHeadingForward = true;
      startDate = new Date();
    }
    requestAnimationFrame(step);
  }
});

function toPx(num) {
  return `${num}px`;
}

function toNumber(px) {
  return Number(px.replace('px', ''));
```



# Animations

```
document.addEventListener('DOMContentLoaded', () => {
  const img = document.getElementById('img');
  let isHeadingForward = true;
  let startDate = new Date();
  const duration = 1000;
  const maxLeft = 130;

  step();

  function step() {
    const currentLoc = toNumber(img.style.left);
    const endDate = new Date();
    const ratio = (endDate - startDate) / duration;
    let nextLoc = maxLeft * ratio;
    nextLoc = isHeadingForward ? nextLoc : maxLeft - nextLoc;
    img.style.left = toPx(nextLoc);
    if(isHeadingForward && currentLoc >= maxLeft) {
      isHeadingForward = false;
      startDate = new Date();
      ....
```

# Animations

```
let startDate = new Date();
const duration = 1000;
const maxLeft = 130;

step();

function step() {
  const currentLoc = toNumber(img.style.left);
  const endDate = new Date();
  const ratio = (endDate - startDate) / duration;
  let nextLoc = maxLeft * ratio;
  nextLoc = isHeadingForward ? nextLoc : maxLeft - nextLoc;
  img.style.left = toPx(nextLoc);
  if(isHeadingForward && currentLoc >= maxLeft) {
    isHeadingForward = false;
    startDate = new Date();
  }
  if(!isHeadingForward && currentLoc <= 0) {
    isHeadinaForward = true;
  }
  .....
}
```

# Animations

```
const duration = 1000;
const maxLeft = 130;

step();

function step() {
  const currentLoc = toNumber(img.style.left);
  const endDate = new Date();
  const ratio = (endDate - startDate) / duration;
  let nextLoc = maxLeft * ratio;
  nextLoc = isHeadingForward ? nextLoc : maxLeft - nextLoc;
  img.style.left = toPx(nextLoc);
  if(isHeadingForward && currentLoc >= maxLeft) {
    isHeadingForward = false;
    startDate = new Date();
  }
  if(!isHeadingForward && currentLoc <= 0) {
    isHeadingForward = true;
    startDate = new Date();
  }
  ....
```

# Animations

```
step();  
  
function step() {  
    const currentLoc = toNumber(img.style.left);  
    const endDate = new Date();  
    const ratio = (endDate - startDate) / duration;  
    let nextLoc = maxLeft * ratio;  
    nextLoc = isHeadingForward ? nextLoc : maxLeft - nextLoc;  
    img.style.left = toPx(nextLoc);  
    if(isHeadingForward && currentLoc >= maxLeft) {  
        isHeadingForward = false;  
        startDate = new Date();  
    }  
    if(!isHeadingForward && currentLoc <= 0) {  
        isHeadingForward = true;  
        startDate = new Date();  
    }  
    requestAnimationFrame(step);  
    .....  
}
```

# Animations

```
img.style.left = toPx(nextLoc);
if(isHeadingForward && currentLoc >= maxLeft) {
    isHeadingForward = false;
    startDate = new Date();
}
if(!isHeadingForward && currentLoc <= 0) {
    isHeadingForward = true;
    startDate = new Date();
}
requestAnimationFrame(step);
};

function toPx(num) {
    return `${num}px`;
}

function toNumber(px) {
    return Number(px.replace('px', ''));
```

.....



# Animations

```
#container {  
    position: relative;  
}  
  
#img {  
    position: absolute;  
    top: 0;  
    left: 0;  
    animation: move 1s linear infinite alternate;  
}  
  
@keyframes move {  
    from { left: 0; }  
    to { left: 130px; }  
}
```



# Animations

```
#container {  
    position: relative;  
}  
  
#img {  
    position: absolute;  
    top: 0;  
    left: 0;  
    animation: move 1s linear infinite alternate;  
}  
  
@keyframes move {  
    from { left: 0; }  
    to { left: 130px; }  
}
```



# Animations

```
#container {  
    position: relative;  
}  
  
#img {  
    position: absolute;  
    top: 0;  
    left: 0;  
    animation: move 1s linear infinite alternate;  
}  
  
@keyframes move {  
    from { left: 0; }  
    to { left: 130px; }  
}
```



# Animations

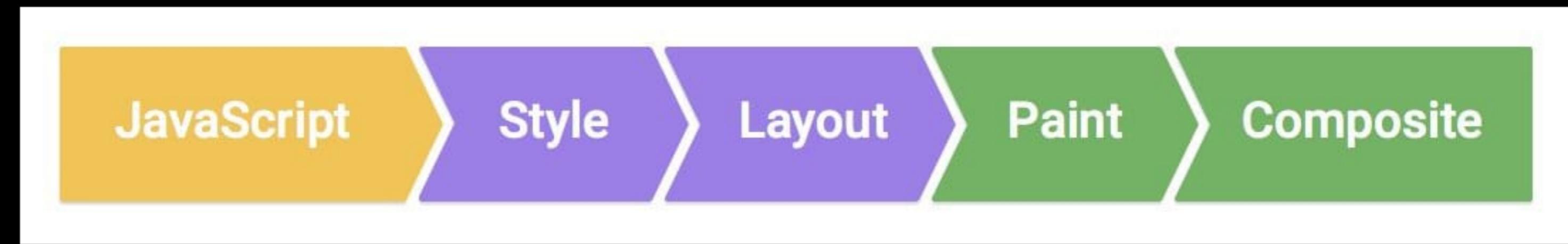
```
#container {  
    position: relative;  
}  
  
#img {  
    position: absolute;  
    top: 0;  
    left: 0;  
    animation: move 1s linear infinite alternate;  
}  
  
@keyframes move {  
    from { left: 0; }  
    to { left: 130px; }  
}
```

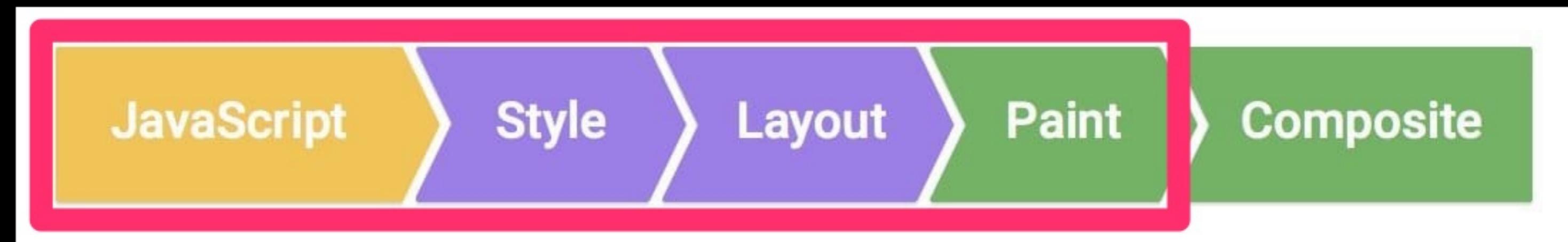




# Pixel Pipeline







● Layout ● Paint ● Composite

Blink Gecko WebKit EdgeHTML

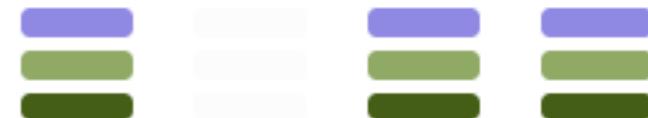
Change from default

Subsequent updates

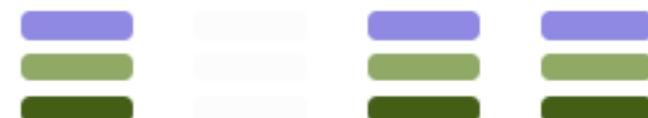
align-content



align-items



align-self



backface-visibility



background-attachment



background-blend-mode



# CSS Triggers



B G W E

B G W E

font-weight



height



justify-content



left



letter-spacing



line-height



list-style-image



list-style-position



# Animations

```
@keyframes move {  
- from { left: 0; }  
- to { left: 130px; }  
+ from { transform: translateX(0); }  
+ to { transform: translateX(130px); }  
}
```





**Slides & Demos & Code**

**footguns.yonatan.dev**





# Nolan Lawson



# Mariko Kosaka



# Andy Davies





.....

A close-up portrait of Tim Berners-Lee, a man with short, light-colored hair and a slight smile. He is wearing a dark, collared shirt.

**Tim Berners-Lee**  
Web Developer