1) A melhor forma seria utilizar algum task runner (Grunt ou Gulp) para junta-los em apenas um arquivo. Fazendo com que só necessite de uma chamada para que o css inteiro seja carregado.

2)

a - Utilizar task runner para melhoria, compressão e minificação dos arquivos html, js e css.

b - Otimizar as imagens conforme cada necessidade.

c - cache

3) jmeter, apache bench.

4) header, footer, section, aside, article, nav, div's.

5) No método get os parâmetros são passados no cabeçalho da requisição, que podem ser vistos na url. O método post envia os parâmetros no corpo da requisição HTTP. Escondendo eles da url.

6) flex, grid, inline, inline-block, block, table, table-cell, table-row, initial, inherit, none, inline-flex e inline-grid

7) O elemento inline não aceita padding nem magin. Já o elemento inline-block se comportará como um elemento inline, mas aceitando padding, margin, height e width.

8) O position relative em si não altera o posicionamento, ele necessitará de algum valor de posicionamento para tal (top, left, right, bottom). E também pode ser setado um z-index nele.

O position fixed como o próprio nome já diz vai deixar o elemento estatico na tela, no caso no viewport do usuário. Utilizado bastante para desenvolver menus que acompanham o scroll.

O position abolsute literalmente te deixa posicionar o objeto aonde voce quiser utilizando valores de posicionamento (top, left, right, bottom).

Static é o modo padrão setado para todos os elementos html.

9) o .call invoca uma função com um dado valor this e argumentos passados individualmente, enquanto o .apply os argumentos são providos com um array ou objetos parecidos com um array.

10) == faz comparação somente de valor, === faz comparação de valor e de tipo