Video (10)

Assets Optimization

Frontend performance optimization technique

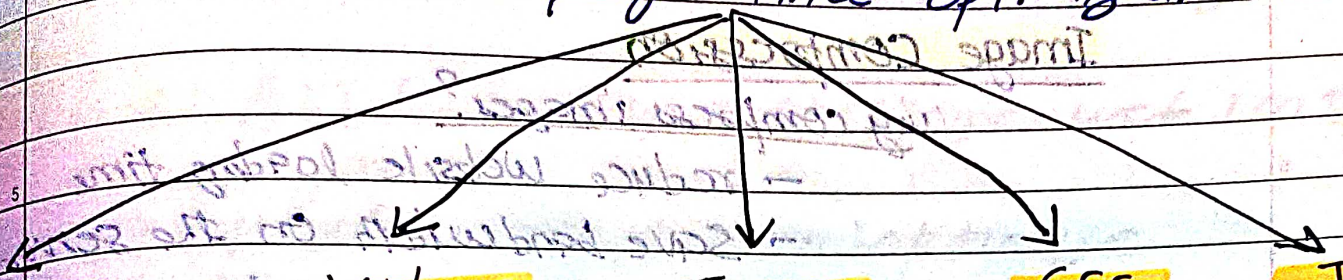


Image	Videos	Fonts	CSS	JS
mpression	- progressive enhancement	- font display	- lazy loading	- defer Vs async
zy loading	- Replace Gif's with video	- FLOUT with decorator	- critical css	- Web worker
ogressive nhancement	- Responsive poster Image	- Data URI class	scheduling	- lazy loading
ient HTTP hint	- Streaming	- preload		
sponsive mage	- Videos with no audio	- progressive enhancement		
daptive mage	- preload	- Async load		
lur effect		- font face		
alid primary olor		- Observer.		

Image Optimization

Image Compression

Why compress images?

- reduce website loading time
- save bandwidth on the server
- save resource and money
- improve user experience
- enhance SEO ranking

Compression tool example :-

- JPEG Optimizer
- Kraken
- Tiny PNG
- Optimizilla
- JPEG.io
- Image Recycle
- Compressor.io
- Ezgif

Image Progressive Enhancement

AVIF as a progressive Enhancement

```
<picture>                                   AVIF
    <source srcset="photo.avif" type="image
Webp → <source srcset="photo.webp" type="image/webp"
    <img src="photo.jpg" alt="Photo" width="300" height
</picture>
```

fallback JPEG

→ We can ask browser whether it support AVIf & Webp or "fallback" is JPG.

→ AVIf is most optimize web image format.

→ Webp is next most optimize

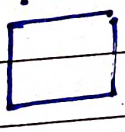
Client hint HTTP headers

Device Pixel Ratio (DPR)

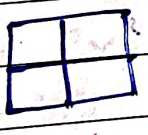
one CSS pixel

Many screen pixels

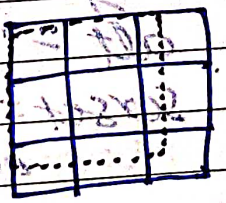
DPR:1



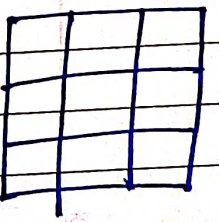
DPR:2



DPR:2.25



DPR:3



Every device has a different DPR

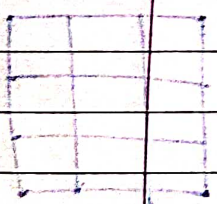
High resolution devices have a higher DPR

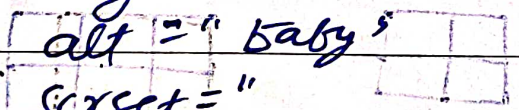
var scale = window.devicePixelRatio

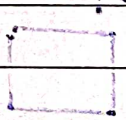
<img
alt = "Hey baby smile"
src = "baby"
srcset = "
baby-highres.jpg 2x,
baby-high-3.jpg 3x
"
>

on the basis of DPR of screen it will load one of the image.

we can also render on the basis of other scenario like we have size of 70vw minimum so on the basis of that the guillotine container have different width & we can load different image







<img
alt = "baby"
srcset = "
baby-s.jpg 300w,
baby-m.jpg 600w,
baby-l.jpg 1200w,
baby-xl.jpg 2000w
"
sizes = "70vmin"
>

different image with different width

container size

Next example

```
<img
srcset="
  baby-s.jpg 300w,
  baby-m.jpg 600w,
  baby-l.jpg 1200w,
  baby-xl.jpg 2000w"

size="
  (min-width: 2420em) 2000px,
  (min-width: 720em) 33.3vw,
  (max-width: 500px) calc(100vw - 20px),
  100vw"
src="baby.jpg"
alt="cute baby"
/>
```

more complex example :-

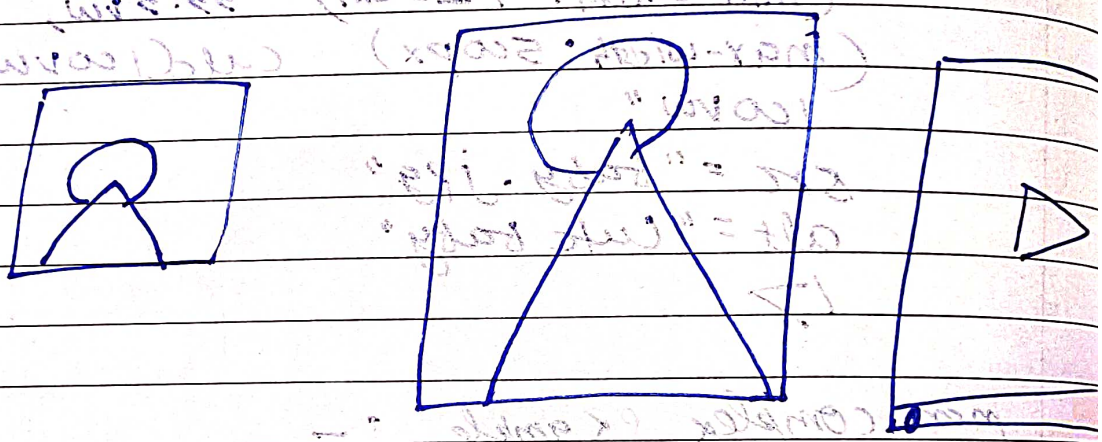
```
<picture>
  <source
  srcset="
      baby 2x,
      baby.jpg
  media="(min-width: 1000px)"
  />
  <source
  srcset="
      baby 2x
      baby.jpg"
  media="(min-width: 600px)"
  />
<img
  srcset="baby.jpg 2x"
  src="baby.jpg"
```

Adaptive Media loading

Home → products → product

Item component media

connection: Slow Median Fast



On the basis of network speed, device configuration & storage space we can take decision what to play

- navigator.connector.saveData → either save mode u
- " .deviceMemory → enough memo
- " .hardwareConcurrency → how many cor... based on that will do multi threading

```
if ('connection' in navigator) {
    if (navigator.connection.saveData == true)
    {
    // Implement data saving operations here.
    }
}

const memory = navigator.deviceMemory;
console.log ('This device has at least ${memory} GIB of
            RAM');


let workerList = [];
for (let i=0; i < window.navigator.hardwareConcurrency; i++)
    {
    let newWorker = {
        worker: new Worker ('cpuworker.js'),
        inUse: false
    };

workerList.push (newWorker);

}
```

→ Disable video autoplay on slow network

```
// only prefetch in-viewport links on 4G

if (!isOKConnection() || isDataSavingMode()) {
    // don't prefetch visible ones
    return;
}
```

→ Limit carousel image loads on Data Saver/3

Small image with Blur effect

If image is not loded : we can show this effect.

we required 1 image (very small image) 1px or 2px.

```
.background-image {
  background-image: url("https ....")
  filter : blur(5px);
  webkit-filter : blur(5px);
  background-size : cover;
}
```

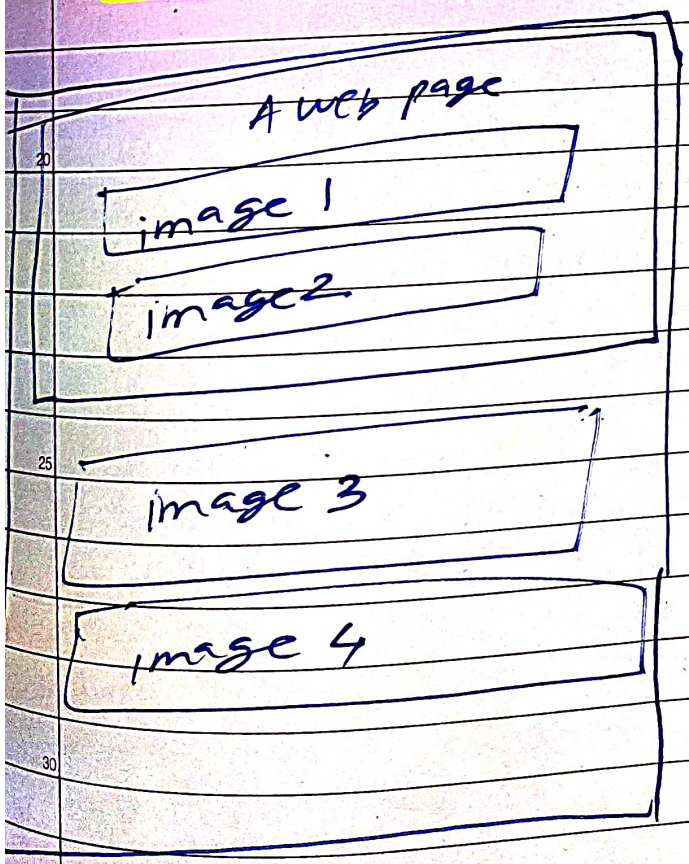
Solid Primary Color As Background

At the time of loading it shows primary color.

CSS Sprites

All the images are club together in a single image such that it form grid & we can use that grid for all the image.

Lazy loading




```
// lazy load
<image src = "turtle.jpg"
       alt = "Lazy turtle"
       loading = "lazy"
>

//Lazy load with eggs
<img src = "rabbit"
     alt = "fast rabbit"
     loading = "eager"
>

Vanilla-lazyload
<img data-src = "turtle.jpg"
     loading = "lazy">
```

Date :
Page No.

Video Optimization (Progressive Enhancement)

```
<video autoplay loop muted playsinline>
  <source src="Chakde-system-design.webm" type="video/webm" />
  <source src="Chakde-system-design.mp4" type="video/mp4" />
</video>
```

fall back

max optimized

Replace gif with HTML5 video

GIF < MP4 < WebM

less optimized.

max optimized

Date :
Page No.

Responsive Poster Image

If you want to show thumbnail in the video. then we can use this approach

```
<video controls class = "responsive-background"
poster = " url-something " >
    <source src = "movie.mp4" type = "video/mp4" >
    <source src = "movie.ogg" type = " video/ogg" >
</video>

<style type = "text/css">
    .responsive-background {
        background-image : url ('tiny-poster.jpeg');
        background-position : center;
        background-size : cover;
    }
    @media (min-width : 75rem) {
        .responsive-background {
            background-image : url ('large.jpeg');
        }
    }
</style>
```

Date :
Page No.

Streaming

(Netflix System Design) Checkout previous.

Remove audio from video

For video played in background

Platform based video dimensions

```
<video controls>
<source src="my-video.mp4" type="video/mp4"
        media="all & (max-width: 480px)">
<source src="my-video.mp4" type="video/webm"
        media="all & (max-width: 480px)">

<source src="my-video.mp4" type="video/mp4">
<source src="my-video.webm" type="video/webm">

</video>
```

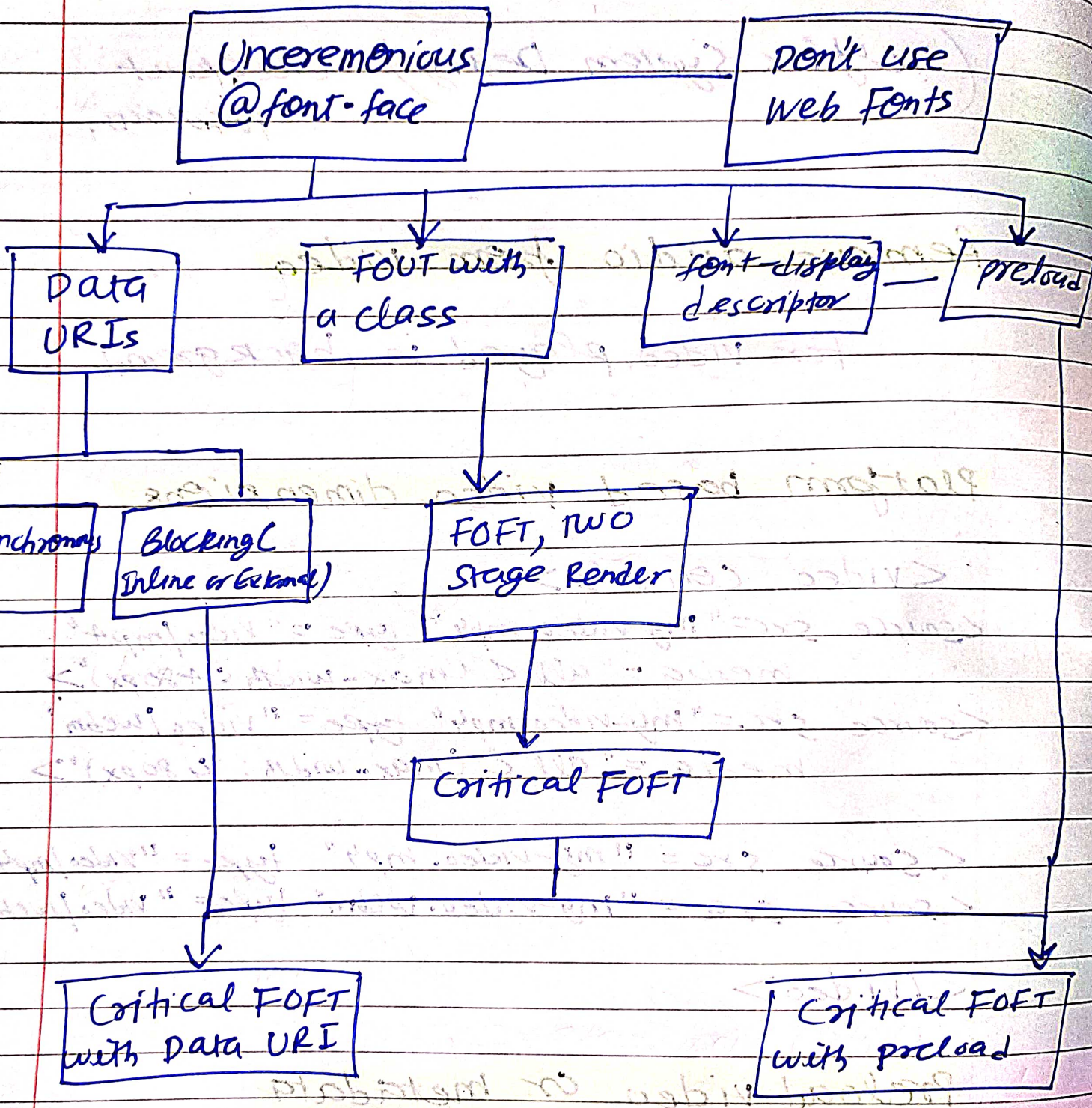
Preload video or metadata

```
<video controls preload="none/auto/metadata">
  <source src="video.mp4" type="video/mp4">
  <source src="video.webm" type="video/webm">

</video>
```

Font Optimization

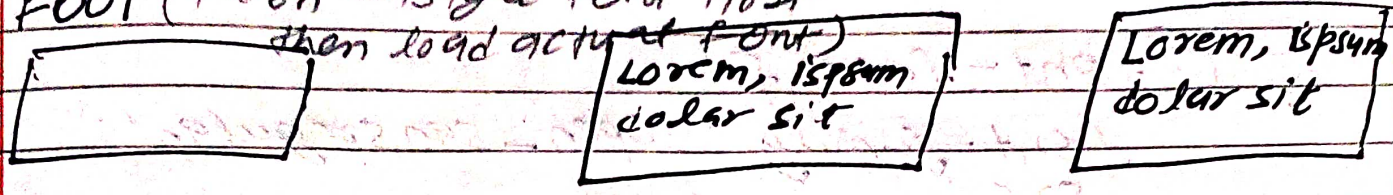
Date :
Page No.



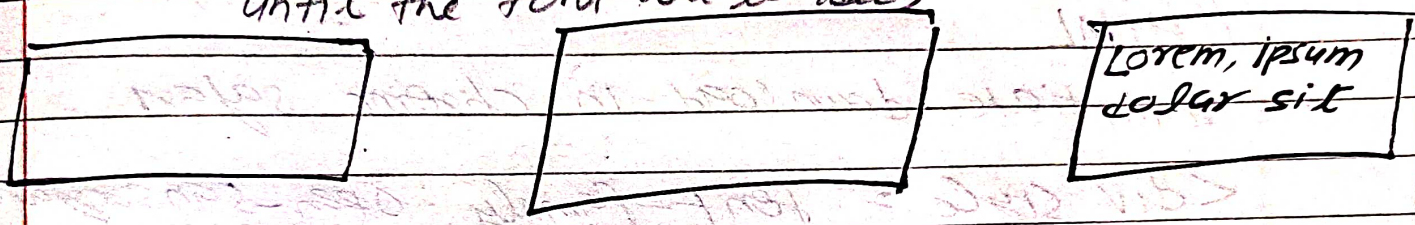
Font Display Descriptor

Date :
Page No.

FOUT (flash unstyle font first then load actual font)



FOIT (flash of invisible text first until the font are loaded)




```
@font-face {
    font-family: "My Webfont"; custom name
    src: url('myfont.woff2') format('woff2'),
         url('myfont.woff') format("woff");
    font-display: swap;
}
```

define how the browser behaves during download

Progressive Enhancement (WOFF 2, WOFF)

WOFF < WOFF2

WOFF: less optimize

WOFF2: max optimize

FLOUT with Class

Date :
Page No.

```
//CSS
@font-face {
    font-family: 'open_sansregular';
    src:
}

//HTML
// initiate download in chrome safari
<div style="font-family: open-sansregular">
    //CONTENT
</div>
```

After using this it will initiate the download.

```
//JS
// Does not download
var el = document.createElement('div');
el.style.fontFamily = 'open_sansregular';

// Initiates download in firefox, IE, qt
document.body.appendChild(el);

// Initiate download in webkit/Blink
el.innerHTML = 'content';
```

Date :
Page No.

Data URI

```
@font-face {
    font-family: Lato;
    src: url('data.....; base64, .....')
        format('woff');
}
```

Preload

```
<link rel='preload"
    href = "url"
    as "font'
    type = "font/woff2 crossorigin>
```

Subset Web fonts

- Subfont
- Data URI or self host

Asynchronous Stylesheet

```
<link rel = "stylesheet"
    href = "$CSS&display = swap"
    media = "print"
    onload = "this.media = 'all'"
/>
```

Font face Observer

Date:
Page No.

```
@font-face {
    font-family: 'lato";
    src: url(' '),
         url(' '),
         url(' ')
    font-weight: 300;
    font-style: normal;
}

body {
    font-family: sans-serif;
}

.font-loaded body {
    font-family: lato, sans-serif;
}
```

// Terminal

```
npm i fontfaceobserver --save
```

JS

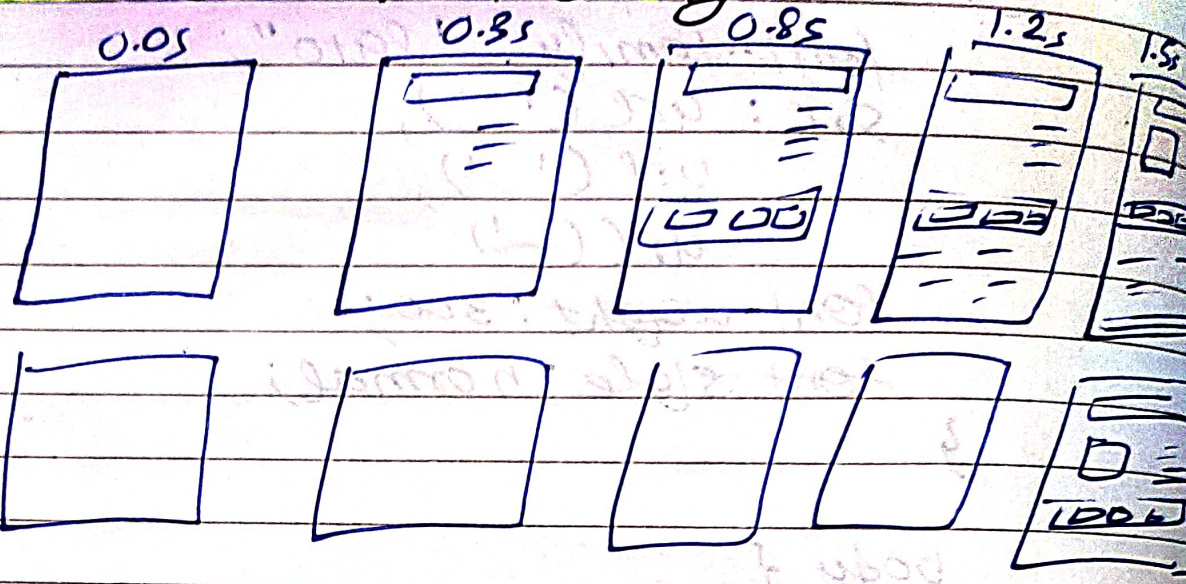
```
var fontfaceobserver = require('fontfaceobserver')

    new fontfaceobserver('lato')
        .check()
        .then(function(){
document.documentElement.className +=
                    " fonts-loaded";
        });
```

CSS Optimization

Date :
Page No.

Critical CSS Rendering



49:21

Date :
Page No.

Lazy loading

```
<link href="style.css" rel="stylesheet" media="all">
<link href="portrait.css" rel="stylesheet" media="(orientation: portrait)">
<link href="print.css" rel="stylesheet" media="print">
```

media is giving our of box optimization

```
// Add script at bottom of page
// If you want to load in the end
<script>
  (function() {
    var cssMain = document.createElement('link');
        cssMain.href = '/css/main.css';
        cssMain.rel = 'stylesheet';
        cssMain.type = 'text/css';
    document.getElementsByTagName('head')[0].appendChild(cssMain);

  })();
</script>
```

```
// Preload style asynchronously.

<link rel="preload" href="style.css" as="style"
      onload="this.onload=null;this.rel='stylesheet'">

<noscript>
  <link rel='stylesheet' href="style.css">
</noscript>
```

Date :
Page No.

JavaScript Optimization

Defer Vs Async

```
<script>
<script defer>
<script async>
<script type = "module">
<script type = "module" async>
```

Using Web Worker.