Event Listener

Conhecendo a ferramenta e aplicações.

Contextualizando....

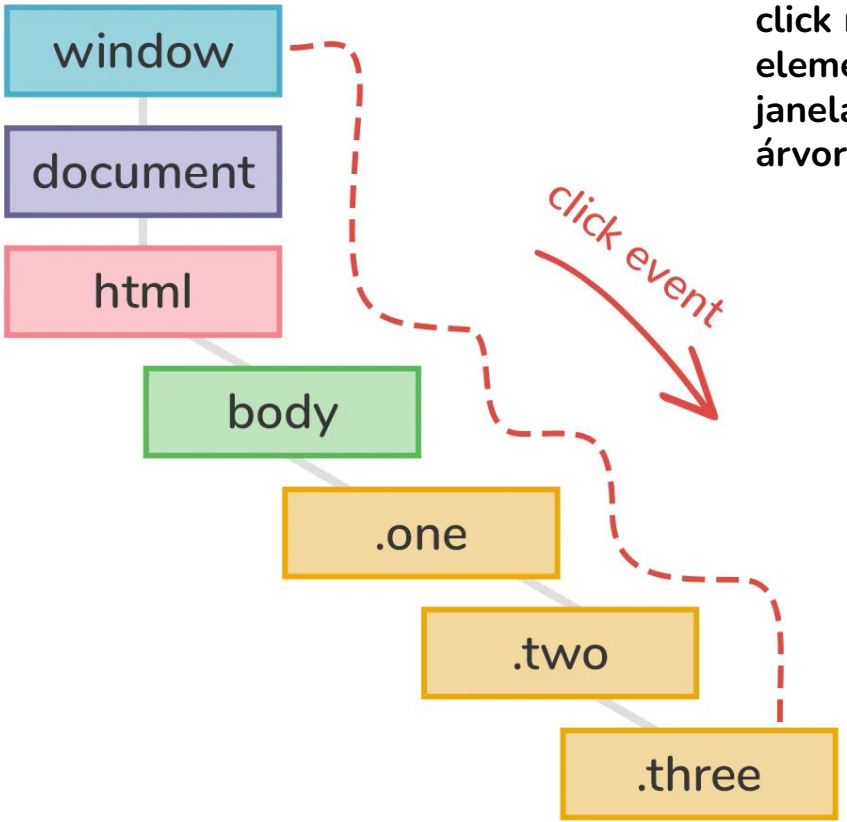
Como o próprio nome sugere, um Event Listener funciona como uma espécie de “escutador de eventos” esses eventos são capturados pelo Listener através de diversas entradas, como por exemplo, um mouse click, um hover, um keypress, dentre vários outros...

certo, mas o que é um evento ?

Segundo o MDN WEB, Eventos são ações ou ocorrências que acontecem no sistema que estamos desenvolvendo, no qual este te alerta sobre essas ações para que você possa responder de alguma forma, se desejado. Por exemplo, se o usuário clica em um botão numa página web, você pode querer responder a esta ação mostrando na tela uma caixa de informações.

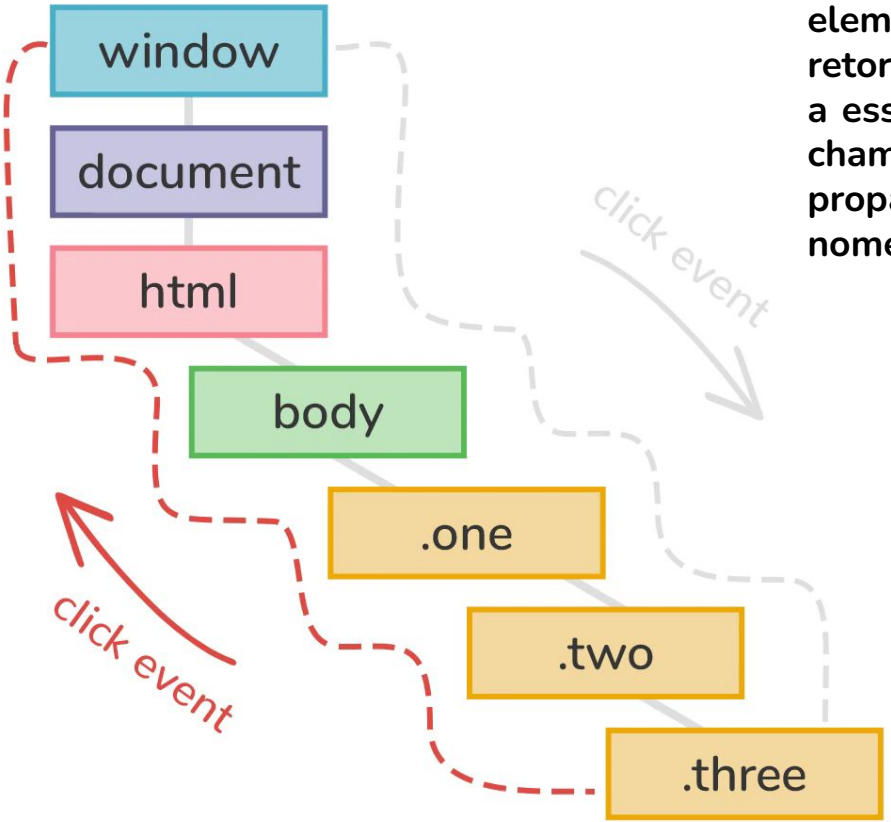
Cada evento que ocorre, possui um manipulador de eventos, um **event handler** (geralmente uma função em javascript criada pelo programador), que será executado quando o evento for disparado.

- A seguir veremos como funcionam as duas fases de um evento no DOM.



#1- Por padrão, o evento click não acontece a partir do elemento clicado, mas sim da janela pra baixo, como numa árvore.

#2- Essa é a primeira fase para encontrar um elemento, é chamada de fase de captura, ou *capturing*.



#3- Após o event encontrar o elemento procurado ele retorna ao topo da árvore, e a esse processo de retorno é chamada de fase de propagação (guarde esse nome) ou *bubbling*.

Bubbling e Capturing

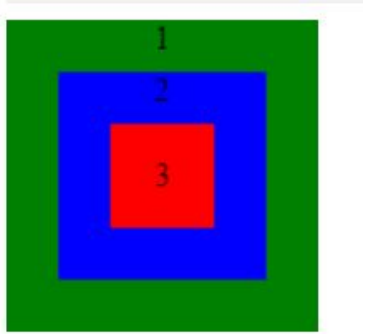
Bubbling.

Segundo o Imasters, O princípio fundamental do efeito bubbling diz o seguinte: depois que um evento é disparado no elemento mais distante de uma cadeia aninhada do DOM ele é disparado em seus elementos ancestrais na ordem crescente de aninhamento.

O efeito bubbling do JavaScript faz com que um clique na div 3 dispare o evento ligado a ele primeiro na div 3(também chamado de target) de dentro para fora , depois na div 2 e finalmente na div 1.

– clique na div 2 dispara o evento ligado a ele primeiro na div 2 (também chamado de target), depois na div 1.

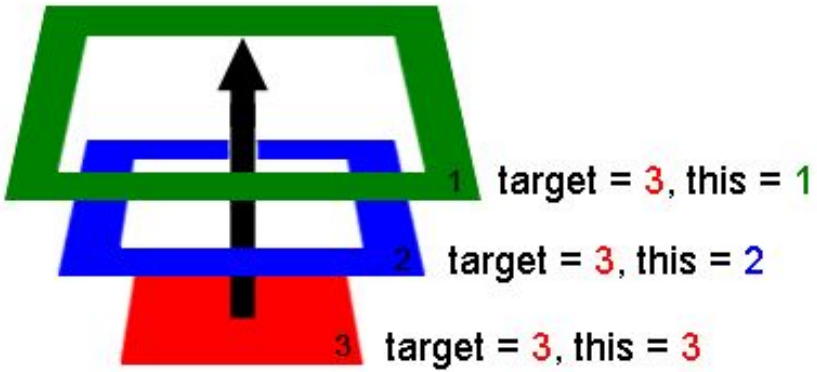
– um clique na div 1 dispara o evento a ele atrelado (também chamado de target) e nada mais.



***A ordem de disparo é chamada de bubbling order, pois o evento “borbulha” do elemento descendente mais baixo para seus ancestrais, tal como ocorre com uma bolha de ar na água.**

```
function logClass(){
    this.style.background="#123456";
    console.log(this.classList.value);
}
```

- this – refere-se ao elemento corrente, aquele para o qual o evento “borbulhou” ou ainda, aquele que dispara o handler.



Existe uma forma de inverter essa captura decrescente para uma captura crescente ?

- Sim, existe uma forma capaz de alterar o funcionamento do event, ao invés de funcionar como bubbling, ele funciona já a partir do capturing, dessa forma:

```
$divs.forEach(
    $div => $div.addEventListener('click', logClass,{
        capture:true
    })
);
```

dentro dos parâmetros do addEventListener, recentemente, foi adicionada a funcionalidade dele receber também objetos no parâmetro, sendo assim, nós setamos um objeto ao lado da função logClass , para permitir que o evento seja ativado desde a fase de captura (capture: true)

Ok, mas se quisermos cancelar o bubbling, para pegarmos apenas o elemento clicado ?

É possível interromper o efeito bubbling antes que ele percorra todos os elementos aninhados. Para tanto, utilizamos uma funcionalidade chamada stopPropagation();

- A seguir veremos como ele poderia ser implementado em nosso código.

```
$divs.forEach(
    $div => $div.addEventListener('click', logClass,{
        capture:false
    })
);

function logClass(e){
    e.stopPropagation();
    this.style.background="#123456";
    console.log(this.classList.value);
}
```

Após setar o capture como false, adicionamos ao e (event) à chamada de função **stopPropagation().**

- Essa função faz o javascript não executar o bubbling na fase 2, com isso ele retorna somente o **target** selecionado.

Dúvidas ?

Muito obrigado :)

Fontes:

https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/Building_blocks/Events

https://developer.mozilla.org/pt-BR/docs/Web/API/EventTarget/addEventListener

https://developer.mozilla.org/pt-BR/docs/Web/API/KeyboardEvent

https://www.freecodecamp.org/portuguese/news/funcoes-de-callback-em-javascript-o-que-sao-e-como-usa-las/

bubbling : https://www.youtube.com/watch?v=esXH65f7_n8&t=12s

Bubling & Capturing: https://imasters.com.br/front-end/javascript-bubbling-e-capturing