3 Keys 2 Performance

- Improve frontend
- Improve network latency
- Improve backend

Network Performance

→ Shrink the files

Minimize text and images!

↳ webpack solves it (pre-build & production)

→ Subtler...

↳ PNGs are usually smaller than JPGs

↳ SVG are versatile and generally compact

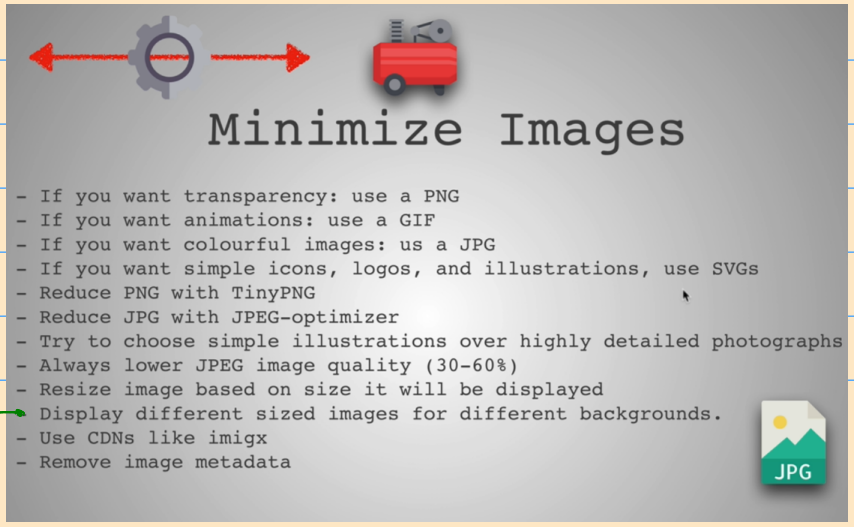
also, check those links:

https://99designs.com/blog/tips/image-file-types/

https://pageweight.imgix.com/

https://www.sitepoint.com/gif-png-jpg-which-one-to-use/

AND remember this image =>



use, e.g.: media queries!

→ Minimize delivered content!
- Limit the # of stuff you send at each request!

https://stackoverflow.com/questions/985431/max-parallel-http-connections-in-a-browser

Critical Render Path

DOM → CSSOM → Render Tree → Layout → Paint

1) JS scripts => loaded later!
Let CSS have higher priority.
general rule: { script tags at the bottom
{ style tags ASAP, i.e., at the top
2) Make CSS as light as you can
- Only load whatever is needed (maybe not using extend CSS...)
- Above the fold loading
- Use Media Attrs
- Use less specificity.
3) JS is the worst!

- Load Scripts Asynchronously
- Defer script loading
- Minimize DOM manipulation
- Avoid long running JS.

Read more about it

https://stackoverflow.com/questions/10808109/script-tag-async-defer

Also, check out "Prefetching"

https://css-tricks.com/prefetching-preloading-prebrowsing/

"Resource: Performance Tools" has some neat stuff, as well!

About HTTP/2...

https://developers.google.com/web/fundamentals/performance/http2/

... and HTTP/3

https://blog.cloudflare.com/http3-the-past-present-and-future/