

## 1 RTT render

Implemente a técnica de renderizar o topo do site em 1 RTT como vimos na aula.

1. Divida o CSS em duas partes: o inicial e o restante.

Os arquivos do inicial são: `reset.css`, `base.css`, `colors.css`, `font.css`, `block-header-busca.css`, `block-header.css`, `block-buttons.css`, `block-categoriaCard.css`, `block-titulo-destaque.css`, `home-fundo.css`, `home.css`, `home-diferenciais.css`

Os demais não são necessários pro topo.

1. Faça o carregamento **inline** dos arquivos iniciais. Com nosso plugin do gulp, isso significa colocar o atributo `inline` na tag link de *cada arquivo*. Ex:

```
<link inline rel="stylesheet" href="assets/css/reset.css">
```

1. O resto dos arquivos deve ser buildado pelo gulp num arquivo `async.css`. Faça o truque do duplo comentário como vimos na aula para evitar que a tag link gerada seja colocada no HTML.

```
<!--
<!-- build:css assets/css/async.css -->
  <link rel="stylesheet" href="assets/css/block-conteudo.css">
  <link rel="stylesheet" href="assets/css/block-cursoCard.css">

.... ETC DEMAIS ARQUIVOS AQUI...
<!-- endbuild -->
-->
```

1. Agora precisamos carregar esse `async.css` de forma assíncrona com o truque do `preload`. No head da página:

```
<link href="assets/css/async.css" rel="preload" as="style" onload="this.rel='stylesheet'">
<noscript><link href="assets/css/async.css" rel="stylesheet"></noscript>
```

1. Se testar no Chrome ou no Opera já deve funcionar. Mas vamos adicionar os polyfills para navegadores antigos.

```
<script async src="assets/js/loadcss.js"></script>
<script async src="assets/js/cssrelpreload.js"></script>
```

1. Ufa, agora faça o build da página rodando `gulp` no terminal e teste.

O que ganhamos?

