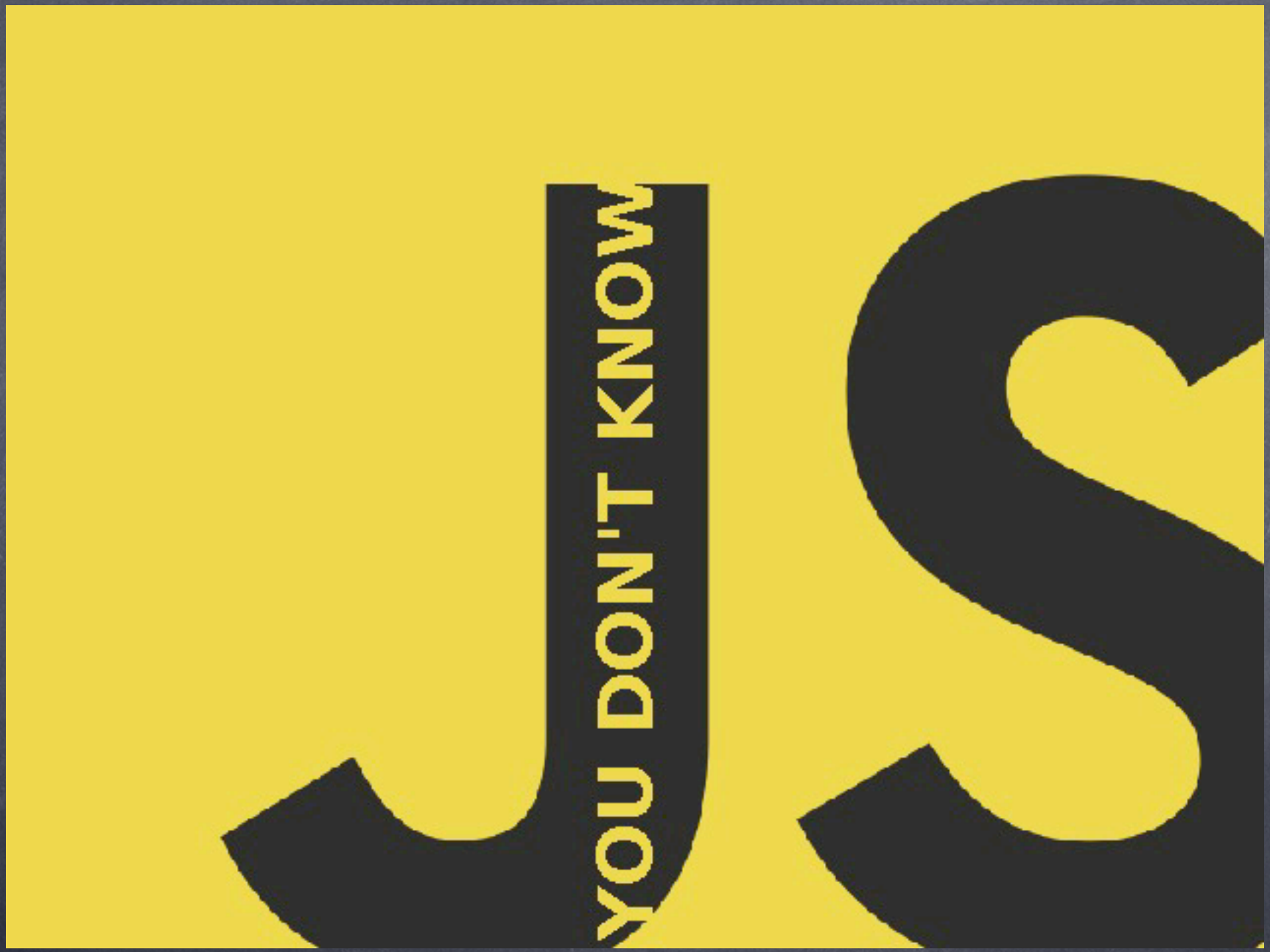
Kyle Simpson
@getify
http://getify.me

- LABjs
- grips
- h5ive
- We Puzzle It!

Kyle Simpson
@getify
http://getify.me

http://speakerdeck.com/getify



http://YouDontKnowJS.com

Real-Time: HTML5 & node.js

setup

http://python.org/download/

you “need” python

http://nodejs.org/download/

you **need** node.js

https://github.com/getify/unnamed.git

Unnamed is a game!

```
Blackbeard:unnamed getify$ git checkout step1
Note: checking out 'step1'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

  git checkout -b new_branch_name

HEAD is now at 4ccf634... rename
Blackbeard:unnamed getify$ ls -la
total 8
drwxr-xr-x   6 getify  staff  204 Jul 23 12:38 .
drwx------   7 getify  staff  238 Jul 23 12:36 ..
drwxr-xr-x  16 getify  staff  544 Jul 23 12:38 .git
-rw-r--r--   1 getify  staff   61 Jul 17 11:46 README.md
drwxr-xr-x   5 getify  staff  170 Jul 23 12:38 server
drwxr-xr-x   6 getify  staff  204 Jul 23 12:38 web
Blackbeard:unnamed getify$
```

```
Blackbeard:web getify$ python -m SimpleHTTPServer 8080
Serving HTTP on 0.0.0.0 port 8080 ...
```

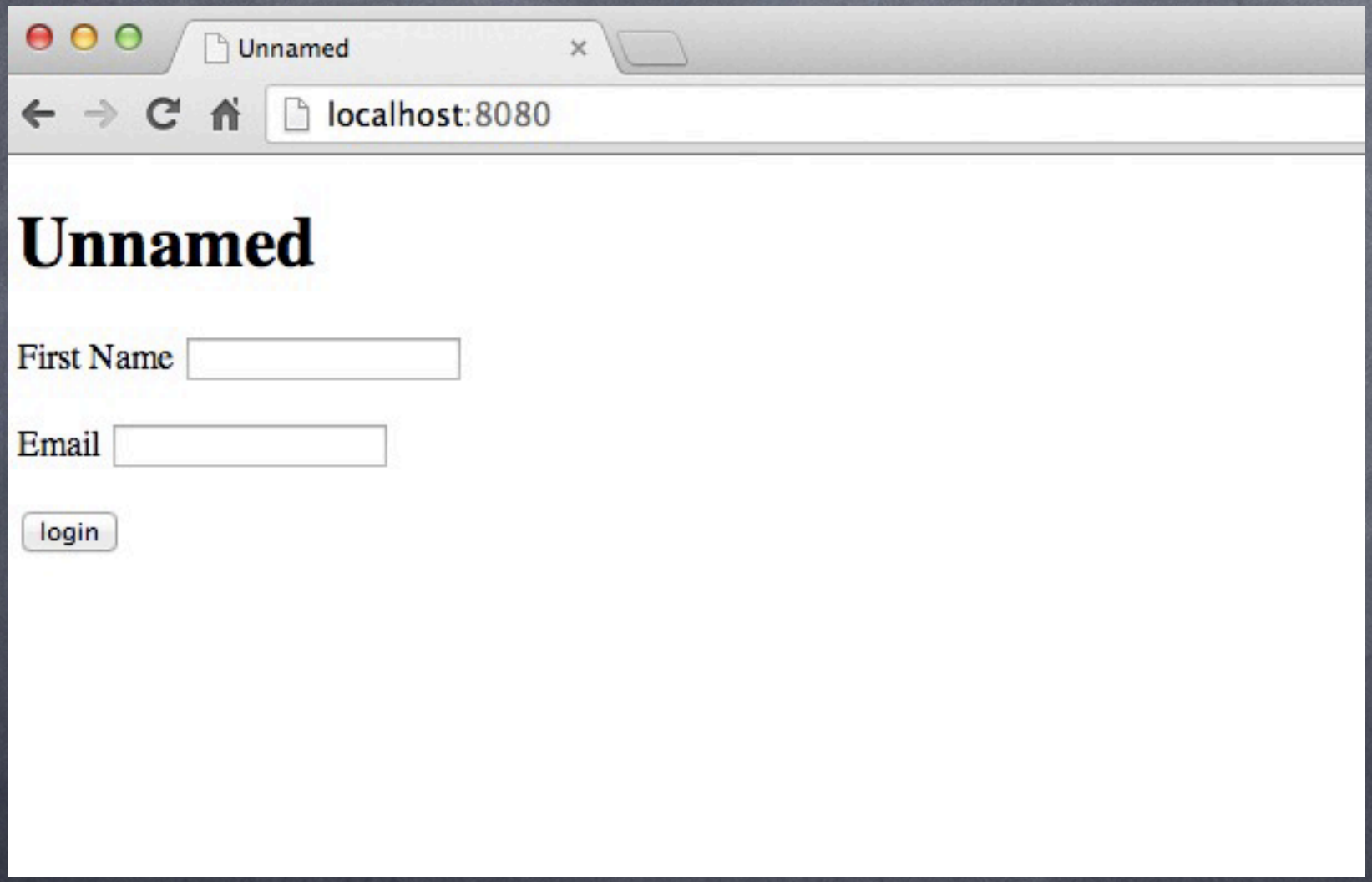
or: python3 -m http.server 8080

```
Blackbeard:server getify$ cat secret.js
exports.PUBLIC_SERVER_ADDR = "..";
exports.PUBLIC_SERVER_PORT = 0;

exports.INTERNAL_SERVER_ADDR = "127.0.0.1";
exports.INTERNAL_SERVER_PORT = 8005;

exports.CORS_GET_HEADERS = {
        "Access-Control-Allow-Origin": "http://localhost:8080",
        "Access-Control-Allow-Credentials": false,
        "Access-Control-Allow-Methods": "GET, OPTIONS",
        "Access-Control-Allow-Headers": "Content-Type, User-Agent, X-Requested-With"
};
exports.CORS_POST_HEADERS = {
        "Access-Control-Allow-Origin": "http://localhost:8080",
        "Access-Control-Allow-Credentials": false,
        "Access-Control-Allow-Methods": "POST, OPTIONS",
        "Access-Control-Allow-Headers": "Content-Type, User-Agent, X-Requested-With"
};
Blackbeard:server getify$
```

```
Blackbeard:server getify$ node server.js
```


Unnamed
localhost:8080
Unnamed
First Name
Email
login

Agenda

- HTML5
- node.js
- Web Sockets (socket.io)
- WebRTC

HTML5

facades

https://github.com/getify/h5ive

storage

```
var keep = h5.storage();
var session = h5.storage({ expires: "session" });
var temp = h5.storage({ expires: 5*60*1000 }); // 5 minutes

keep
.save({
    prefs: { /* ... */ }
});

session
.save({
    session_id: 123456,
    foo: "bar baz"
})
.discard(["foo"]);

temp.save({
    active_login: session.get("session_id") // only keep this ar
});
```

<canvas>

```
var cnv = h5.canvas({
    width: 500,
    height: 500,
    matchDimensions: true // make the CSS dimensions match the attribute dimen
});

document.body.appendChild(cnv.element());

cnv
.clear()
.setStyles({
    alpha: 0.8,
    stroke: {
        width: 3,
        color: "#fc6"
    },
    fill: {
        color: "#009"
    }
})
.startPath(75,100)
.defineSegments([
    { lineTo: [200,200] },
    { lineTo: [100,50] }
])
.endPath({
    close: true,
    stroke: true,
    fill: true
```

getUserMedia, <video>

```
h5
.userMedia({
    video: true
})
.stream(function(src){
    var video = document.getElementById("mycam");
    video.src = src;
    video.play();
})
.failed(function(){
    alert("Access to the media failed.");
});
```

requestAnimationFrame

```
var aFrame = h5.animationFrame,
    body = document.body,
    text, id1, id2, id3
;

id1 = aFrame.queue(function(){
    // canceled, won't ever get called
    text = document.createTextNode("##");
    body.appendChild(text);
});
id2 = aFrame.queueAfter(function(){
    // canceled, won't ever get called
    text = document.createTextNode("!!");
    body.appendChild(text);
});
id3 = aFrame.queueAfter(function(){
    // canceled, won't ever get called
    text = document.createTextNode("$$");
    body.appendChild(text);
});

aFrame.queueAfter(function(){
    text = document.createTextNode("third.");
    body.appendChild(text);
});
aFrame.queue(function(){
    aFrame.cancel(id3); // still time to cancel something before the
```

WebSockets

```
55          socket.on("game_session_invalid", function(data) {
56              self.postMessage({error: "game_session_invalid"});
57          });
58
59          socket.on("reset_player",function() {
60              self.postMessage({reset_player: true});
61          });
62      }
63
64      if (!game_session_started) {
65          game_session_started = true;
66          socket.emit("establish_game_session", {game_session_id: game
67      }
68  }
```

socket.io

node.js

Hello World

```
1  function handleHTTP(req,res) {
2      if (req.method == "GET") {
3          if (req.url == "/") {
4              res.writeHead(200,{ "Content-type": "text/plain" });
5              res.end("Hello World: " + Math.random());
6          }
7          else {
8              res.writeHead(403);
9              res.end();
10         }
11     }
12     else {
13         res.writeHead(403);
14         res.end();
15     }
16 }
17
18 var http = require("http"),
19     httpserv = http.createServer(handleHTTP),
20     port = 8006,
21     host = "127.0.0.1";
22
23 httpserv.listen(port, host);
```

async flow

https://github.com/getify/asynquence

```
ASQ()
.gate(
    function(done){ setTimeout(done,100); },
    function(done){ setTimeout(done,200); },
    function(done){ setTimeout(done,300); }
)
.then(function(){
    alert("All tasks are complete, and that only took ~300ms, not 600ms!");
});
```

```
ASQ()
.then(function(done){ setTimeout(done,100); })
.then(function(done){ setTimeout(done,200); })
.then(function(done){ setTimeout(done,300); })
.then(function(){
    alert("All tasks are complete, and that took ~600ms!");
});
```

```
ASQ()
.then(function(done){ setTimeout(done,100); })
.gate(
    function(done){ setTimeout(done,200); },
    function(done){ setTimeout(done,300); }
)
.then(function(){
    alert("All tasks are complete, and that took ~400ms!");
});
```

socket.io

http://socket.io/

Hello World

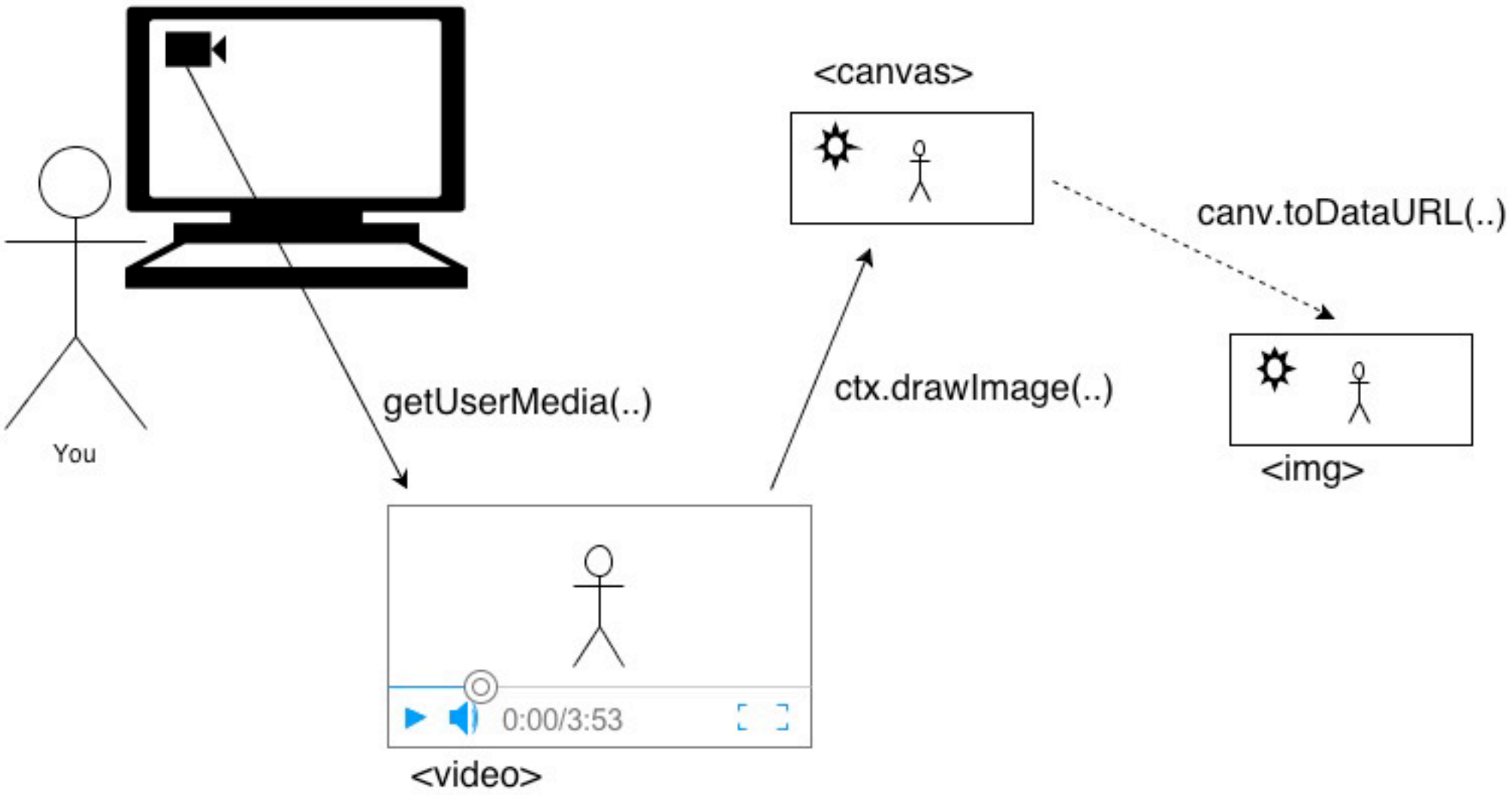
```
46      io = require("socket.io").listen(httpserv)
47  ;
48
49  // configure socket.io
50  io.configure(function(){
51      io.enable("browser client minification"); // send minif
52      io.enable("browser client etag"); // apply etag caching
53      io.set("log level", 1); // reduce logging
54      io.set("transports", [
55          "websocket",
56          "xhr-polling",
57          "jsonp-polling"
58      ]);
59  });
60
61  httpserv.listen(port, host);
62
63  io.on("connection",connection);
```

```
21  function connection(socket) {
22
23      function disconnect() {
24          console.log("disconnected");
25      }
26
27      socket.on("disconnect",disconnect);
28
29      var intv = setInterval(function(){
30          socket.emit("hello",Math.random());
31      },1000);
32  }
```

WebRTC

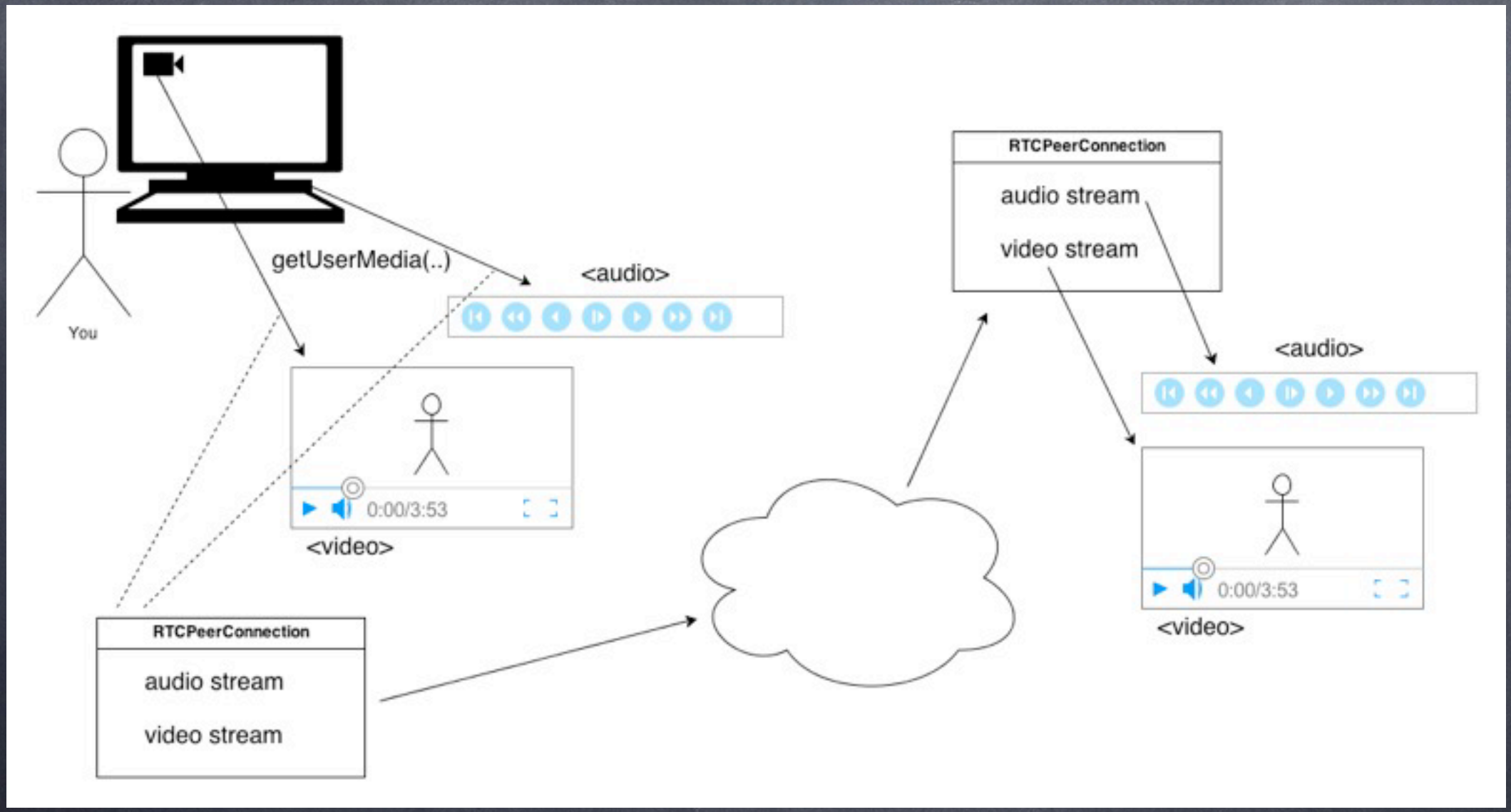
http://www.webrtc.org/

webcam capture

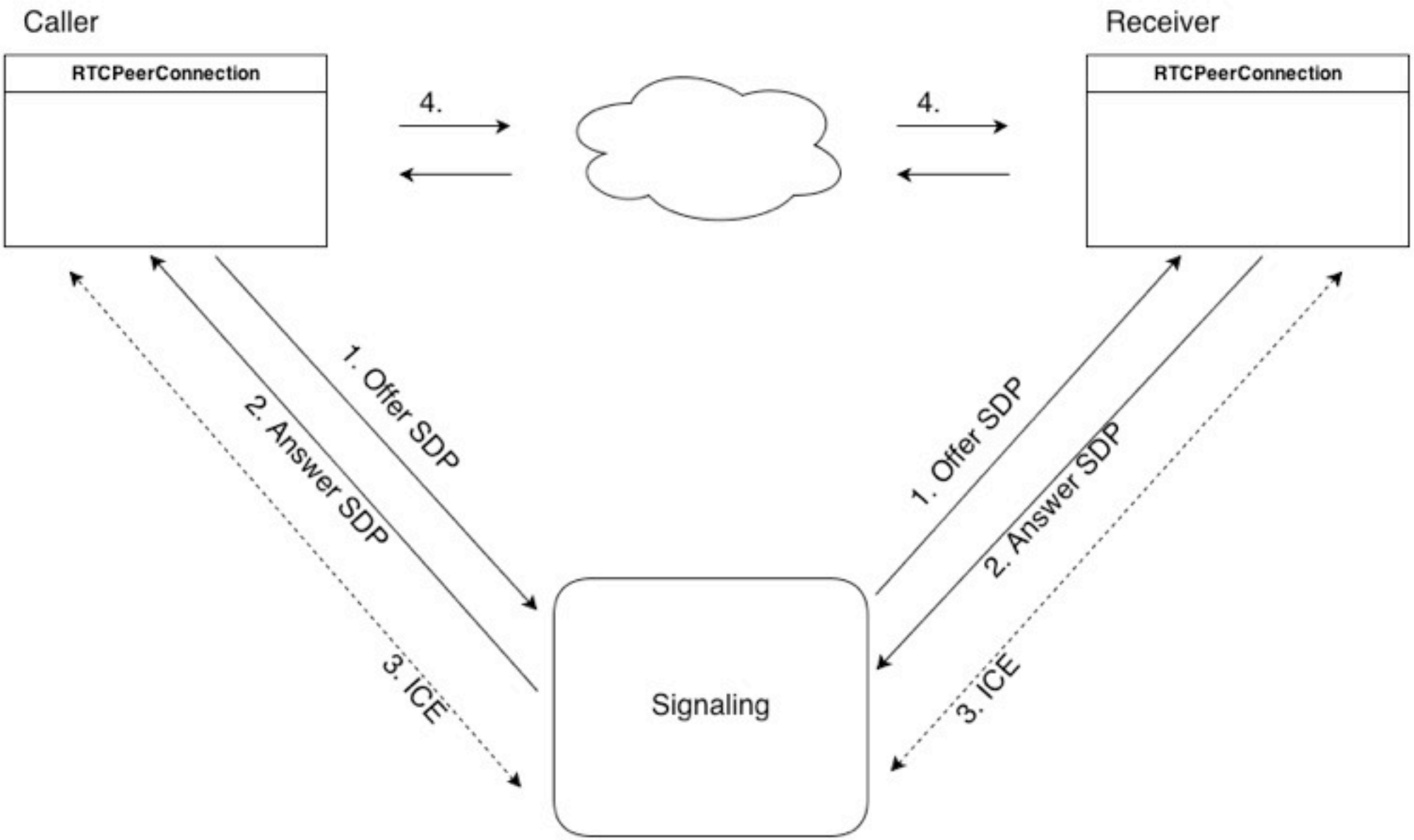

You
getUserMedia(..)
0:00/3:53
<video>
ctx.drawImage(..)
<canvas>
canv.toDataURL(..)
<img>

RTCPeerConnection

Video, Audio


You
getUserMedia(..)
<audio>
0:00/3:53
<video>
RTCPeerConnection
audio stream
video stream
RTCPeerConnection
audio stream
video stream
<audio>
0:00/3:53
<video>

Signaling


Caller
Receiver
RTCPeerConnection
RTCPeerConnection
4.
4.
1. Offer SDP
2. Answer SDP
3. ICE
1. Offer SDP
2. Answer SDP
3. ICE
Signaling

DataChannel

Caller
RTCPeerConnection
audio stream
video stream
DataChannel
Receiver
RTCPeerConnection
audio stream
video stream
DataChannel

```
20  // rtc stuff
21  function createPeerConnection(config,optional) {
22      if (global.RTCPeerConnection) return new RTCPeerConnection(config,opt
23      else if (global.webkitRTCPeerConnection) return new webkitRTCPeerConn
24      else if (global.mozRTCPeerConnection) return new mozRTCPeerConnection
25      throw new Error("RTC Peer Connection not available");
26  }
27
28  function createIceCandidate(candidate) {
29      if (global.RTCIceCandidate) return new RTCIceCandidate(candidate);
30      else if (global.webkitRTCIceCandidate) return new webkitRTCIceCandida
31      else if (global.mozRTCIceCandidate) return new mozRTCIceCandidate(can
32      throw new Error("RTC Ice Candidate not available");
33  }
34
35  function createSessionDescription(desc) {
36      if (global.RTCSessionDescription) return new RTCSessionDescription(de
37      else if (global.webkitRTCSessionDescription) return new webkitRTCSess
38      else if (global.mozRTCSessionDescription) return new mozRTCSessionDes
39      throw new Error("RTC Session Description not available");
40  }
41
42  function signal(message) {
43      if (socket) {
44          socket.emit("message",message);
45      }
46  }
```

unnamed

Take a deep breath

Kyle Simpson
@getify
http://getify.me

Thanks!

Questions?