Tratamento de eventos em JavaScript – forma indicada

- Em JavaScript, é possível associarmos qualquer evento a qualquer elemento de uma página web por meio do seguinte comando **addEventListener("evento", manipuladorDoEvento)**;
- addEventListener é um método, com os parâmetros especificados acima. Este método deve estar associado a um objeto (por meio do DOM) que representa qualquer elemento da página web;
- O argumento "evento" representa qualquer evento suportado pelo elemento HTML, como, click, onkeypress, onkeydown, load, submit, etc.
- O argumento manipuladorDoEvento representa uma função, que irá executar determinado código quando o evento for disparado;
- Saiba mais em https://www.w3schools.com/jsref/met_document_addeventlistener.asp

Literais de função

- Em JavaScript, um manipulador de eventos pode ser representado por uma literal de função. Uma literal de função age como se fosse uma função regular da linguagem, tendo como, principal diferença, o fato de não termos a presença de parênteses após o nome da função. Veja a seguir:

  Botao.addEventListener("click", calcularMediadeNotas);

- A linha acima especifica que um objeto botão foi criado para representar um controle de formulário do tipo botão, e associar o método a este botão. O JavaScript, então, fica permanentemente à escuta, para verificar se o evento clique foi disparado sobre o botão. Assim que isto ocorre (e só então), é que o código dentro da função calcularMediaDeNotas (cuja declaração estará em outra parte do arquivo) será executado pelo interpretador da linguagem;
- Notar que, se houvesse parênteses após o nome da função, o seu código seria executado imediatamente pelo JavaScript, sem aguardar o disparo no evento no botão;
- **Botao.addEventListener("click", calcularMediadeNotas( ));**

Funções anônimas

- Em JavaScript, um manipulador de eventos pode ser representado por uma função anônima. Como o próprio nome aponta, uma função anônima não tem nome e, portanto, não pode ser invocada e nem utilizada em outro local do código, que não seja aquele onde foi feita a associação do evento. Veja a seguir:

```
Botao.addEventListener("click", function()
        {
        //declaração do corpo da função;
        });
```

- Os dois métodos para manipulação de funções tem sua aplicabilidade em determinadas circunstâncias, com vantagens e desvantagens. Note que uma literal de função não permite a passagem de parâmetros, mas uma função anônima permite. Por outro lado, uma função anônima não pode ser invocada em qualquer outra parte do código, mas uma literal de função pode, e assim, por diante;
- Na próxima fase, daremos especial atenção a outro tipo de função muito importante da linguagem: função flecha.