Measure! What?

Given task:
edit on you smartphone
a small part of a very large (xGB)
image, log file, or video.

How to measure and log the performance to find bottlenecks?

Search for APIs

The browser is your "operating system".

Specifications

This is a list of all the APIs that are available.

A
- Alarm API
- Ambient Light Events
- App Cache
- Application Compatibility Layer
- Apps
- Archive API
- Audio Channels API

B
- Background Tasks
- Battery API
- Beacon
- Bluetooth API
- Bluetooth API (Firefox OS)
- Broadcast Channel API
- Browser API

C
- CSS Counter Styles
- CSS Font Loading API
- CSSOM
- CSSOM View
- Camera API
- Canvas API
- Channel Messaging API
- Clipboard API
- Console API
- Contacts API
- Credential Management API

D
- DOM
- DOM (Non-standard)
- DOM Events
- Data Store API
- Device Orientation Events
- Device Storage API
- Directory Upload API
- Download API

E
- Encoding API
- Encrypted Media Extensions
- Engineering Mode API

F
- FMRadio API
- Fetch API
- File API
- File System API
- Firefox OS
- Frame Timing API
- Fullscreen API

G
- Gamepad API
- Geolocation API
- Geometry Interfaces

H
- HTML DOM
- HTML Drag and Drop API
- HTML Microdata API
- HTML Undo Manager API
- High Resolution Time

I
- Identity
- Idle API
- Image Capture API
- IndexedDB
- Input Port API
- Inter-App Connection API
- Intersection Observer API

K
- Kill Switch API

L
- L10N API
- Long Tasks

M
- MSISDN Verification API
- Media Capabilities API
- Media Capture and Streams
- Media Session API
- Media Source Extensions
- MediaStream Recording
- Mobile Connection API
- Mobile Messaging API
- Mozilla Payment API

N
- NFC API
- Navigation Timing
- Network Information API
- Network Stats API

P
- Page Visibility API
- Payment Request API
- Performance API
- Performance Timeline API
- Permissions API
- Permissions API (Firefox OS)
- Pointer Events
- Pointer Lock API
- Power Management API
- Presentation API
- Proximity Events
- Push API

R
- Request Sync API
- Resize Observer API
- Resource Statistics API
- Resource Timing API

S
- SVG
- Screen Capture API
- Screen Orientation API
- Selection API
- Server Sent Events
- Service Workers API
- Settings API
- Simple Push API
- Social API
- Speaker Manager API
- Storage
- Streams
- System Update API

T
- TCP Socket API
- TV API
- Time and Clock API
- Touch Events

U
- UDP Socket API
- URL API
- User Timing API

V
- Vibration API
- Voicemail API

W
- Wake Lock API
- Web Activities
- Web Animations
- Web Audio API
- Web Authentication API
- Web Components
- Web Crypto API
- Web MIDI API
- Web Manifest
- Web Notifications
- Web Speech API
- Web Storage API
- Web Telephony API
- Web Workers API
- WebGL
- WebRTC
- WebVR API
- WebVTT
- Websockets API
- WiFi Information API
- WiFi P2P API
- WiFi Tethering API

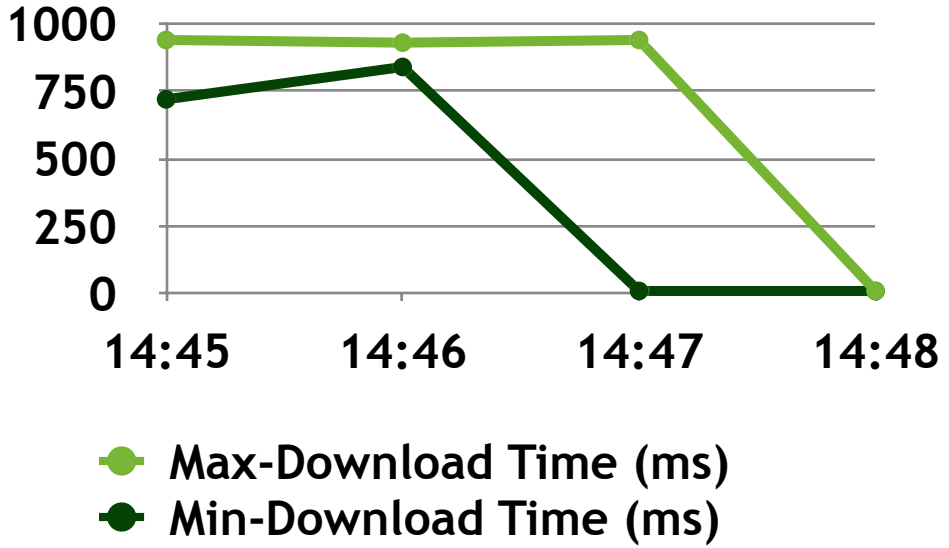
X
- XDomain
- XMLHttpRequest

https://developer.mozilla.org/en-US/docs/Web/API

Java Script Timing

```
loadEventEnd, navigationStart,
requestStart, responseEnd,
domComplete, ...
```

https://developer.mozilla.org/en-US/docs/Web/API/Navigation_timing_API

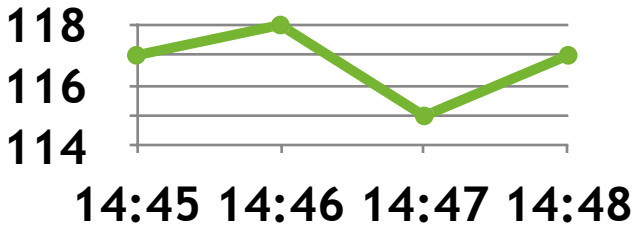


For example:

```
Performance.now()
```

https://developer.mozilla.org/en-US/docs/Web/API/Performance/now

Java Script Memory Tracking



We might use

`window.performance.memory`

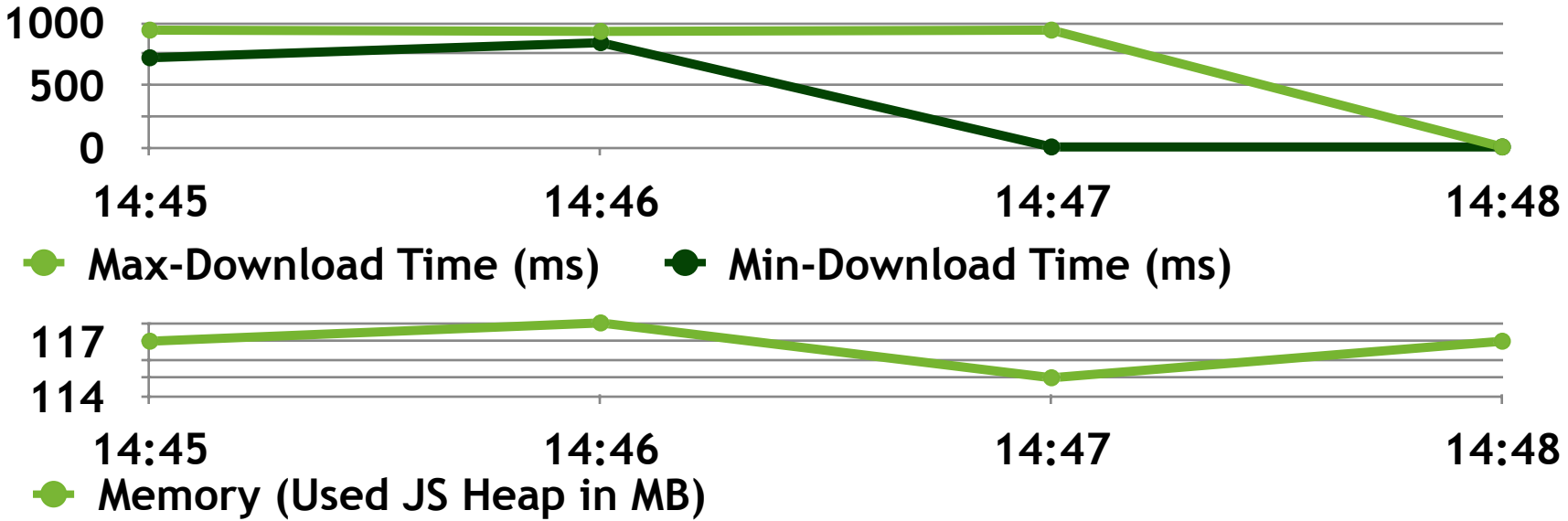
but

`window.performance.memory` is a propriety extension in Google Chrome

```
{
  totalJSHeapSize: 29400000,
  usedJSHeapSize: 15200000,
  jsHeapSizeLimit: 1530000000
}
```

https://trackjs.com/blog/monitoring-javascript-memory/

Design Log Information, View and Interpret



Design Your Logging Strategy and Design the Format of Your Log Messages

Logging — Why?

- For interpreting logged data (later)

Logging — How often?

- Allow configuration (e.g. log every 0.5 seconds)
- Enable/disable logging

Logging — Where?

- Display and store in file
- and/or send to server

Logging — Which information?

- Timestamp and current memory consumption
- Number of requests / requests per time (second)
- Max/min download time of requests
- Optional: network, latency, (overall) bytes received/sent, ...