

Entendendo posicionamento com CSS de uma vez por todas

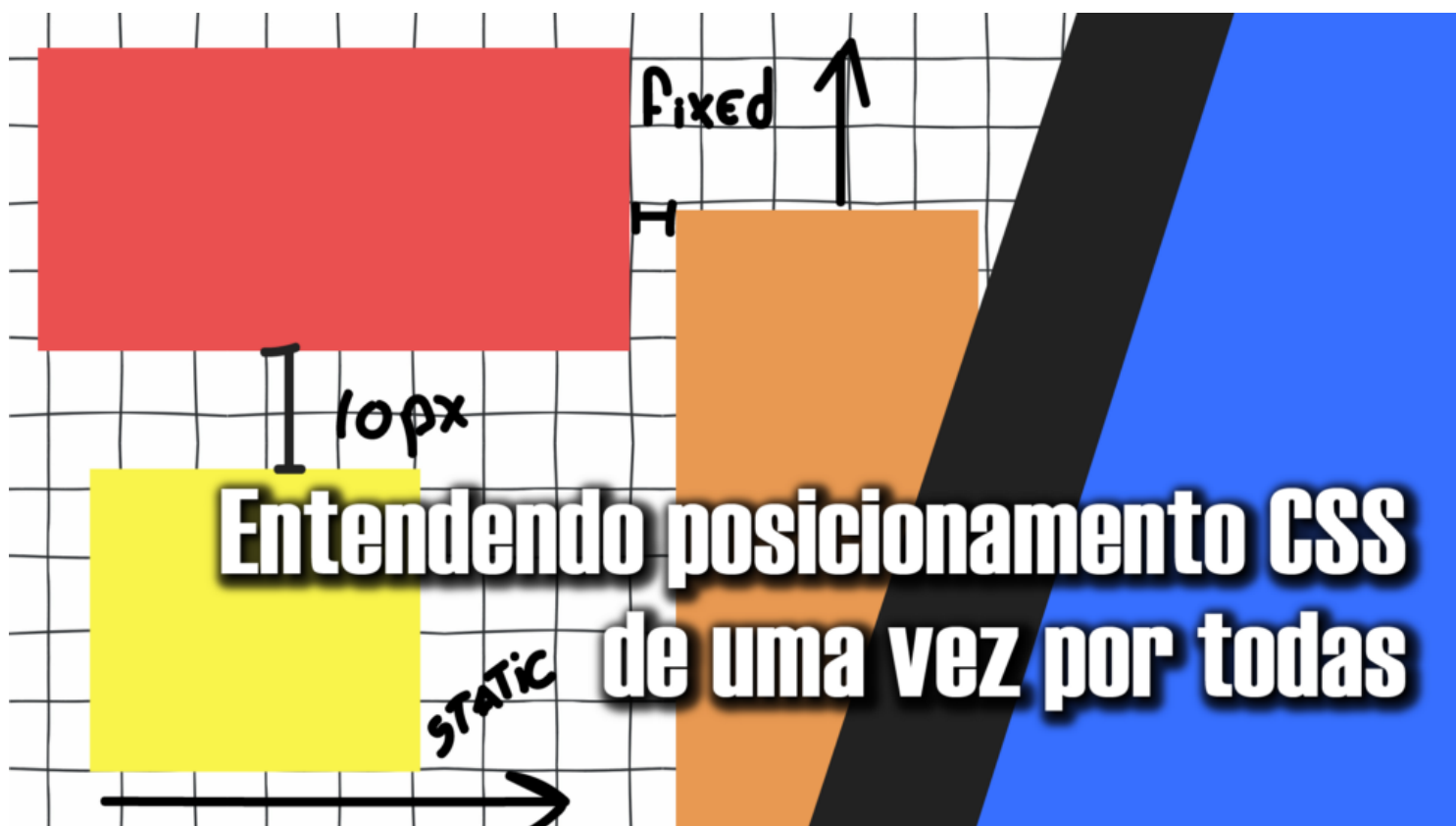
Static, absolute, relative, fixed e sticky são mais fáceis de usar do que parece.



Diego Martins de Pinho

Follow

Mar 11 · 5 min read



Antes de começarmos, tenho uma confissão a fazer: não sou o maior fã do mundo de CSS. Na verdade, sempre tive muito mais interesse no JavaScript (apesar de enxergá-lo com olho torto por muito tempo também). No entanto, essa concepção começou a mudar quando percebi que na verdade tudo não passava de uma simples ignorância minha: eu não tinha como gostar daquilo que eu não entendia direito.

Para tentar fazer com que você também perca esse possível desgosto por CSS, neste artigo vamos tratar de algo fundamental nesta tecnologia e que sempre causa bastante confusão: **como trabalhar com o posicionamento dos elementos.**

Se você nunca saiu chutando os posicionamentos dos elementos na esperança de que a página magicamente se consertasse, você é uma pessoa fora da curva e provavelmente este artigo não é pra você, rs.

Com esperança, ao final deste artigo você nunca mais chutará o valor do posicionamento dos elementos no seu CSS (ou ao menos fará chutes mais precisos).

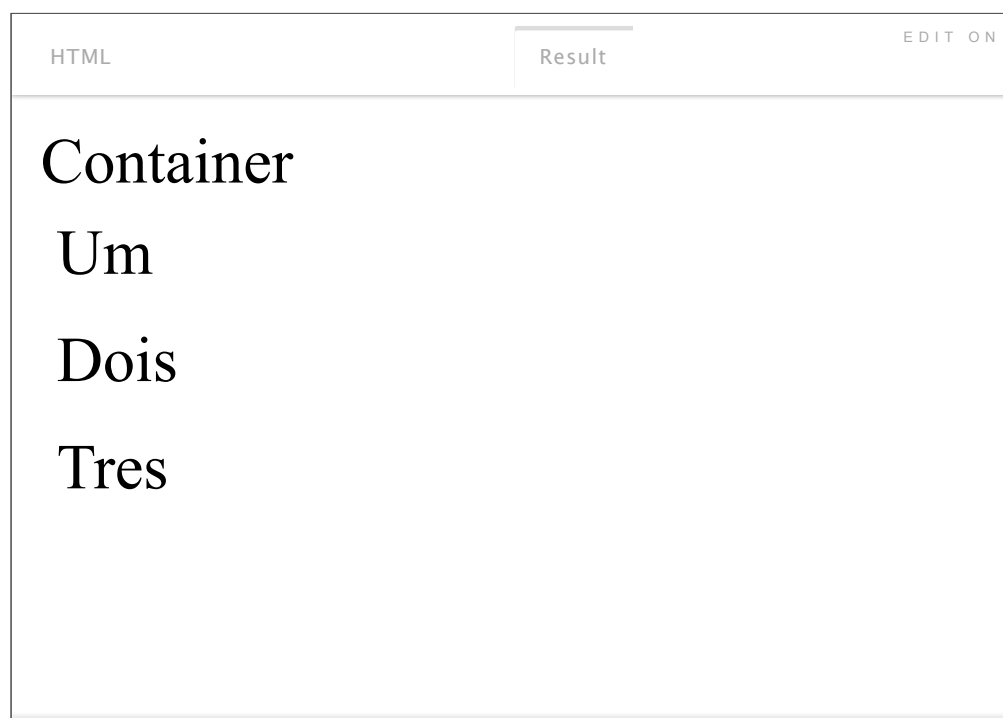
Agora sem mais delongas, vamos tratar individualmente cada um dos valores possíveis para o posicionamento de elementos no CSS. Recomendo que você os leia na ordem disponibilizada a seguir, pois as regras se complementam e entendimento pode ficar comprometido. Combinado? Então vamos lá.

Static (estático)

Este é o **comportamento de posicionamento padrão** que os seus elementos vão aderir ao serem adicionados na página. Em termos práticos isso significa que se nenhuma regra é associada de forma explícita, este valor é assumido por definição.

O `static` simplesmente indica aos elementos que eles devem seguir o fluxo definido declarado no documento (HTML). Ou seja, se tivermos três divs declaradas uma após a outra e elas tiverem o `static` como valor para posicionamento, elas simplesmente serão colocadas uma embaixo da outra na renderização.

Como mostra o exemplo abaixo:



Resources

1×

0.5×

0.25×

Rerun

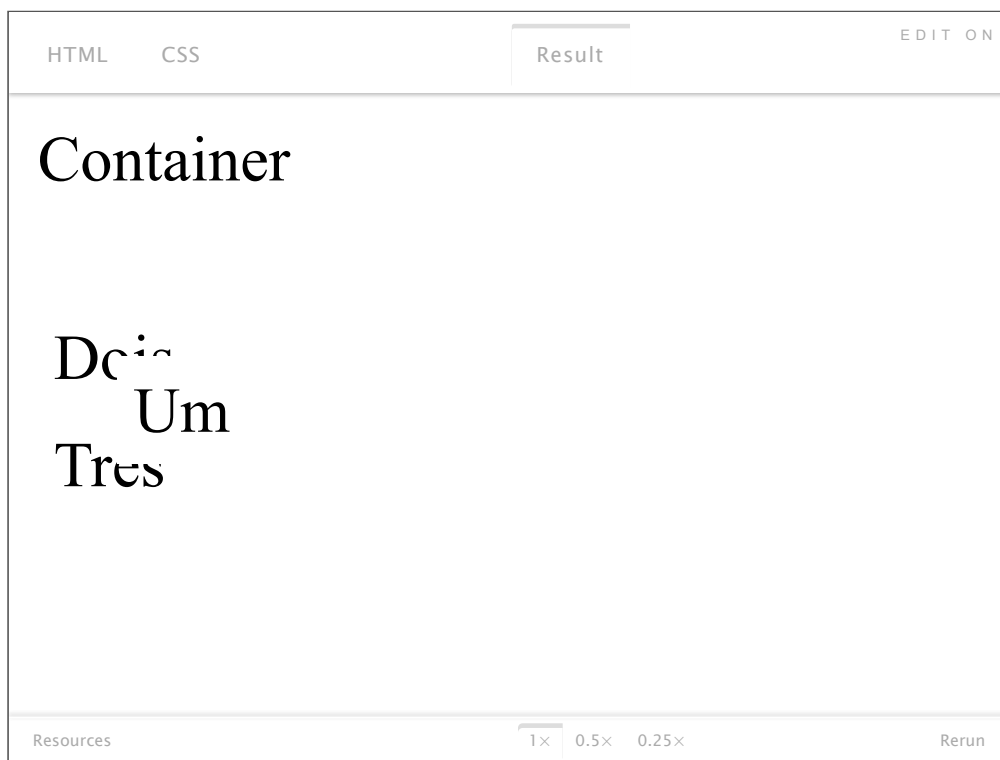
Relative (relativo)

Este tipo de posicionamento funciona de forma muito semelhante ao `static`, a única real diferença é que é possível configurar quatro propriedades do elemento ao qual o posicionamento está vinculado:

1. **top** (*cima*)
2. **bottom** (*baixo*)
3. **left** (*direita*)
4. **right** (*esquerda*)

Este tipo de posicionamento foi batizado como `relative` pois os valores inseridos em qualquer uma destas quatro propriedades fará com que o **elemento se desloque em relação a qual seria o seu posicionamento no documento** com a propriedade `static`. O elemento é retirado do fluxo do documento enquanto os outros elementos se comportam como se ele ainda estivesse lá.

É por isso que conseguimos sobrepor elementos usando essa propriedade, como mostra o exemplo a seguir:



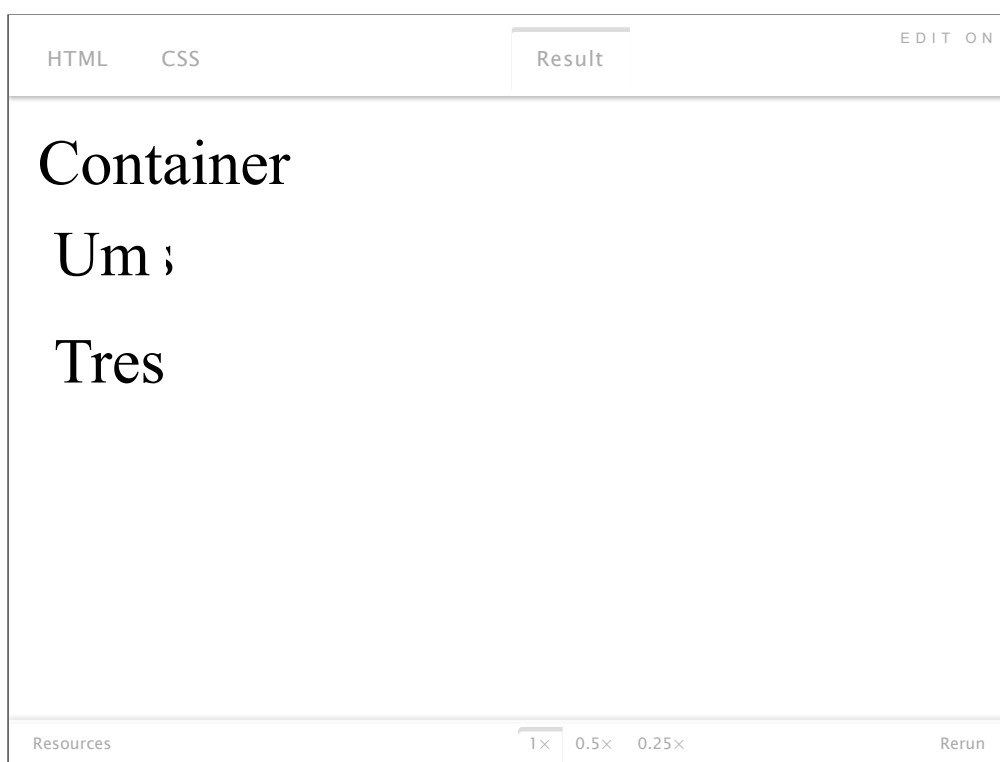
Difícilmente usaremos estas propriedades em um elemento de forma isolada, afinal, como o elemento sai do fluxo do documento, se torna muito difícil estilizar os elementos ao redor. No exemplo citado acima, teríamos que mudar as regras das outras divs também para que tudo ficasse no lugar.

Isso daria um trabalho absurdo dependendo da página, por isso temos o posicionamento absoluto.

Absolute (absoluto)

Este tipo faz que o elemento designado seja totalmente retirado do fluxo do documento, ou seja, os elementos ao redor reagem como se este elemento realmente não existisse.

Para entender o que isso significa, observe atentamente o exemplo abaixo onde atribuímos `position: absolute` a `div-um`.



Reparou como as outras divs se moveram para onde a `div-um` está?

Como o próprio nome sugere, a ideia aqui é que o elemento seja posicionado de forma absoluta em relação aos seus elementos pais (ou em outras palavras, seu *container*). No entanto, **aqui vai a grande sacada**: o elemento definido com posição absoluta só vai ser posicionado em relação a um elemento que não tenha uma posição estática. Caso não o encontre, o elemento sobe a sua árvore até encontrar um elemento que satisfaça essa condição.

Conseguimos ver isso bem no exemplo abaixo em que o `container` possui posicionamento estático e o posicionamento da `div-um` fica absoluta em relação ao início do documento e não do `container` em si.



É neste tipo de caso em que o posicionamento absoluto e o relativo conversam muito bem. Note o que acontece quando colocamos `position: relative` no `container`.



Fixed (fixo)

O posicionamento fixo está associado ao documento como um todo, ao contrário do posicionamento absoluto que leva em consideração um outro elemento. Ao usar o `fixed` em um elemento, ele terá sempre o mesmo posicionamento, mesmo levando em conta o *scroll* da página.

Em outras palavras, um elemento com posicionamento fixo tem um comportamento parecido com o que vimos no exemplo do posicionamento absoluto quando não há nenhum elemento de referência. Ele sempre leva em consideração o HTML como um todo.

Vamos ver como isso funciona na prática com o exemplo abaixo.



Sticky (grudento)

Este tipo é uma mistura interessante dos comportamentos obtidos com os posicionamentos `fixed` e `relative`. Enquanto o elemento não ultrapassa um limite especificado (ex: `top: 10`), ele possui o comportamento de elemento com posicionamento relativo; ao ultrapassar este valor limite, ele troca de comportamento e torna-se fixo até voltar ao limite especificado anteriormente.

Isso fica mais claro com um exemplo prático. Repare na página abaixo como o a `div-um` se comporta. Antes de “descermos” a página, o elemento tem o comportamento de