

FULL STACK WEB DEVELOPER

FERNANDO LIRA

FLAG

JAVASCRIPT

APRESENTAÇÃO – FERNANDO LIRA

3



it.fernandolira@gmail.com



<https://www.linkedin.com/in/fernandolira74/>



+351 93 317 99 21



@fernandolira74



DRUM KIT

Event Listener

```
document.querySelector('button').addEventListener("click",()  
=> alert("Clicou no primeiro botão!"));
```

... aplicar a todos

<https://developer.mozilla.org/pt-BR/docs/Web/Events>

Event Listener

```
const buttons = document.querySelectorAll('.drum');  
buttons.forEach((button, indice) => {  
    button.addEventListener("click", () => alert("Clicou no botão "));  
});
```

...OU

```
const numeroBotoes = document.querySelectorAll('.drum').length;  
for (let i = 0; i < numeroBotoes; i++)  
    document.querySelectorAll('.drum')[i].addEventListener("click", () =>  
        alert("Clicou no primeiro botão!"));
```

Play Sound

```
const numeroBotoes = document.querySelectorAll('.drum').length;
for (let i = 0; i < numeroBotoes; i++)
  document.querySelectorAll('.drum')[i].addEventListener("click",
function() {

  var buttonInnerHTML = this.innerHTML;

  switch (buttonInnerHTML) {
    case "w":
      var audio = new Audio("sounds/tom-1.mp3");
      break;
    case "a":
      var audio = new Audio("sounds/tom-2.mp3");
      break;
    case "s":
      var audio = new Audio("sounds/tom-3.mp3");
      break;
```

Play Sound

```
        case "d":
            var audio = new Audio("sounds/tom-4.mp3");
            break;
        case "j":
            var audio = new Audio("sounds/snare.mp3");
            break;
        case "k":
            var audio = new Audio("sounds/crash.mp3");
            break;
        case "l":
            var audio = new Audio("sounds/kick-bass.mp3");
            break;
    }
    audio.play();

});
```


Event Listener - Keydown

```
document.addEventListener("keydown",function(e) {  
    playSound(e.key);  
})
```

Animações

```
function buttonAnimation(currentKey) {  
    var activeButton = document.querySelector("." + currentKey);  
    activeButton.classList.add("pressed");  
    setTimeout(() => {  
        activeButton.classList.remove("pressed");  
    }, 100);  
}
```

EXERCÍCIOS

Desafios

SIMON GAME

<https://londonappbrewery.github.io/Simon-Game/>

Event Object

```
<h1>Event object</h1>  
<button>Click me</button>  
<p><a href="http://www.google.pt">GOOGLE</a></p>
```

```
<script>  
  const h1 = document.querySelector('h1');  
  const btn = document.querySelector('button');  
  const link = document.querySelector('a');  
  
  console.log('h1 ',h1);  
  console.log('btn ',btn);  
  console.log('link ',link);  
</script>
```

Event Object

```
//Passar o Event Object na callback como e (ou event ou qualquer
outra coisa)
//Verificar o Current Target e o This (sem arrow)
h1.addEventListener("click",function(e) {
    h1.style.color='red';
    console.log('Current target: ',e.currentTarget);
    console.log('This: ',this);
    //Equivalete a h1.style.color
    //e.currentTarget.style.color='green';
});

//O mesmo que o Evento anterior mas com função Arrow
//Verificar diferença no this!
/*
    h1.addEventListener("click",(e) => {
        h1.style.color='red';
        console.log('Current target: ',e.currentTarget);
        console.log('This: ',this);
    }); */
```

Event Object

```
//Event Type
btn.addEventListener('click', (e) => {
    e.currentTarget.style.color = "blue";
    console.log('Tipo de evento: ', e.type);
});

//Prevent default
link.addEventListener('click', (e) => {
    e.preventDefault();
});
```

Event Object – currentTarget vs. Target

```
<body>
  <button>Primeiro botão</button><br><br>
  <button>
    Segundo <strong>botão <strong>aninhado</strong> </strong>
  </button><br><br>
  <button>Terceiro Botão</button>
  <script>
    const buttons = document.querySelectorAll('button');

    buttons.forEach( button => {
      button.addEventListener('click', (e) => {
        console.log('Current target: ',e.currentTarget);
        //O target apanha o elemento efetivamente clicado!
        console.log('Target: ',e.target);
        e.currentTarget.style.color = 'red';
        //Verificar com o target
        //e.target.style.color = 'green';
      })
    })
  </script>
</body>
```


Event Propagation – Capture e Bubbling (default)

```
<div>  
  <ul>  
    <li>Ponto 1</li>  
    <li>Ponto 2</li>  
    <li>Ponto 3</li>  
  </ul>  
</div>
```

Event Propagation – Capture e Bubbling (default)

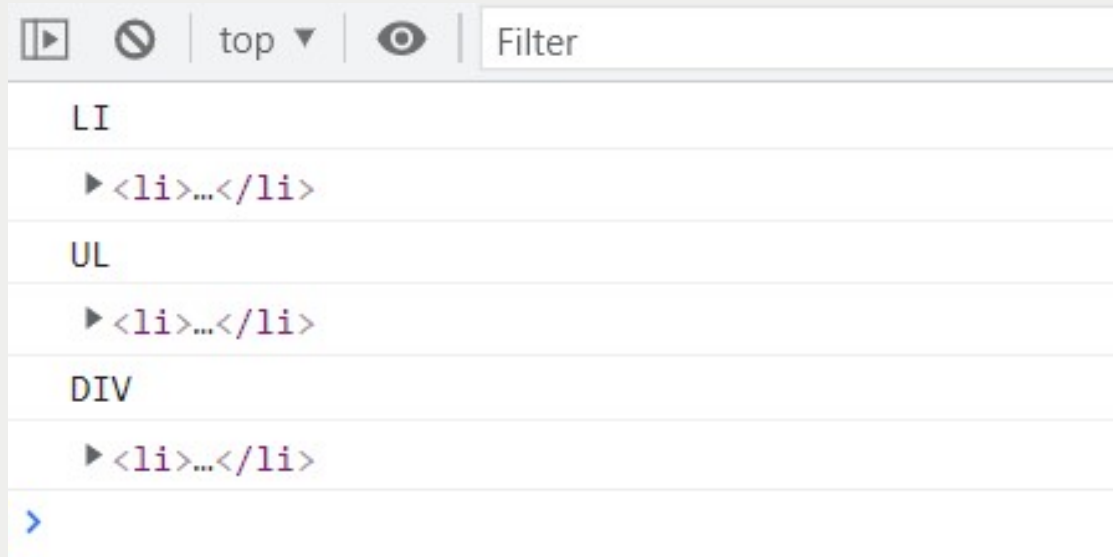
```
const div = document.querySelector('div');
const ul = document.querySelector('ul');
const li = document.querySelectorAll('li');

div.addEventListener('click', (e) => {
  console.log("DIV");
  console.log(e.target);
});

ul.addEventListener('click', (e) => {
  console.log("UL");
  console.log(e.target);
});

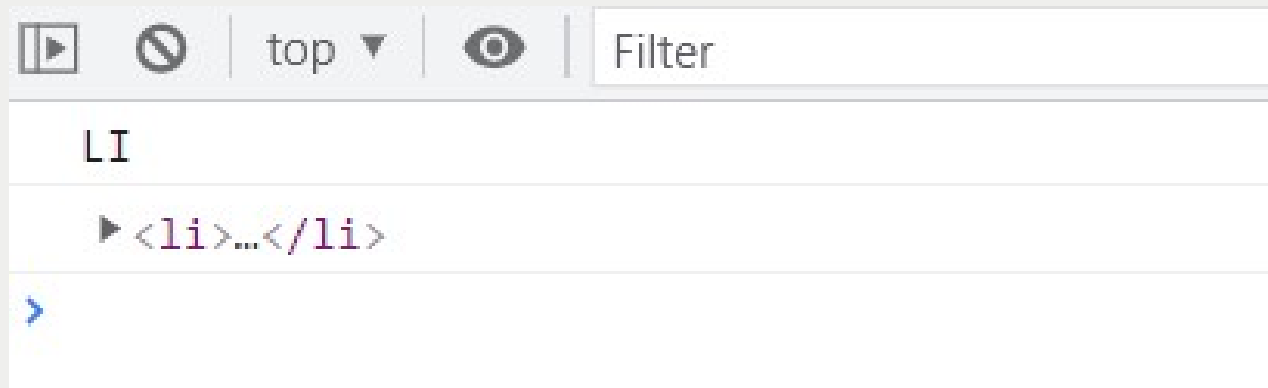
li.forEach( element => {
  element.addEventListener('click', (e) => {
    console.log("LI");
    console.log(e.target);
  });
});
```

Event Propagation – Capture e Bubbling (default)



Event Propagation – Capture e Bubbling (default)

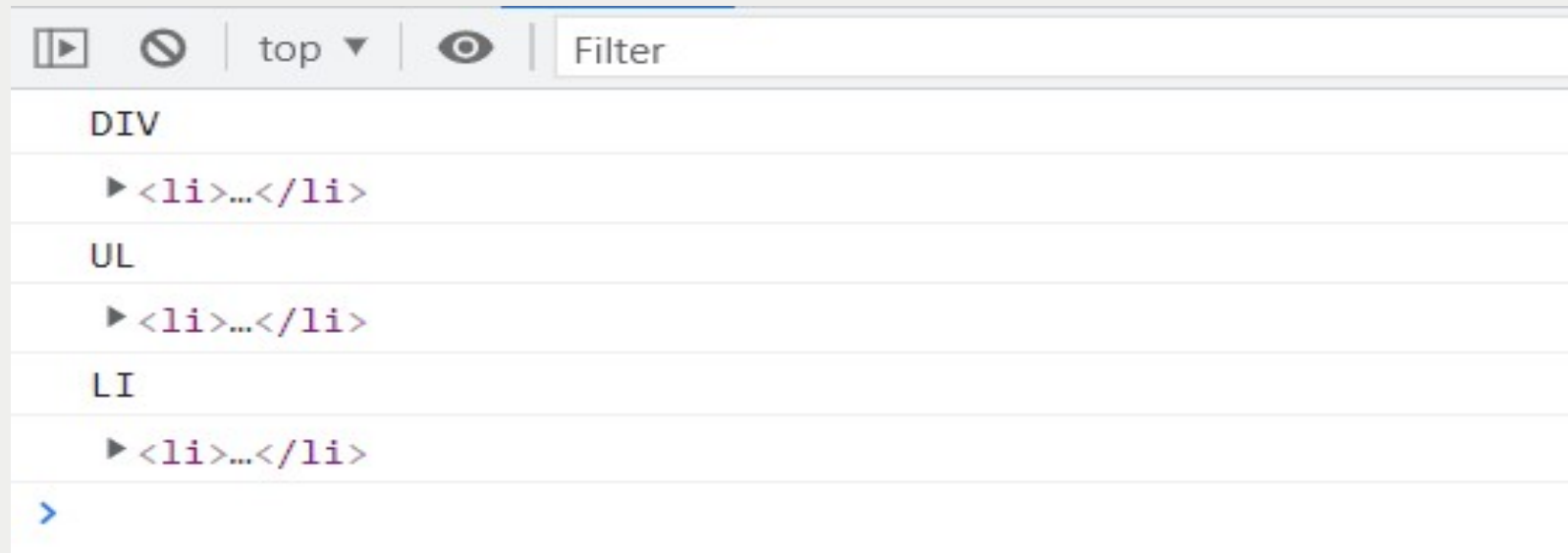
```
li.forEach( element => {  
  element.addEventListener('click', (e) => {  
    console.log("LI");  
    console.log(e.target);  
    e.stopPropagation();  
  });  
});
```



Event Propagation – Capture e Bubbling (default)

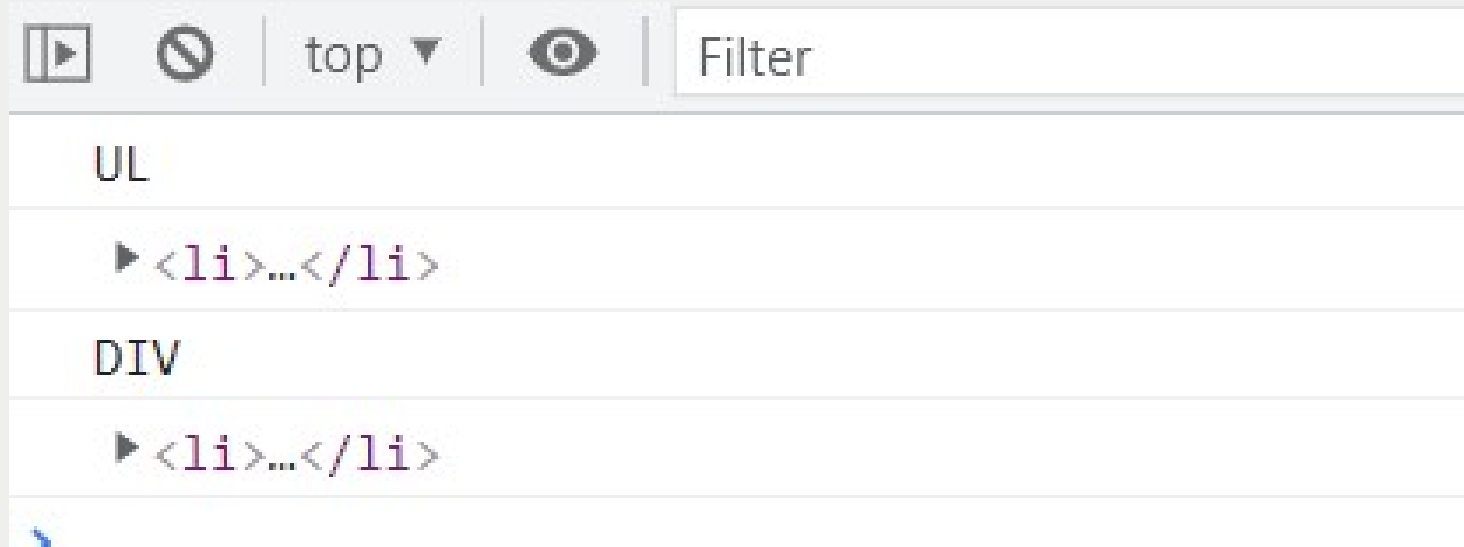
```
div.addEventListener('click', (e) => {  
    console.log("DIV");  
    console.log(e.target);  
}, true);  
  
ul.addEventListener('click', (e) => {  
    console.log("UL");  
    console.log(e.target);  
}, true);  
  
li.forEach( element => {  
    element.addEventListener('click', (e) => {  
        console.log("LI");  
        console.log(e.target);  
    }, true);  
})
```

Event Propagation – Capture e Bubbling (default)



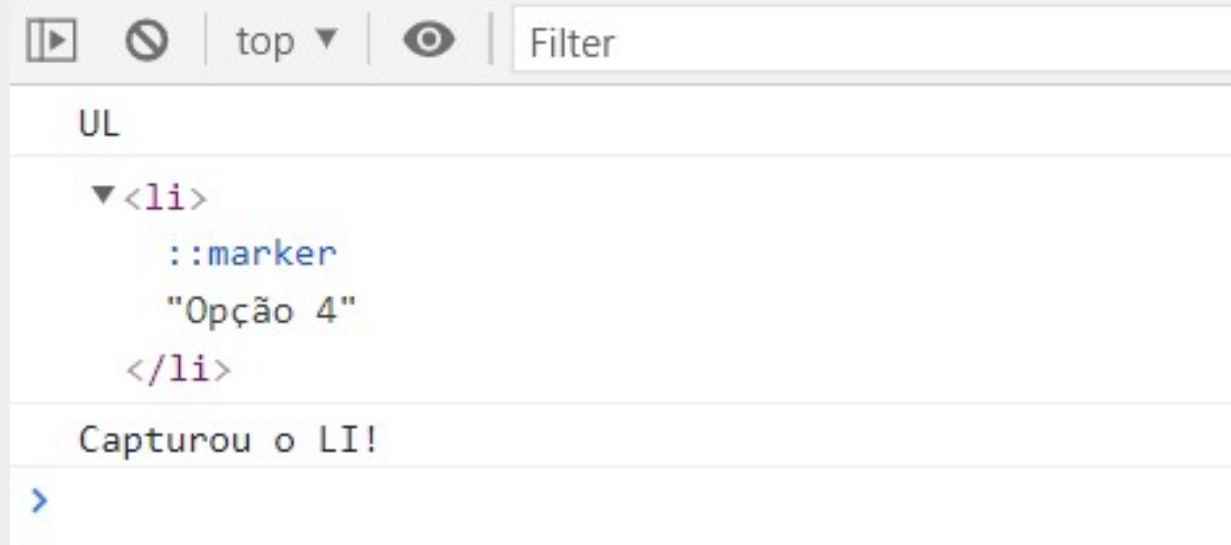
Event Propagation – Capture e Bubbling (default)

```
const element = document.createElement('li');  
element.textContent = "Opção 4";  
ul.append(element);
```



Event Propagation – Capture e Bubbling (default)

```
ul.addEventListener('click', (e) => {  
  console.log("UL");  
  console.log(e.target);  
  if (e.target.tagName == "LI")  
    console.log("Capturou o LI!");  
});
```



Forms and Events

```
<body>
  <form action="">
    <input type="text" id="nome" placeholder="Username">
    <input type="submit" value="Enviar">
  </form>
  <script>
    const form = document.querySelector('form');

    //O evento é na form e não no clique do botão!
    form.addEventListener('submit', (e) => {
      //Parar a execução normal que seria definida no action da form
      e.preventDefault();

      //Notação mais simples.. pode ser pelo id ou pelo name
      console.log(form.nome.value);

    });
  </script>
</body>
```

Forms and Events

```
<body>
  <form action="">
    <input type="text" id='nome' placeholder="Username">
    <input type="submit" value="Enviar">
  </form>
  <script>
    const form = document.querySelector('form');

    //O evento é na form e não no clique do botão!
    form.addEventListener('submit', (e) => {
      //Parar a execução normal que seria definida no action da form
      e.preventDefault();

      //Notação mais simples.. pode ser pelo id ou pelo name
      console.log(form.nome.value);

    });
  </script>
</body>
```

Forms and Events - Regex

https://pt.wikipedia.org/wiki/Express%C3%A3o_regular

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Regular_Expressions

<https://regex101.com/>

https://www.youtube.com/playlist?list=PL4cUxeGkcC9g6m_6Sld9Q4jzqdqHd2HiD

<https://medium.com/factory-mind/regex-cookbook-most-wanted-regex-aa721558c3c1>

Forms and Events – Validação no Submit

```
<form action="">
  <input type="text" id="nome" placeholder="Username">
  <input type="submit" value="Enviar">
</div></div>
</form>
<script>
  const form = document.querySelector('form');
  const div = document.querySelector('div');
  //Letras minúsculas e números entre 6 a 10
  const patternNome = /^[a-z0-9]{6,10}$/;

  //O evento é na form e não no clique do botão!
  form.addEventListener('submit', (e) => {
    e.preventDefault();
    if (patternNome.test(form.nome.value))
      console.log(div.innerText="Nome válido");
    else
      console.log(div.innerText="O nome tem de ter apenas letras e número
entre 6 a 10");
  });
</script>
```

Forms and Events – Validação no keyup

```
form.nome.addEventListener('keyup', (e) => {  
  console.log(e.target);  
  if (patternNome.test(e.target.value)) {  
    e.target.setAttribute('class', 'success');  
  } else {  
    e.target.setAttribute('class', 'error');  
  }  
});
```

EXERCÍCIOS

Desafios

TODO List