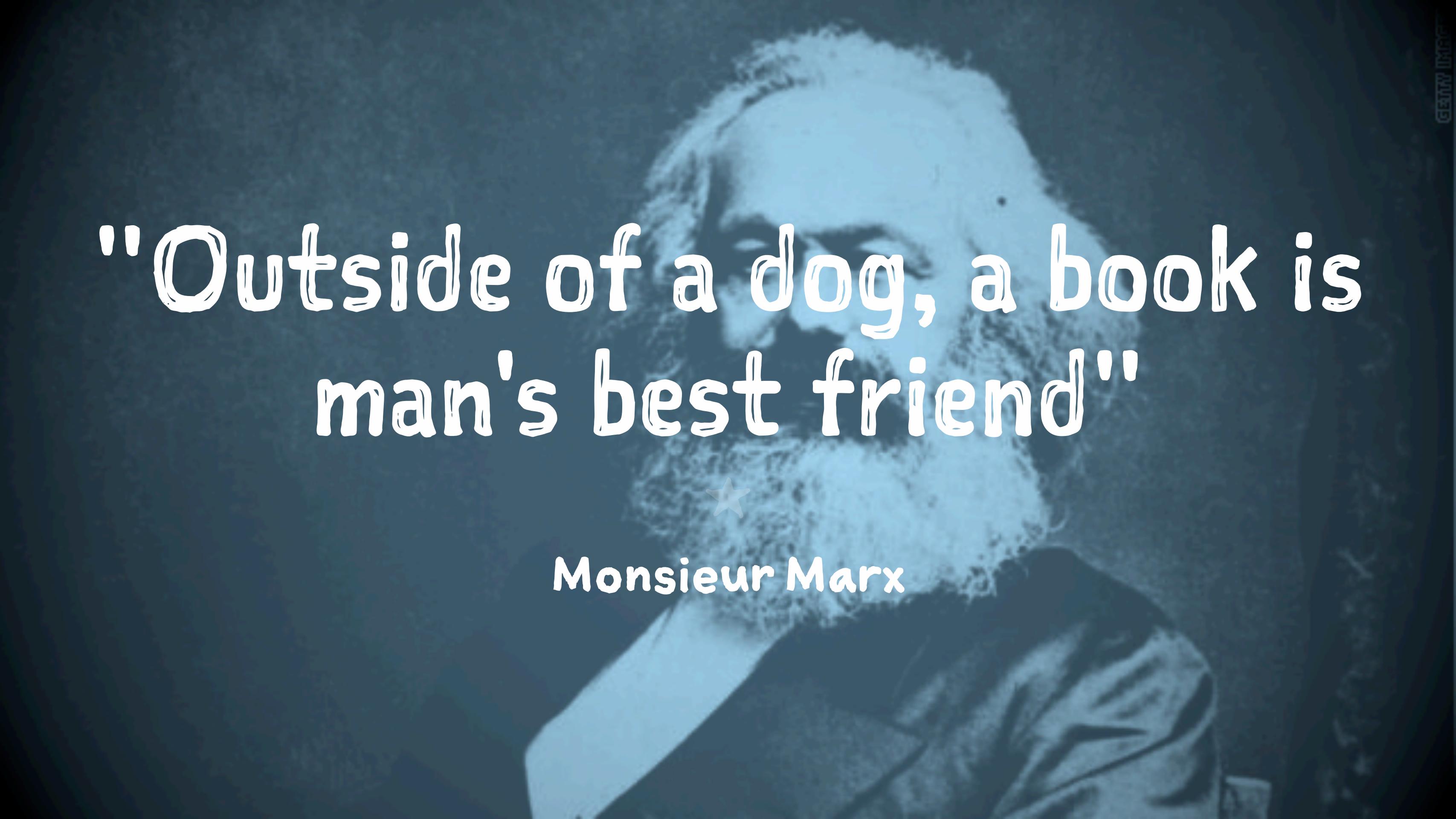


Message in a Browser

I like books



"Outside of a dog, a book is
man's best friend"

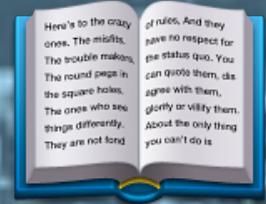
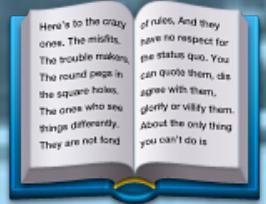


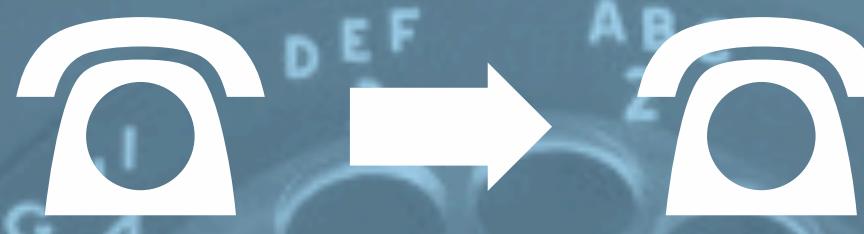
Monsieur Marx



Let's make a better book

paper is paper





phone is phone

device is multiplexer



"The modality of medium is
the circumflex of loci"

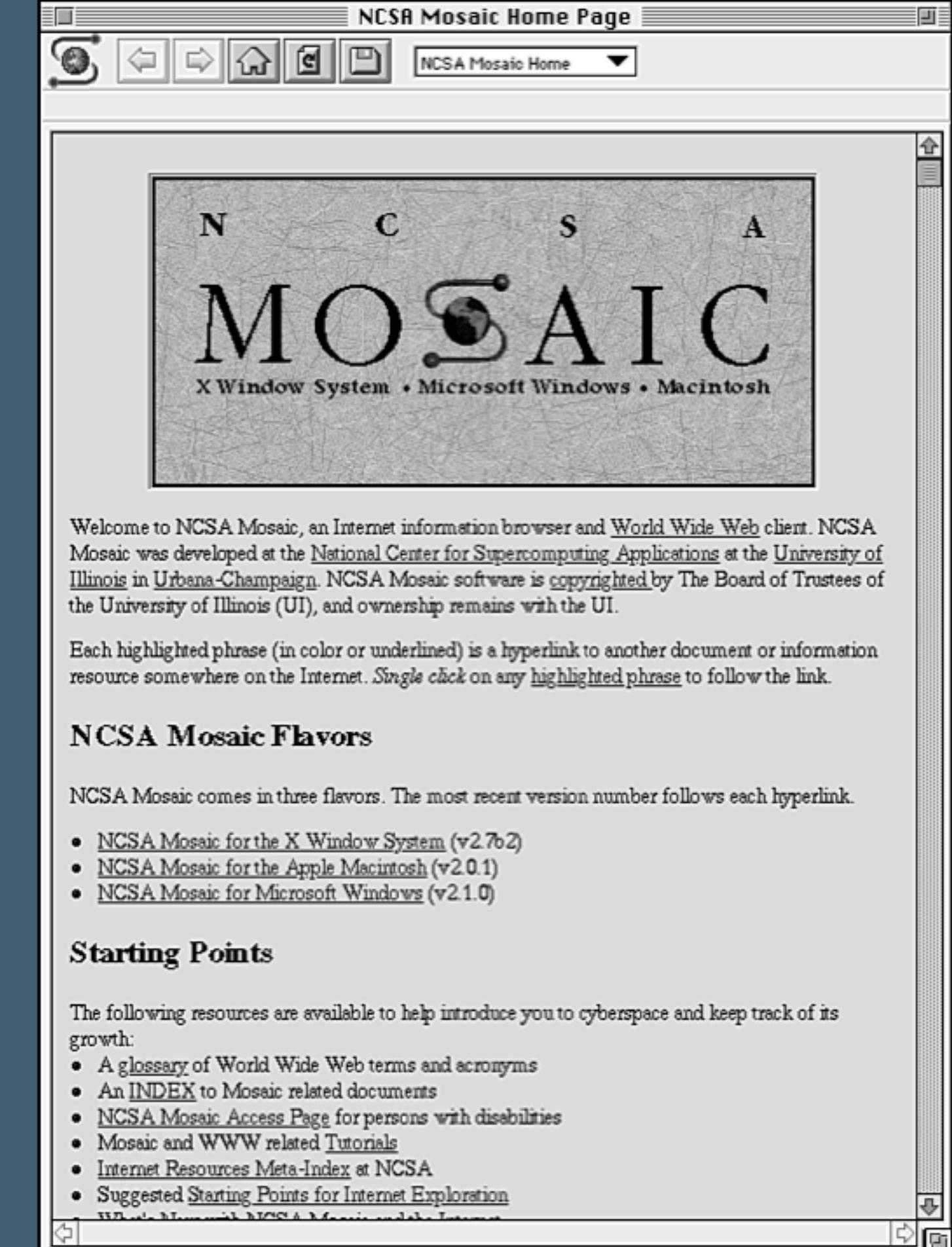


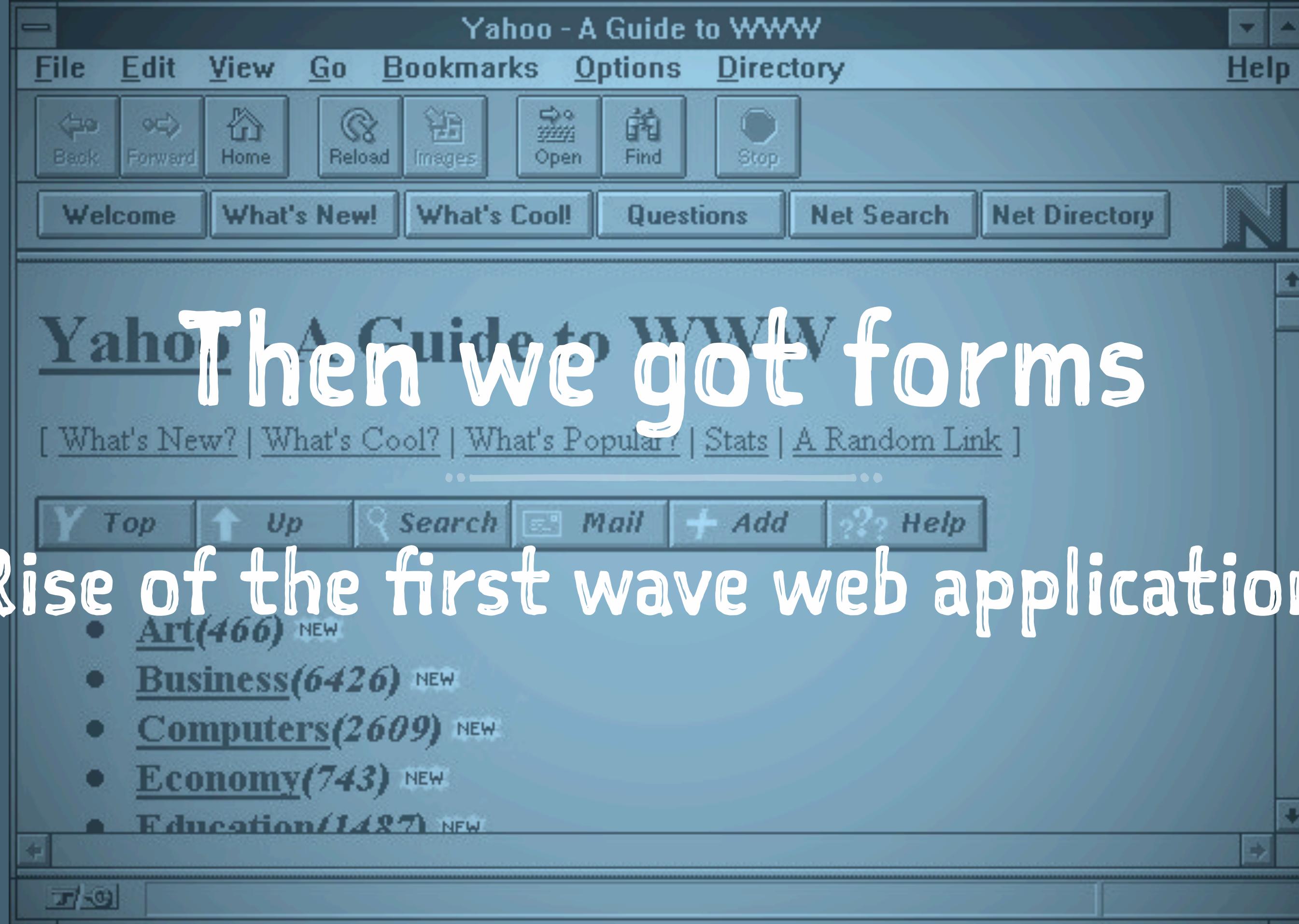
Adele Dazeem

Once upon a time

The web was documents

Internet + hypertext = ❤



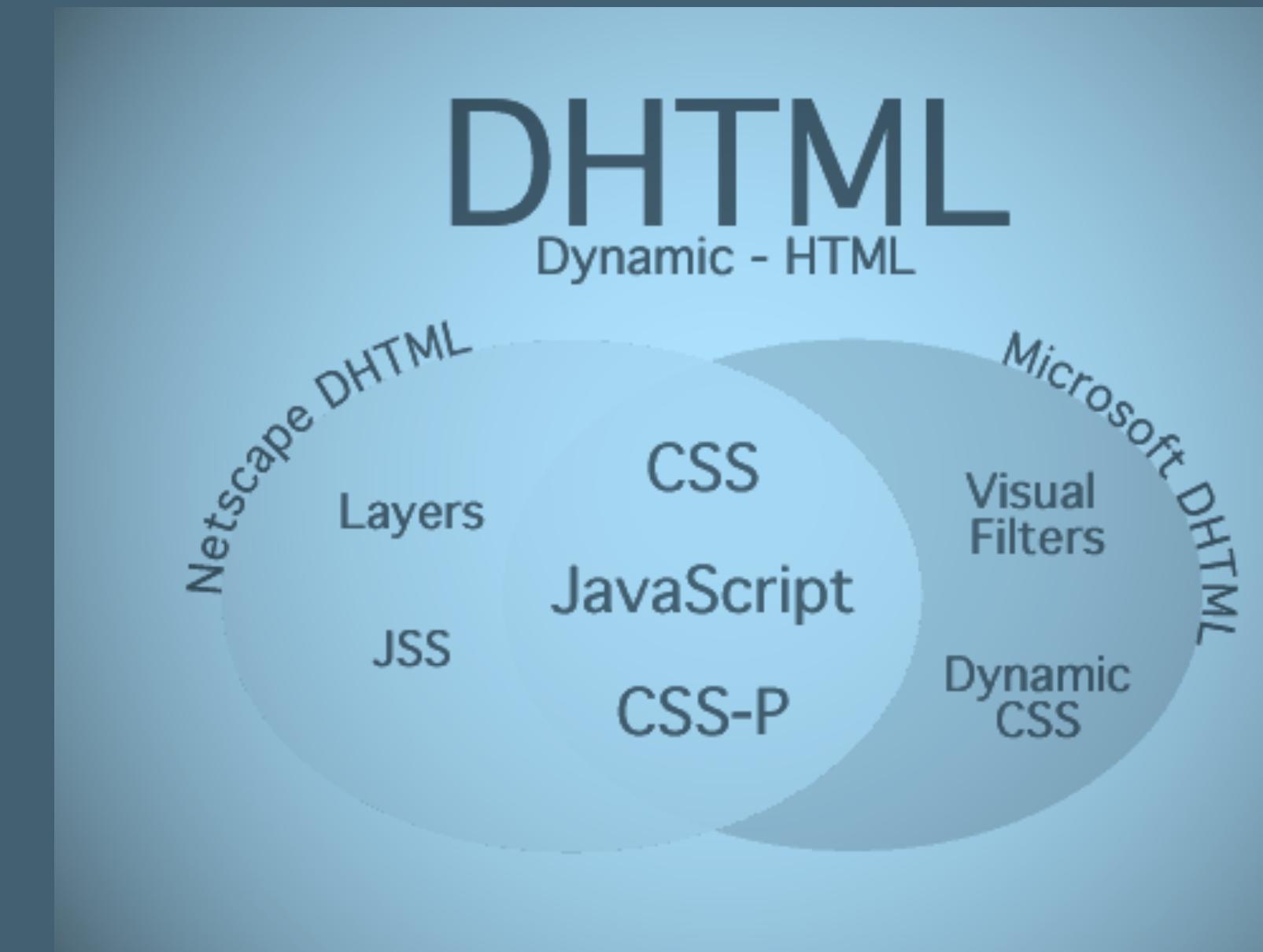


Rise of the first wave web applications

- [Art\(466\)](#) NEW
- [Business\(6426\)](#) NEW
- [Computers\(2609\)](#) NEW
- [Economy\(743\)](#) NEW
- [Education\(1487\)](#) NEW

JS IS BORN

DHTML ruins everything



JS IS REBORN

Second wave web apps

AKA SPAs

Data Store(s)

RPC or
Websocket

Middleware

Lib(s)

HTML

JSON

JSON

Native Mob

Desktop

Web App



THE WEB WAS:
hypertext documents
& form-driven apps



THE WEB IS:
single page applications
& desktop applications
& interactive games



The image is a collage of three screenshots from a video game interface, likely a mod or a specialized application for creating content. The top-left screenshot shows a 'Capture to Canvas' dialog with options for height, quality, clear, and send. The top-right screenshot shows a grid-based tool with a 'finalize' button. The bottom-left screenshot shows a 'finalizing gif done' message with options for image, quality, delay, matte, and finalize. The bottom-right screenshot shows a 'finalizing gif done' message with options for datauri, title, and caption. Overlaid on this collage in large, white, hand-drawn-style font is the text: 'THE WEB CAN BE: immersive experiences & interactive tools'.

THE WEB CAN BE:

immersive experiences

& interactive tools

WHY WEB?

Browser is OS

Web is app store

URL is app install

Interactive Experiences

inputs ->  -> outputs

Inputs

- HTTP (Page load and Ajax via XHR)
- WebSockets, SSE, WebRTC, PostMessage
- Geolocation, Orientation
- Mouse, Keyboard, Touch
- Gamepad, MIDI
- Mic, Camera

HTTP

```
var xhr = new XMLHttpRequest()

xhr.onreadystatechange = function() {
  if (xhr.readyState == 4) {
    callback(xhr.responseText)
  }
}

xhr.open('GET', target, true)
xhr.send(null)
```

WebSockets (server)

```
var io = require('socket.io').listen(80);

io.sockets.on('connection', function (socket) {
  socket.emit('news', { hello: 'world' });

  socket.on('my other event', function (data) {
    console.log(data);
  });
});
```

WebSockets (browser)

```
<script src="/socket.io/socket.io.js"></script>

<script>
  var socket = io.connect('http://localhost');

  socket.on('news', function (data) {
    console.log(data);
    socket.emit('my other event', { my: 'data' });
  });
</script>
```

WebRTC

```
var conn = peer.connect('another-peers-id');
conn.on('open', function(){
  conn.send('hi!');
});

peer.on('connection', function(conn) {
  conn.on('data', function(data){
    console.log(data);
  });
});
```

Location

```
var options = { enableHighAccuracy: true
  , maximumAge: 0
  , timeout: 5000
}

function success(pos) {
  var crd = pos.coords
  console.log( ' Latitude: ' + crd.latitude
    + ' Longitude: ' + crd.longitude
    + ' More or less ' + crd.accuracy + ' meters.' )
}

function error(err) { console.warn('ERROR(' + err.code + '): ' + err.message) }

navigator.geolocation.getCurrentPosition(success, error, options)
```

Orientation

```
screen.addEventListener("orientationchange", handleScreenOrientation)
```

```
screen.lockOrientation('landscape')
```

```
window.addEventListener("deviceorientation", handleOrientation, true)
```

```
window.addEventListener("devicemotion", handleMotion, true)
```

MIDI

```
function onMIDISuccess( midiAccess ) { window.midiAccess = midiAccess }
function onMIDIFailure(msg) { console.log( "Failed to get MIDI access - " + msg ) }
navigator.requestMIDIAccess().then( onMIDISuccess, onMIDIFailure )

function onMIDIMessage( event ) {
    var str = "MIDI message received at timestamp " + event.timestamp
        + "[ " + event.data.length + " bytes]: "
    for (var i=0; i < event.data.length; i++)
        str += "0x" + event.data[i].toString(16) + " "
    console.log( str )
}

function startLoggingMIDIInput( midiAccess, indexOfPort ) {
    midiAccess.inputs.entries[indexOfPort].onmidimessage = onMIDIMessage }
```

Gamepad

```
function runAnimation() {
  window.requestAnimationFrame(runAnimation)

  var gamepads = navigator.getGamepads()

  for (var i = 0; i < gamepads.length; ++i) {
    var pad = gamepads[i]
    console.log(pad.axes, pad.buttons)
  }
}

window.requestAnimationFrame(runAnimation);
```

Microphone

```
var constraints = { audio: true }
var context = new webkitAudioContext()

var gotMic = function(localMediaStream) {
  context.createMediaStreamSource(stream)
    .connect(context.destination)

}

var noMic = function(err) {
  console.log("Error: " + err) }

navigator.getMedia (constraints, gotMic, noMic)
```

Camera

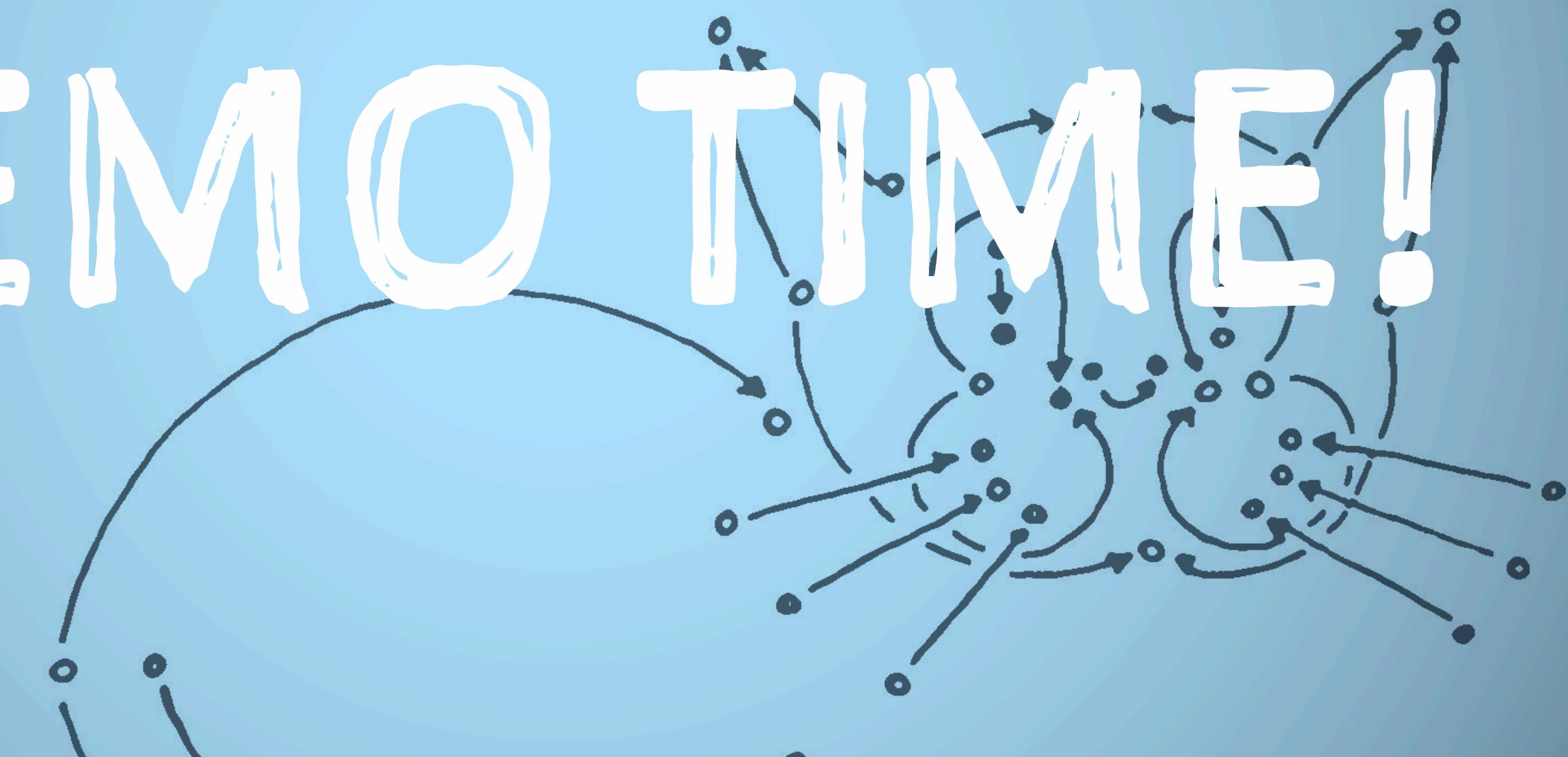
```
var constraints = { video: true, audio: true }

var gotStream = function(localMediaStream) {
  var video = document.querySelector('video')
  var canvas = document.getElementById('photo')
  video.src = window.URL.createObjectURL(localMediaStream)
  video.onloadedmetadata = function(e) {
    canvas.width = video.videoWidth
    canvas.height = video.videoHeight
  }
}

var noStream = function(err) { console.log("The following error occurred: " + err) }

navigator.getMedia (constraints, gotStream, noStream)
```

DEMO TIME!



@dann

dxnn @ github

dann @ bentobox . net

ack too far