

# Manipulación de Video de la Webcam

@davsket

[github.com/davsket/webcam-experiments](https://github.com/davsket/webcam-experiments)

David  
Avellaneda  
  
@davsket

Co-organizador de:  
BogotáJS  
JSConf.co  
Char.la  
ColombiaDev

Co-fundador de:  
Monoku





- Soy Tauro
- Me gusta correr en la playa
- Tengo un hijo (el de la foto)
- En serio me gusta mucho correr
- Me gusta compartir lo que se

# Agenda



# Agenda

- ¿Qué vamos a hacer?
- ¿Qué necesitamos?
- JUST DO IT!!! step++
- Momento philosoraptor



# ¿Qué vamos a hacer?

un video vale más que 1.000 imágenes

¿Qué necesitamos?

# ¿Qué necesitamos?

- getUserMedia (no mucha ciencia aquí)
- promesas (wtf?)
- video element (lo básico)
- canvas (lo básico + un poquito)
- raf (request animation frame)
- teoría de color (un poco)



getUserMedia

# getUserMedia

```
navigator.mediaDevices  
  .getUserMedia({video: bool, audio: bool})  
  .then(successCb, errorCb)
```

- Obtiene no solamente video sino también audio del micrófono
- Devuelve una promesa
- Al ser exitoso da un stream de video
- En el video se puede especificar dimensiones, frame rate, etc. [mozilla/getUserMedia](https://developer.mozilla.org/en-US/docs/Web/API/MediaDevices/getUserMedia)

# Promises

```
var p = new Promise((resolve, reject) => { resolve(5) })  
p.then(num => console.log(num)) // 5  
p.then(num => num * 5)  
  .then(num => console.log(num)) // 25
```

- Solo se puede resolver o rechazar una vez
- De ahí en adelante puedes preguntar cuantas veces sea el resultado
- Si una función retorna un valor, el siguiente then obtiene el valor de esta
- getUserMedia, fetch...

# Video (elemento)

- para recibir el stream
- para dibujarlo luego en un canvas

# Canvas

```
var ctx = canvas.getContext('2d')
```

```
ctx.drawImageData(0, 0, width, height, video) ✨
```

```
var frData = ctx.getImageData(0, 0, width, height) ✨ ✨
```

- para coger el frame del video y poder leerlo (luego lo manipulamos, no tiene nada que ver con canvas)
- luego para re-dibujar un nuevo frame modificado
- este va a ser el video resultado

# RAF

*(Request Animation Frame)*

```
var reqId = requestAnimationFrame(drawCb)  
var cancelAnimationFrame(reqId)
```

- se usa principalmente para animaciones
- optimiza la ejecución del callback para que coincida con el dibujado de un frame en la pantalla
- el navegador elige el frame más óptimo
- funciona como un setTimeout (no como un setInterval)

# Teoría Colores

rgb => hsl

hsl => rgb

para modificar correctamente los colores  
(luego lo veremos más claro)





# 1. Obter video

fase-1

# 2. Pintar el video

fase-2

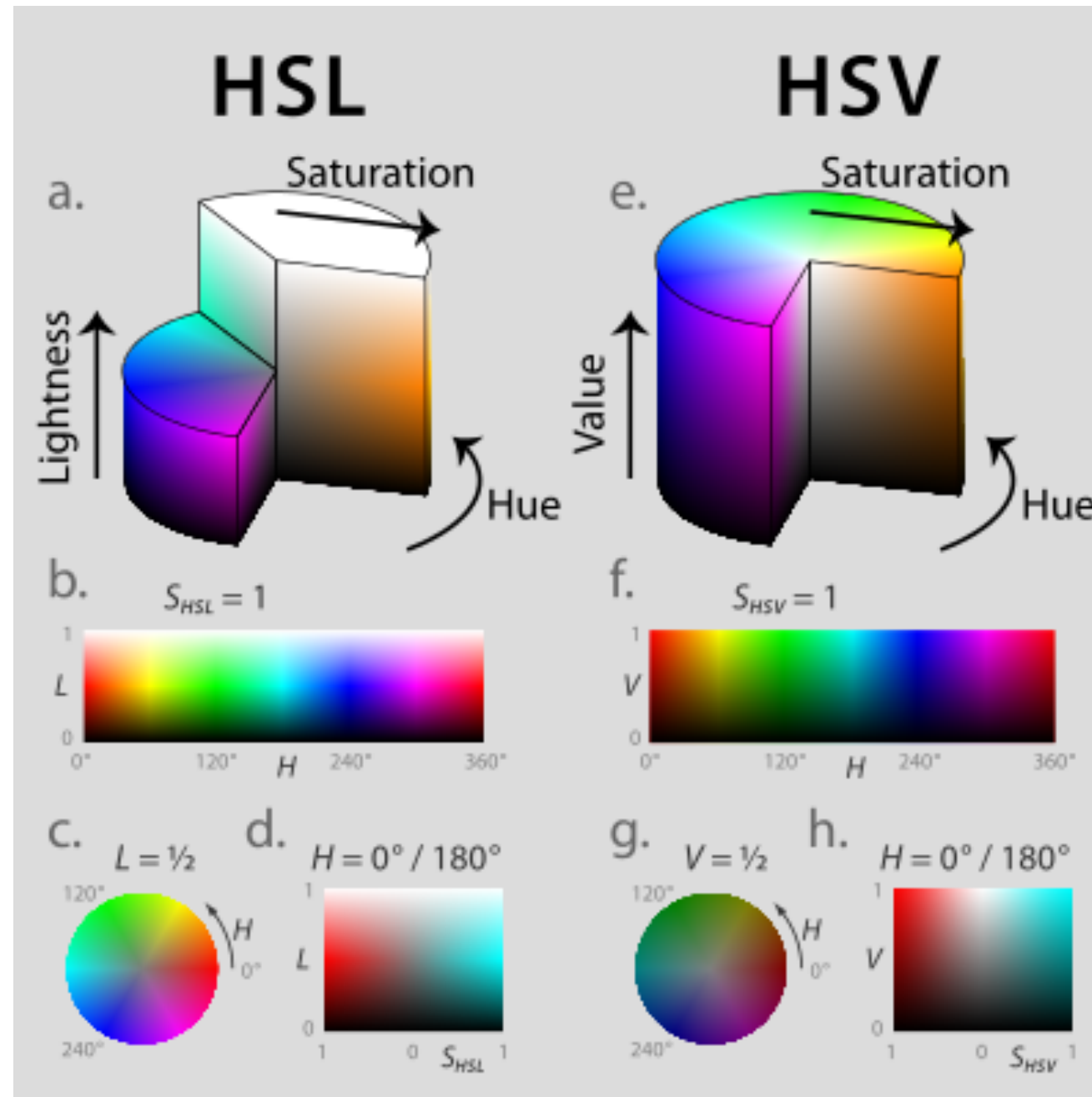
# 3. Dibujar en Canvas

fase-3

# 4. Manipular el Video

fase-4

# Teoría Colores



# HSL o HSV





# 5. Manipular ++

fase-5

Detectar fondos verdes  
y reemplazar por imágenes



Reconocer caras y montar  
máscaras encima como Snapchat

Preguntas?

Gracias!

@davsket

[github.com/davsket/webcam-  
experiments](https://github.com/davsket/webcam-experiments)