

BrowserSync e pasta servida

Para usarmos o BrowserSync precisamos instalá-lo como qualquer outro módulo do Node.js e importá-lo em nosso `gulpfile.js`. Quando formos configurar sua tarefa, precisamos passar um objeto como parâmetro para a função `browserSync.init()`. Qual das opções configura corretamente para servir todos os arquivos de `projeto/src`?

** Lembre-se que o seu `gulpfile.js` fica na pasta `projeto`.

A

```
var browserSync = require('browser-sync');

gulp.task('server', function() {
  browserSync.init({
    {
      baseDir: 'src'
    }
  });
});
```

B

```
var browserSync = require('browser-sync');

gulp.task('server', function() {
  browserSync.init({
    server: {
      baseDir: 'projeto/src'
    }
  });
});
```



```
var browserSync = require('browser-sync');
gulp.task('server', function() {
  browserSync.init({
    server: {
      baseDir: 'src'
    }
  });
});
```

D

```
var browserSync = require('browser-sync');

gulp.task('server', function() {
  browserSync.init({
    server: 'src'
  });
});
```

A resposta correta é:

```
var browserSync = require('browser-sync');

gulp.task('server', function() {
  browserSync.init({
    server: {
      baseDir: 'src'
    }
  });
});
```

A função `init` do BrowserSync recebe um objeto com a chave `server`. É no atributo `baseDir` deste objeto que indicamos a pasta que o BrowserSync considerará como raiz quando acessarmos o endereço padrão `localhost:3000`. Isso significa que se houver o arquivo `projeto/src/index.html` podemos acessá-lo como `localhost:3000/index.html`, porque a pasta raiz é `src` da pasta `projetos`. É importante lembrar que essa configuração ainda não é suficiente para o `livereloading` funcionar.

PRÓXIMA ATIVIDADE