BOAS PRÁTICAS DE PROGRAMAÇÃO

Boas práticas de programação é um tema bem discutido e importante, alguns diriam de forma cômica que boas práticas é o que difere os homens dos meninos, ou como diria a maioria dos programadores inclusive meu professor Paulo: “as boas práticas é o que difere o bom programador do mau programador”, e não existe explicação melhor.

Boas práticas nada mais é do que uma gama de “técnicas” que faz nosso código ficar mais legível e de fácil compreensão. Nesse tutorial vou falar e ensinar um pouco sobre as principais técnicas que seriam:

1 – Indentação hierárquica do código

Indentar o código é o primeiro passo para usar as boas práticas de programação. Indentar o código ajuda a mapear as estruturas mais facilmente, pois quando indentamos estamos ressaltando as estruturas.

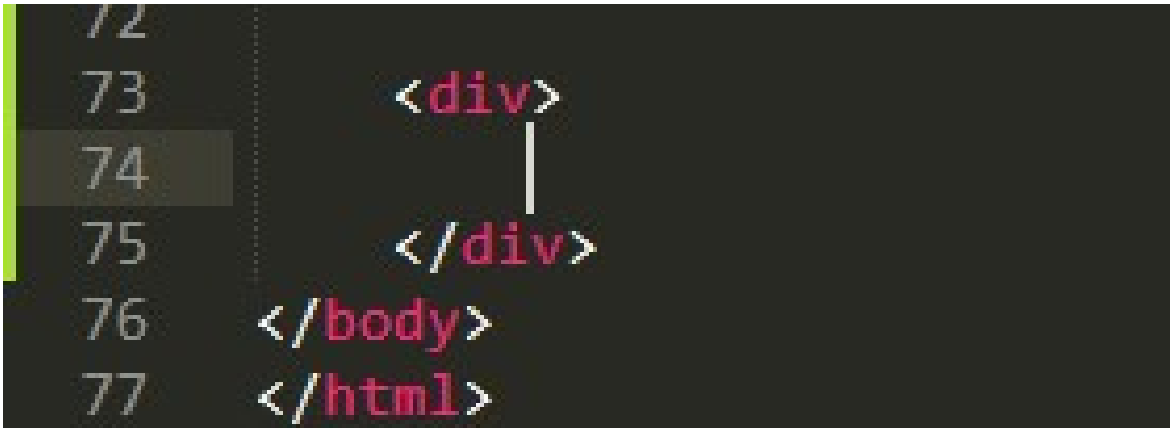
Em seu código busque sempre deixar as tags iguais na mesma margem de parágrafo. Conforme o exemplo de código em HTML:

```
41
42      <div id="VelocidadeMedia">
43          <h1 id="resultado2">Velocidade Média</h1>
44          <input type="text" placeholder="Variação da Velocidade" id="delta1">
45          <input type="text" placeholder="Variação do tempo" id="delta2">
46          <input type="button" value="Calcular" onclick="calcvelmedia()">
47      </div>
48
49      <div id="Velocidadefinal">
50          <h1 id="resultado3">Velocidade Final</h1>
51          <input type="text" placeholder="Posição Inicial" id="vinicial">
52          <input type="text" placeholder="Aceleração" id="vaceleracao">
53          <input type="text" placeholder="Variação do Espaço" id="vespaco">
54          <input type="button" value="Calcular" onclick="calcvelfinal()">
55      </div>
56
57      <div id="Aceleracao">
58          <h1 id="resultado4">Aceleração</h1>
59          <input type="text" placeholder="Variação de espaço" id="VespacoAC">
60          <input type="text" placeholder="Variação de tempo" id="VtempoAC">
61          <input type="button" value="Calcular" onclick="Calcacele()">
62      </div>
63
64       <div id="FHP">
65          <h1 id="resultado5">Função horária da posição</h1>
66          <input type="text" placeholder="Posição Inicial" id="SO">
67          <input type="text" placeholder="Velocidade Inicial" id="VO">
68          <input type="text" placeholder="Aceleração" id="A">
69          <input type="text" placeholder="Tempo" id="T">
70          <input type="button" value="Calcular" onclick="calcfuncao()">
71      </div>
```

Percebam que a “div” está um pouco atrás dos “inputs” e do “h1”, isso para indicar que o “h1” e os “inputs” pertencem as suas respectivas ”div’s”, isso seria dispor o código de forma hierárquica.

2 – Seguir IDE

Seguir seu editor de texto ou IDE é uma ótima dica para quem esta iniciando nesse mundo de programação, o Sublime Text (editor de texto), por exemplo, se dermos espaços entre tags ele já direciona o cursor de forma hierárquica;



Notem que o cursor está um pouco à frente da “div”, o que indica que o que escrever ali pertence a essa “div”.

3 - Espaçamento

Os espaçamentos principalmente entre cálculos são essenciais para facilitar a visualização rápida do código.

Colocar espaços entre símbolos de cálculo (/, *, =, +) e variáveis nos ajuda muito quando estamos corrigindo erros comuns, quando há esses espaçamentos achamos o erro rápido e arrumamos de prontidão.

Nesse exemplo em Java podemos ver os sinais com um espaço entre os números, parênteses ou variáveis, notem como fica fácil a visualização dos valores:

```
function calcvelfinal() {
    vinicial = parseInt(document.getElementById("vinicial").value);
    acele = parseInt(document.getElementById("vaceleracao").value);
    vespaco = parseInt(document.getElementById("vespaco").value);
    vfinal =  Math.sqrt((vinicial * vinicial) + 2 * (acele) * vespaco);
    document.getElementById('resultado3').textContent = vfinal;
}
```

Agora o mesmo código, porém sem espaçamento:

```
function calcvelfinal() {
    vinicial = parseInt(document.getElementById("vinicial").value);
    acele = parseInt(document.getElementById("vaceleracao").value);
    vespaco = parseInt(document.getElementById("vespaco").value);
    vfinal =  Math.sqrt((vinicial*vinicial)+2*(acele)*vespaco);
    document.getElementById('resultado3').textContent = vfinal;
}
```

Notem como dificulta o entendimento pois o código com espaçamento basta você olhar para entender.

4 – Comentários

Os comentários são anotações por todo o código que podem explicar o que foi feito em determinada parte do código ou no inicio do código para dizer qual o objetivo daquele programa e assim por diante. É importante frisar que os comentários devem ser curtos e objetivos, a intenção do comentário não é que você leia 5, 6 linhas, e sim uma ou duas e entenda o que certa parte do código faz.

Cada linguagem de programação tem sua sintaxe de comentar no código aqui vai alguns exemplos:

Em Java, C++ e C# usamos // para comentários de apenas uma linha e para mais de uma linha: /*conteúdo*/

Em Phyton são iniciados pelo caractere #, e se estendem até o final da linha física.

Para comentar qualquer código, basta buscar a sintaxe e aplicar ao código sem muito mistério, o que você escrever no comentário será considerado caracteres que o programa deve ignorar. Vejamos o código em Java contendo anotações com // e /*conteúdo*/:

```
//declarando variaveis
var vinicial;
var acele;
var vespaco;
var vfinal;

function calcvelfinal() {
    vinicial = parseInt(document.getElementById("vinicial").value);
    acele = parseInt(document.getElementById("vaceleracao").value);
    vespaco = parseInt(document.getElementById("vespaco").value);
    /*Apliquei a fórmula de calcular velocidade média, indiquei com o uso de parenteses quais contas eram prioridadea, porém ainda preciso
    conerir valor da conta e ajustar css */
    vfinal =  Math.sqrt((vinicial*vinicial)+2*(acele)*vespaco);
    document.getElementById('resultado3').textContent = vfinal;
}
```

Percebam que no segundo comentário nesse código em Java, é um comentário que posteriormente pode ser apagado, como trabalhamos em um código por mais de um dia, podemos colocar comentários para não esquecer de revisar código ou cálculo, lembrar de estilizar partes especificas, anotar no que será dado continuidade e assim por diante.

5 - Nomeando variáveis

Nomear variáveis quando se inicia a programar é sempre uma incógnita, não sabemos muito como nomear e acabamos nomeando de uma forma não tão legal, muitos programadores criam sua própria técnica conforme vão ganhando experiência, mas para quem ainda não ganhou essa habilidade aqui vai alguns preceitos que podem ser seguidos na hora de nomear uma variável:

- Escolha nomes curtos.

Os nomes das variáveis não devem ser muitos extensos, mas devem ser descritivos. Quando o nome do valor que será atribuído à variável for curto, opte por colocar ele mesmo. Um exemplo: eu tenho o valor ”idade” que precisa ser atribuído a uma variável, ao invés de criar um nome eu coloco simplesmente “idade”, por si só é um nome curto e bem descritivo que não precisa de alterações.

- Abrevie nomes longos, mas faça uso do recurso quatro.

Quando temos um nome longo para atribuir à uma variável é preferível abreviá-lo, mas ao deixe de colocar um comentário informando o nome por inteiro da variável.

- Não nomeie variáveis apenas com uma letra (n, s, u, v)

Não atribua nomes para variáveis que sejam apenas uma letra, isso complica na busca da variável, por ela ficar escondida em meio ao código, além de que não fica nada descritivo.

- Não use Notação Húngara

Notação Húngara resumidamente é informar o tipo da variável no nome dela, por exemplo: temos a variável “valorjoia” e ela é um valor inteiro, usando a notação húngara ficaria “valorjoiaint” , além de ser inútil e dar volume desnecessário ao código, dificulta a pronúncia.

6 - Mau uso de tags

O mau uso de tags ou símbolos (^, /, *, +) seria as famosas gambiarras, funciona, porém o código não fica legível e se for um trabalho de escola, faculdade ou um serviço monetizado, pode ser considerado um erro, por mais que funcione.

Chegamos ao fim do tutorial e quero dizer para não se assustarem, boas práticas é adaptação, é normal errar, só não pode permanecer no erro, com o tempo fazer o uso das boas práticas vai se tornar hábito e irá facilitar sua vida!