

Fácil Aprendizagem JavaScript



Iniciantes aprendem melhor Javascript

Fácil Aprendizagem JavaScript



YANG HU

Aprenda fácil, rápido e bem.

<http://br.verejava.com>

Copyright © 2020 Yang Hu

All rights reserved.

ISBN: 9798571864657

CONTEÚDO

1. [Conceito Básico de JavaScript](#)
 - 1.1 [JavaScript HelloWorld](#)
 - 1.2 [Variável](#)
 - 1.3 [Tipos de Dados Básicos](#)
2. [Operador Aritmético](#)
 - 2.1 [Operador Aritmético](#)
 - 2.2 [Função](#)
 - 2.3 [Classe](#)
 - 2.4 [Conversão de Tipo](#)
 - 2.5 [DOM](#)

- 2.6 [Evento](#)
 - 2.7 [Crie uma Calculadora da Web](#)
 - 2.8 [Operador de Atribuição](#)
 - 2.9 [Operador Relacional](#)
 - 2.10 [If](#)
 - 2.11 [Operadores Lógicos](#)
 - 3. [Declaração de Controle](#)
 - 3.1 [Switch](#)
 - 3.2 [While Loop](#)
 - 3.3 [For Loop](#)
 - 4. [Matriz](#)
 - 4.1 [Matriz Unidimensional](#)
 - 4.2 [Selecionar Tudo Caixa de Seleção](#)
 - 4.3 [Matriz Bidimensional](#)
 - 4.4 [Jogo Find Dog.](#)
 - 4.5 [Lista Suspensa de Ligação Secundária](#)
 - 5. [Mais Eventos](#)
 - 5.1 [Página de Login da Web](#)
 - 5.2 [MouseOver Miniatura Para](#)
- [Maior](#)
- 6. [CSS](#)
 - 6.1 [CSS Dentro do Estilo da Tag](#)
 - 6.2 [Estilo CSS Inline](#)
 - 6.3 [Estilo de Importação CSS](#)
 - 6.4 [Estilo de Referência de Classe](#)
 - 6.5 [Estilo de Referência de ID](#)

- 6.6 [Estilo de Fonte CSS](#)
- 6.7 [CSS Estilo de Fundo](#)
- 6.8 [CSS Estilo de Borda](#)
- 6.9 [Estilo de Espaçamento CSS](#)
- 6.10 [Margens Internas e Externas de CSS](#)
- 6.11 [Estilo de Lista CSS](#)
- 6.12 [CSS Float](#)
- 6.13 [Estilo de Posição CSS](#)
- 6.14 [Estilo de Visibilidade CSS](#)
- 6.15 [Estilo de Associação CSS](#)
- 6.16 [Estilo de Hiperlink CSS](#)
- 6.17 [CSS Estilo União](#)
- 7. [DOM e Hierarquia de Elemento](#)
 - 7.1 [Hierarquia de Elemento](#)
 - 7.2 [Criar Nó de Texto](#)
 - 7.3 [Excluir Nó](#)
 - 7.4 [Substituir Nó](#)
 - 7.5 [Adicionar Exemplo de Contato](#)
 - 7.6 [Fonte de Estilo CSS](#)
 - 7.7 [CSS Alterar Seletor de Classe](#)
 - 7.8 [Expandir e Fechar CSS Overflow](#)
 - 7.9 [Destaque de CSS Flutuante](#)
 - 7.10 [Tabela Criar Colunas de Linhas](#)
 - 7.11 [Excluir Coluna da Linha da Tabela](#)
- 8. [Timer](#)
- 9. [Animação da Barra de Progresso da Tela](#)

- 9.1 [Desenhe Uma Linha na Tela](#)
- 9.2 [Desenhe um Retângulo na Tela](#)
- 9.2 [Desenhar Barra de Progresso](#)
- 10. [Tela de Pintura Ball Bounce](#)
 - 10.1 [Desenhe um Círculo na Tela](#)
 - 10.2 [Crie uma Função drawBall](#)
 - 10.3 [Crie uma Classe Bola Para Desenhar](#)
 - 10.4 [Crie uma Bola e Depois Desça](#)
 - 10.5 [A Bola Quica Para Cima e Para Baixo](#)
- 11. [Javascript ES6 Programação Orientada a Objetos](#)
 - 11.1 [Criar Classe](#)
 - 11.2 [Inheritance](#)
 - 11.3 [Sobrepor](#)
- 12. [Jogo de Avião](#)
 - 12.1 [Criar Classe de Avião](#)
 - 12.2 [Evento Keypress](#)
 - 12.3 [Mover Avião na Tela](#)
 - 12.4 [Avião Dispara uma Bala](#)
 - 12.5 [Avião Inimigo se Move Para Baixo](#)
 - 12.6 [A Bala Colidiu Com o EnemyPlane](#)

JavaScript HelloWorld

JavaScript é uma linguagem interativa dinâmica de página da web usada no desenvolvimento da web.

1. Crie um arquivo **helloworld.html** com o Bloco de notas e abra-o em seu navegador para ver a página da web

<html>: tag informa ao navegador que este é um documento HTML.

<head>: tag é um contêiner para metadados sobre conteúdo e conjunto de caracteres.

<title>: tag define o título do documento.

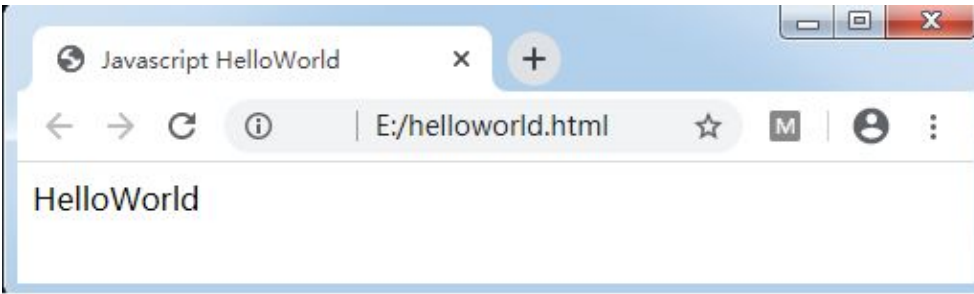
<body>: tag define o corpo do documento.

<script>: tag é usada para definir um script do lado do cliente (JavaScript).

document.write(): grava conteúdo HTML para documento no navegador

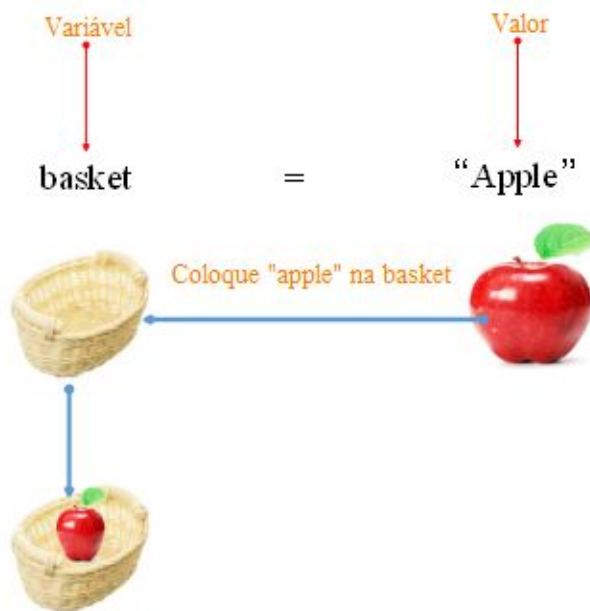
```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>First Web Page</title>
</head>
<body>
  <script type="text/javascript">
    document.write("HelloWorld");
  </script>
</body>
</html>
```

Resultado:



Variável (**Variable**)

Variável (Variable**)**: uma área de memória alocada pelo sistema para armazenar dados.



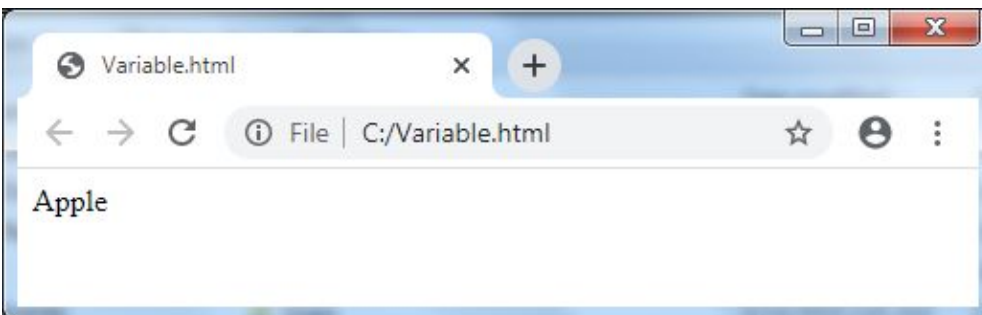
1. Crie um arquivo: **Variable.html** e abra-o em seu navegador

// : comentários únicos não são executados por javascript

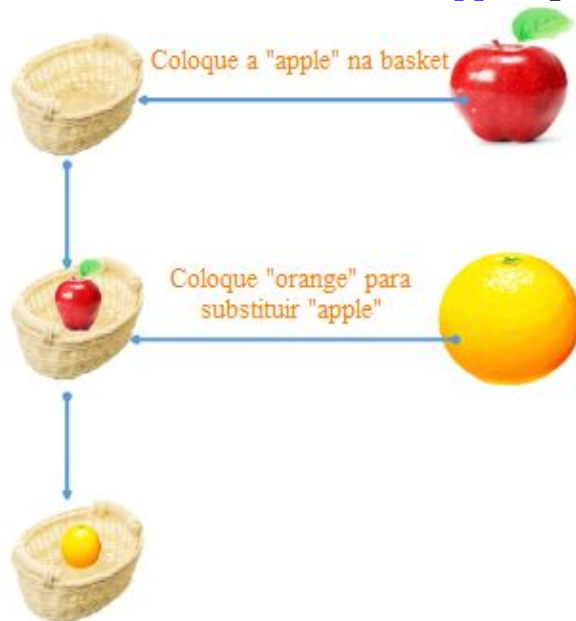
"": tipo de String

```
<script type="text/javascript">  
  var basket = "Apple";  
  document.write(basket);  
</script>
```

Resultado:



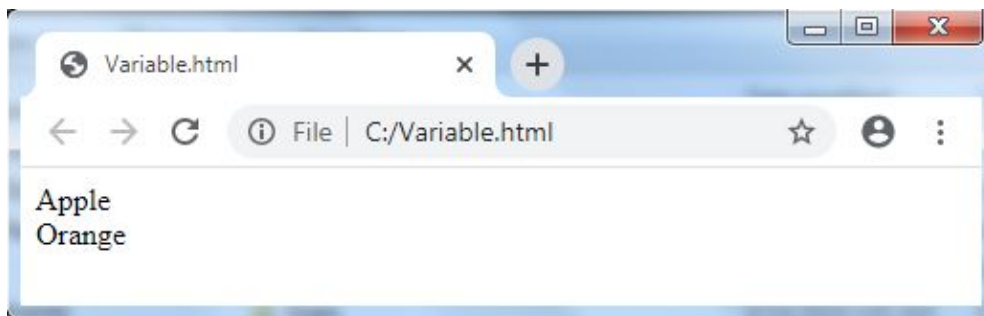
2. Substitua o valor da **basket** variável de "Apple" para "Orange";



**
**: tag html envolve uma nova linha

```
<script type="text/javascript">  
  var basket = "Apple";  
  document.write(basket);  
  
  document.write("<br>");  
  
  basket = "Orange";  
  document.write(basket);  
</script>
```

Resultado:



Tipos de Dados Básicos

Javascript é uma linguagem de tipo dinâmico

1. Crie um arquivo: **var.html**

typeof(): encontre o tipo de uma variável JavaScript

```
<script type="text/javascript">
  var x;           //undefined
  var a=9;         //number
  var b=1.5;       //number
  var c=true;      //Boolean
  var d="abc";     //string
  var e='abc';     //string

  document.write(typeof(x) + "<br>");
  document.write(typeof(a) + "<br>");
  document.write(typeof(b) + "<br>");
  document.write(typeof(c) + "<br>");
  document.write(typeof(d) + "<br>");
  document.write(typeof(e) + "<br>");
</script>
```

Resultado:

undefined
number
number
boolean
string
string

Operador Aritmético



Operador Aritmético: adicionar +, menos -, multiplicar *, divisível /, obter módulo %

1. Crie um arquivo:

```
<script type="text/javascript">
  var a = 1;
  var b = 2;
  var c = 3;
  document.write(a + b); // = 3
  document.write("<br>");

  document.write(a - b); // = -1
  document.write("<br>");

  document.write(a * b); // = 2
  document.write("<br>");

  document.write(b / a); // = 2
  document.write("<br>");

  document.write(c % b); // = 1 // resto número após a / b
  document.write("<br>");
```

</script>

Função (Function)



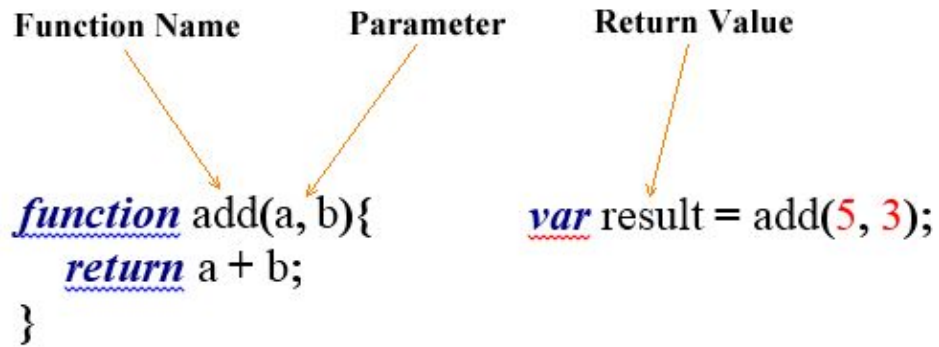
```
function add(a, b){  
  return a + b;  
}
```

→ var result = add(4, 2);



```
function add(a, b){  
  return a + b;  
}
```

→ var result = add(5, 3);



1. Crie um arquivo: **Function.html**

```
<script type="text/javascript">  
  function add(a, b){  
    return a + b;  
  }  
  //////////// test ////////////  
  var result = add(4, 2);  
  document.write(result); // 6  
  
  document.write("<br>");  
  
  var result = add(5, 3);  
  document.write(result) // 8  
</script>
```

Resultado:

6
8

2. Adicione mais 3 funções sobre: -, *, /

```
<script type="text/javascript">
  function add(a, b){
    return a + b;
  }

  function sub(a, b){
    return a - b;
  }

  function multiply(a, b){
    return a * b;
  }

  function divide(a, b){
    return a / b;
  }

  //////////// test ////////////
  var result = add(4, 2);
  document.write(result); // 6

  document.write("<br>");

  var result = sub(4, 2);
  document.write(result) // 2

  document.write("<br>");

  var result = multiply(4, 2);
  document.write(result) // 8

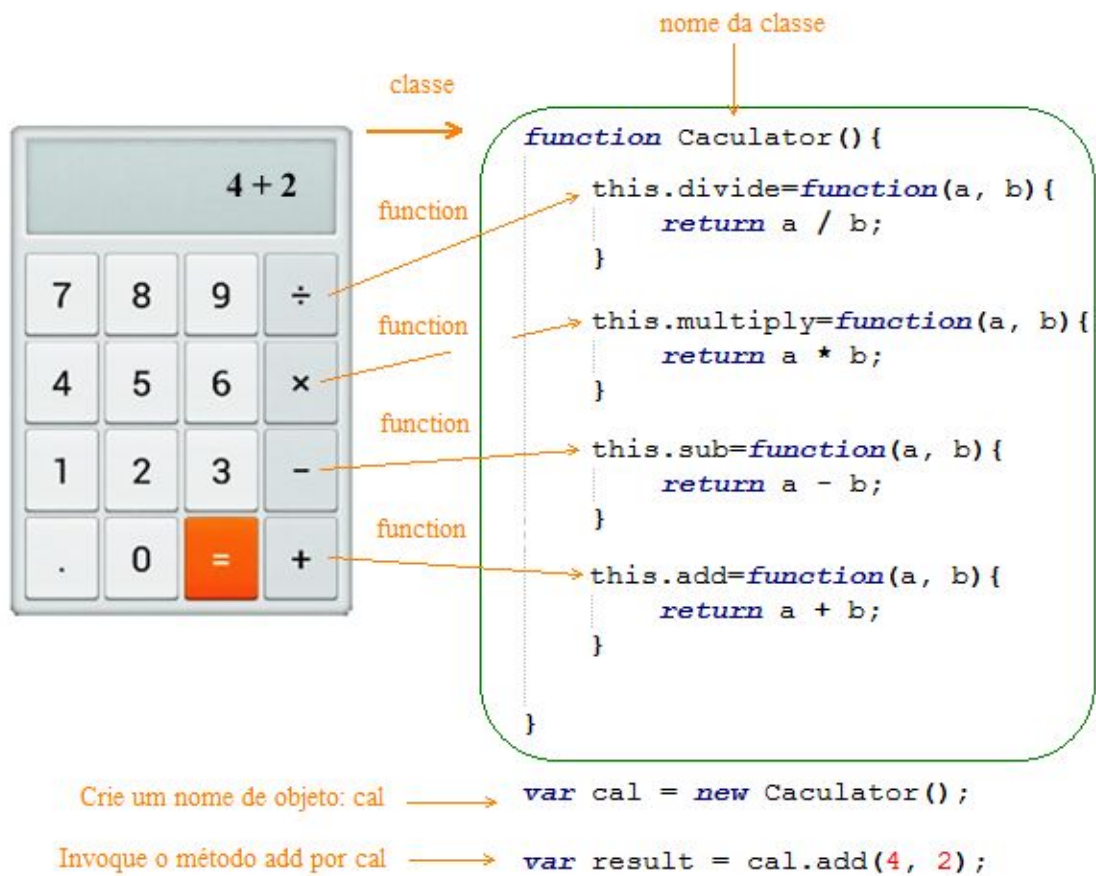
  document.write("<br>");

  var result = divide(4, 2);
  document.write(result) // 2
```

```
</script>
```


Classe (Class)

Classe (class): é um projeto ou protótipo definido pelo usuário a partir do qual os objetos são criados.



1. Crie um arquivo: **Caculator.html**

```
<script type="text/javascript">

    function Caculator(){

        this.add=function(a, b){
            return a + b;
        }

        this.sub=function(a, b){
            return a - b;
        }

        this.multiply=function(a, b){
            return a * b;
        }

        this.divide=function(a, b){
            return a / b;
        }
    }

    //////////// test ////////////

    var cal = new Caculator();

    var result = cal.add(4, 2);
    document.write(result); // 6

    var result = cal.sub(4, 2);
    document.write(result) // 2

    var result = cal.multiply(4, 2);
    document.write(result) // 8

    var result = cal.divide(4, 2);
    document.write(result) // 2
```

</script>

Conversão de Tipo

1. Crie um arquivo: **TypeConversion.html**

parseInt(): esta é a função padrão do javascript pode converter string em inteiro.

```
<script type="text/javascript">
  var a="1";
  var b="2";

  var result = a + b; // + concatenar duas strings
  document.write(result); // = 12

  document.write("<br>");

  result = parseInt(a) + parseInt(b);
  document.write(result); // = 3
</script>
```

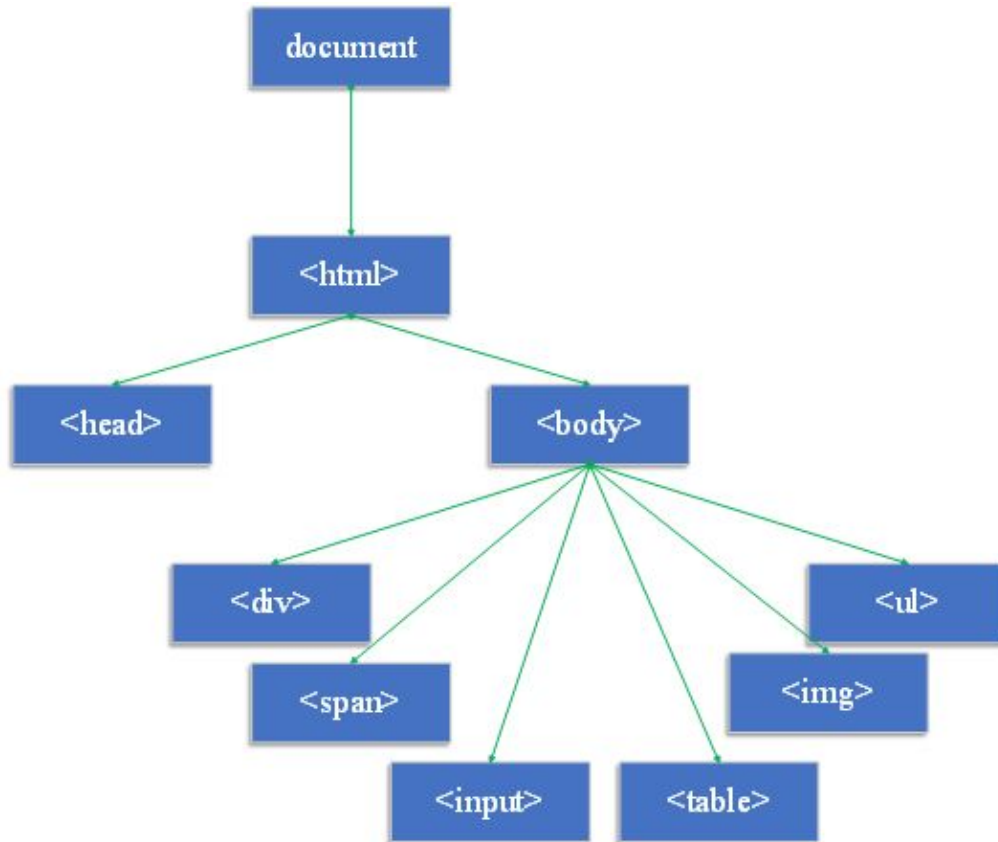
Resultado:

12
3

DOM

DOM: Em Javascript, cada elemento HTML é um objeto de classe.

Diagrama da estrutura do modelo de objeto do documento:



1. Crie um arquivo: **getElementById.html**

document.getElementById(): é um método DOM para obter o elemento por atributo de ID.

alert(): é uma função para abrir uma janela de informações.

Obj.innerHTML: define ou retorna o conteúdo HTML interno de um elemento.

<div>: tag define uma divisão ou seção em um documento HTML.

```
<div id="word">Faith and Hope</div>
```

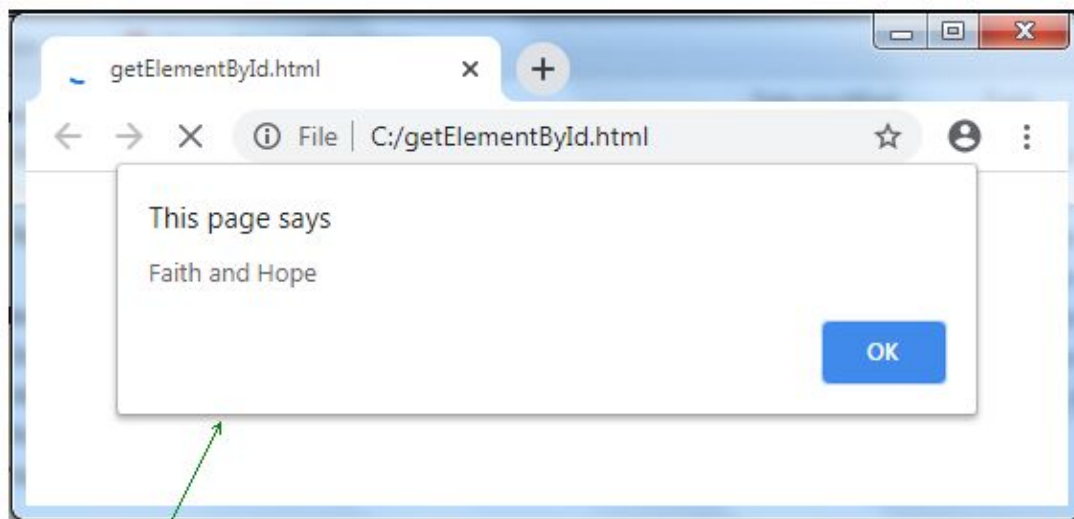
```
<script type="text/javascript">
```

```
  var divObj = document.getElementById("word");
```

```
  var htmlContent = divObj.innerHTML;
```

```
  alert(htmlContent);
```

```
</script>
```



```
<div id="word">Faith and Hope</div>
```

```
var divObj = document.getElementById("word");
```

```
var htmlContent = divObj.innerHTML;
```

```
alert(htmlContent);
```

2. Alterar um arquivo: **getElementById.html**

`<input type="text" />`: A tag html especifica um campo de entrada onde o usuário pode inserir dados.

```
<input type="text" id="username" value="Joseph" />
```

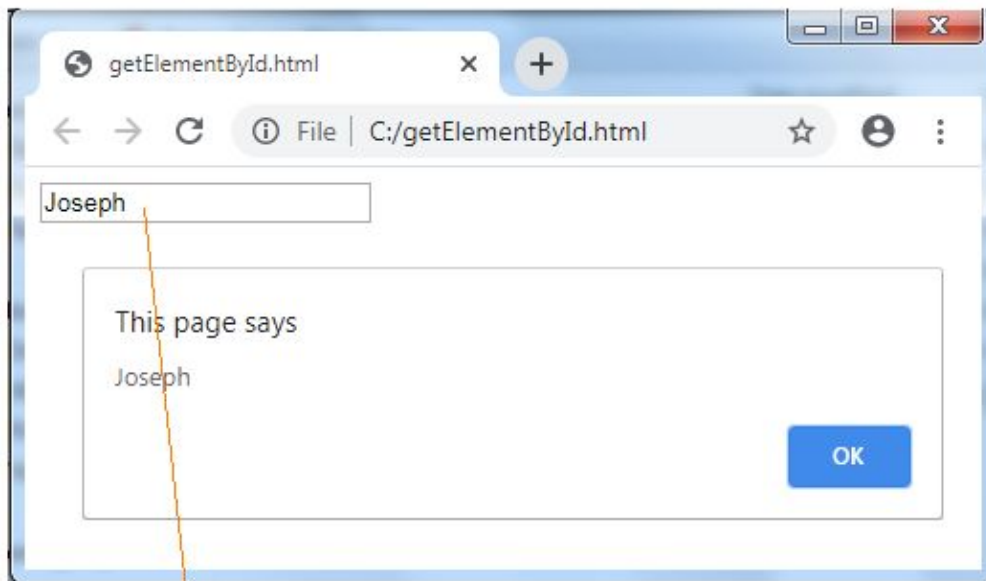
```
<script type="text/javascript">
```

```
  var divObj = document.getElementById("username");
```

```
  var value = divObj.value;
```

```
  alert(value);
```

```
</script>
```



```
<input type="text" id="username" value="Joseph" />
```

```
var divObj = document.getElementById("username");  
var value = divObj.value;  
alert(value);
```

Evento (Event)

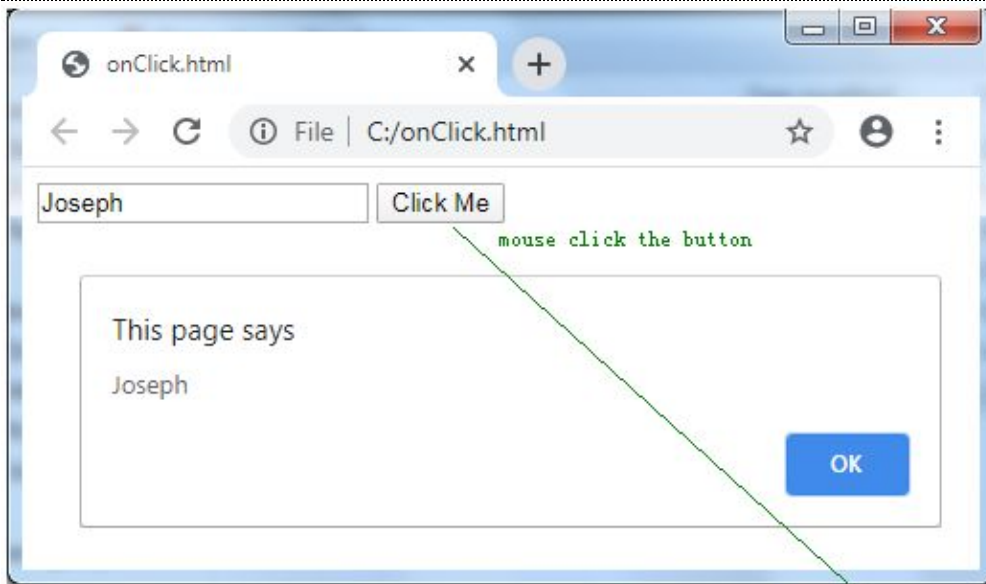
1. Crie um arquivo: **onClick.html**

onClick: quando o mouse clica, o botão chama uma função.

<input type="button" />: A tag html define um botão clicável.

```
<input type="text" id="username" value="Joseph" />
<input type="button" value="Click Me" onClick="doClick()" />

<script type="text/javascript">
  function doClick(){
    var divObj = document.getElementById("username");
    var value = divObj.value;
    alert(value);
  }
</script>
```

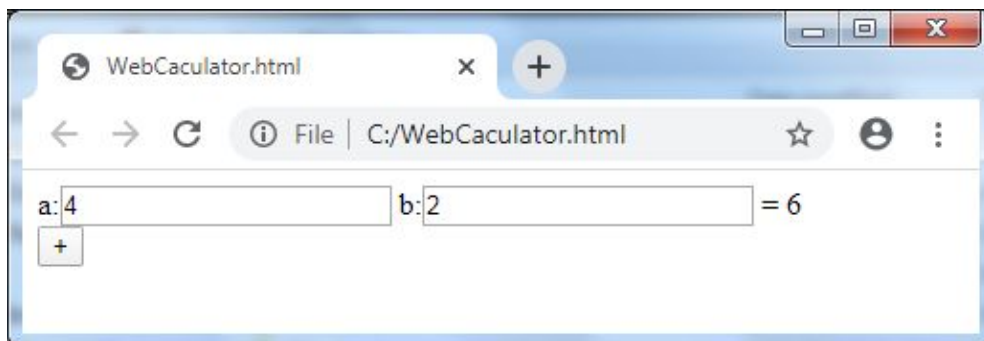


```
<input type="text" id="username" value="Joseph" />
<input type="button" value="Click Me" onClick="doClick()" />

function doClick(){
  var divObj = document.getElementById("username");
  var value = divObj.value;
  alert(value);
}
```


Crie uma Calculadora da Web

1. Crie um arquivo: **WebCaculator.html**



```
a:<input type="text" id="a" value="" />
b:<input type="text" id="b" value="" />
=
<span id="result"></span>
<br>
<input type="button" value="+" onClick="doAdd()" />

<script type="text/javascript">
  function doAdd(){
    var aObj = document.getElementById("a");
    var bObj = document.getElementById("b");
    var resultObj = document.getElementById("result");
    var a = parseInt(aObj.value); // Converte string em inteiro
    var b = parseInt(bObj.value);
    resultObj.innerHTML = a + b;
  }
</script>
```

2. Adicionar função de 3 mores: -, *, /

```
a:<input type="text" id="a" value="" />
b:<input type="text" id="b" value="" />
=
<span id="result"></span>
<br>
<input type="button" value="+" onClick="doAdd()" />
<input type="button" value="-" onClick="doSub()" />
<input type="button" value="*" onClick="doMultiply()" />
<input type="button" value="/" onClick="doDivide()" />

<script type="text/javascript">
  function doAdd(){
    var aObj = document.getElementById("a");
    var bObj = document.getElementById("b");
    var resultObj = document.getElementById("result");
    var a = parseInt(aObj.value); // Converte string em inteiro
    var b = parseInt(bObj.value);
    resultObj.innerHTML = a + b;
  }

  function doSub(){
    var aObj = document.getElementById("a");
    var bObj = document.getElementById("b");
    var resultObj = document.getElementById("result");
    var a = parseInt(aObj.value); // Converte string em inteiro
    var b = parseInt(bObj.value);
    resultObj.innerHTML = a - b;
  }

  function doMultiply(){
    var aObj = document.getElementById("a");
    var bObj = document.getElementById("b");
    var resultObj = document.getElementById("result");
    var a = parseInt(aObj.value); // Converte string em inteiro
```

```
var b = parseInt(bObj.value);  
resultObj.innerHTML = a * b;  
}
```

```
function doDivide(){  
    var aObj = document.getElementById("a");  
    var bObj = document.getElementById("b");  
    var resultObj = document.getElementById("result");  
    var a = parseInt(aObj.value); // Converte string em inteiro  
    var b = parseInt(bObj.value);  
    resultObj.innerHTML = a / b;  
}  
</script>
```

Clique + botão Resultado:

a: 4 b: 2 = 6
+ - * /

Clique - botão Resultado:

a: 4 b: 2 = 2
+ - * /

Clique * botão Resultado:

a: 4 b: 2 = 8
+ - * /

Clique / botão Resultado:

a: 4 b: 2 = 2
+ - * /

Operador de Atribuição

1. Crie um arquivo: **AssignmentOperator.html**

```
<script type="text/javascript">
  var d = 4;
  document.write(d++); // = 4 imprimir d e então incrementar 1
  document.write("<br>");

  d = 4;
  document.write(++d); // = 5 incrementar 1 e imprimir d
  document.write("<br>");

  d = 4;
  document.write(d--); // = 4 imprima d e depois diminua 1
  document.write("<br>");

  d = 4;
  document.write(--d); // = 3 decrementar 1 e imprimir d

  var result = 10;
  result = result + 1;
  document.write(result); // = 11
  document.write("<br>");

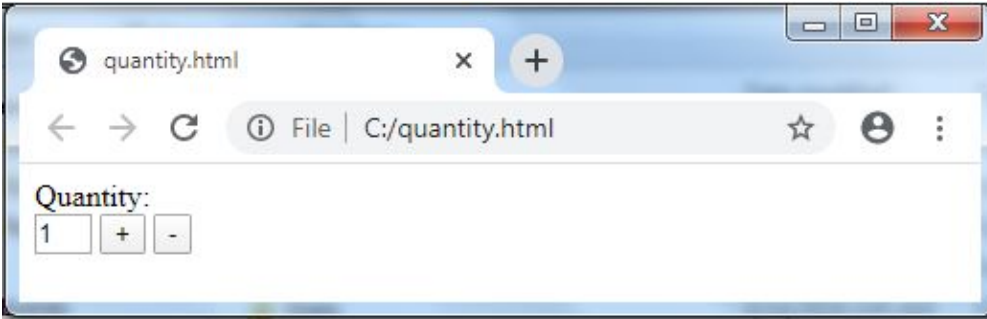
  result = 10;
  result++;
  document.write(result); // = 11
  document.write("<br>");

  result = 10;
  result += 1;
  document.write(result); // = 11
  document.write("<br>");
</script>
```



2. Crie um somador e subtrator de quantidade: **quantity.html**

Clique no botão + para aumentar a quantidade em 1 e clique no botão - para diminuir a quantidade em 1.



Quantity:

<input type="text" style="width:30px" id="quantity" value="1" />

<input type="button" value="+" onClick="doAdd()" />

<input type="button" value="-" onClick="doSub()" />

<script type="text/javascript">

function doAdd(){

var quantityObj = document.getElementById("quantity");

var value = parseInt(quantityObj.value);

value++

quantityObj.value = value;

}

function doSub(){

var quantityObj = document.getElementById("quantity");

var value = parseInt(quantityObj.value);

value--

quantityObj.value = value;

}

</script>

Operador Relacional

1. Crie um arquivo: **RelationalOperator.html**

Operador Relacional: apenas dois resultados: verdadeiro(**true**) ou falso(**false**)

```
<script type="text/javascript">
```

```
document.write(1 > 2); // false  
document.write("<br>");
```

```
document.write(1 >= 1); // true  
document.write("<br>");
```

```
document.write(1 < 2); // true  
document.write("<br>");
```

```
document.write(1 <= 2); // true  
document.write("<br>");
```

```
document.write(1 == 1); // true  
document.write("<br>");
```

```
document.write(1 != 2); // true
```

```
</script>
```

If

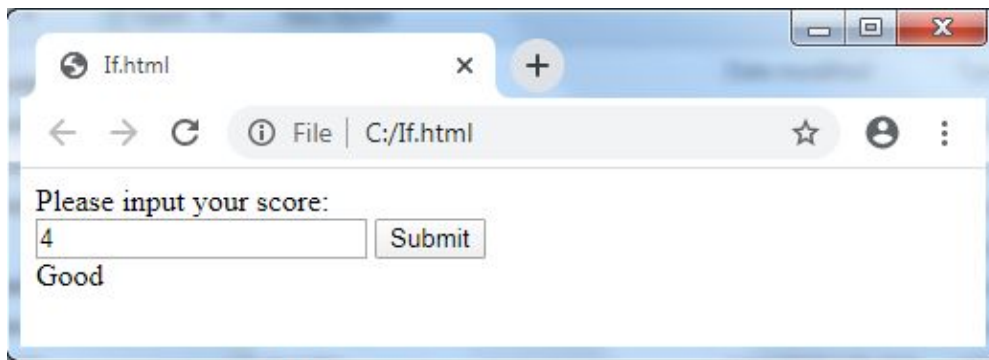
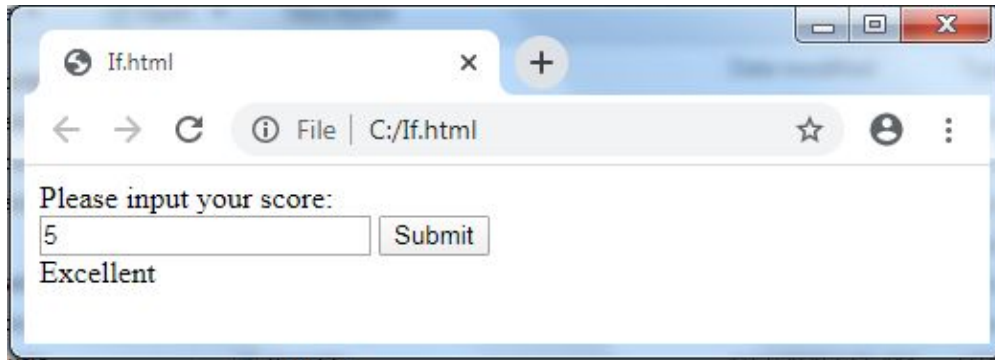
If

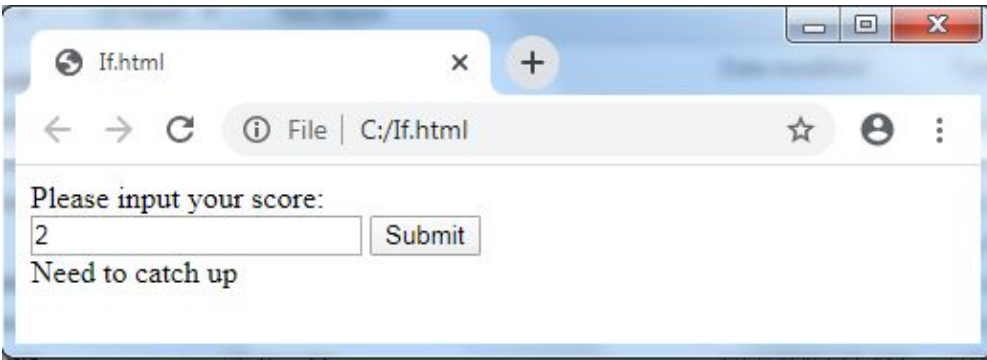
1. `if(expressão){ }`
2. `if (expressão) { } else { }`
3. `if (expressão) { } else if { }`

1. Crie um arquivo: **If.html**

Exemplo de pontuação:

`if score == 5` : Excellent
`else if score == 4`: Good
`else`: Need to catch up





Please input your score:

`
`

`<input type="text" id="score" value="" />`

`<input type="button" value="Submit" onClick="doSubmit()" />`

`
`

``

`<script type="text/javascript">`

`function doSubmit(){`

`var scoreObj = document.getElementById("score");`

`var messageObj = document.getElementById("message");`

`var score = parseInt(scoreObj.value);`

`if (score == 5) {`

`messageObj.innerHTML = "Excellent";`

`} else if (score == 4) {`

`messageObj.innerHTML = "Good";`

`} else {`

`messageObj.innerHTML = "Need to catch up";`

`}`

`}`

`</script>`

Operadores Lógicos

1. Crie um arquivo: **LogicOperator.html**

Operadores Lógicos: e &&, ou ||, não!

1. && retorna **true** se ambos os lados da operação forem **true**, caso contrário, **false**
2. || O resultado é **false** quando ambos os lados da operação são **false**, caso contrário, **true**;
3. ! se retorna **true**, o resultado é **false**, caso contrário é **true**

```
<script type="text/javascript">
  document.write(true && false); // false
  document.write("<br>");

  document.write(false && true); // false
  document.write("<br>");

  document.write(false && false); // false
  document.write("<br>");

  document.write(true && true); // true
  document.write("<br>");

  document.write(true || false); // true
  document.write("<br>");

  document.write(false || true); // true
  document.write("<br>");

  document.write(true || true); // true
  document.write("<br>");

  document.write(false || false); // false
  document.write("<br>");
```

```
document.write(!true); // false  
document.write("<br>");
```

```
document.write(!false); // true  
document.write("<br>");  
</script>
```

2. Crie um arquivo: **LogicOperator2.html**

```
<script type="text/javascript">

    document.write(1>2 && 3>4) // = false
    document.write(2>1 && 3>4) // = false

    document.write("-----")

    document.write(2>1 || 3>4) // = true
    document.write(2>1 || 3>4) // = true
    document.write(1>2 || 3>4) // = false

</script>
```

3. Exemplo de imposto de folha de pagamento:

Tax = salary * rate

level:

500 -- 2000 \$: rate 10%

2000--5000 \$: rate 15%

5000-- 20000 \$: rate 20%

>= 20000\$: rate 30%

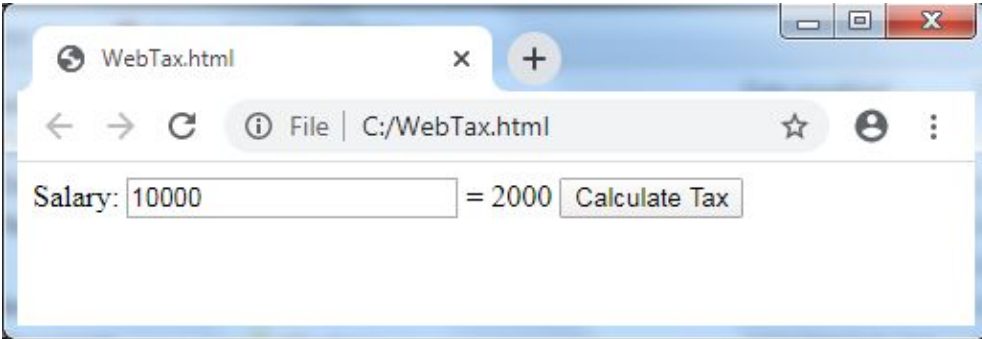
```
<script type="text/javascript">
  var salary = 10000;
  var tax = 0;
  if (salary >= 500 && salary < 2000) {
    tax = salary * 0.1;
  } else if (salary >= 2000 && salary < 5000) {
    tax = salary * 0.15;
  } else if (salary >= 5000 && salary < 20000) {
    tax = salary * 0.2;
  } else {
    tax = salary * 0.3;
  }
  document.write("tax amount=" + tax);
</script>
```

Resultado:

tax amount=2000

2. Crie um arquivo: **WebTax.html**

Insira o salário e clique no botão Calcular imposto



```
Salary: <input type="text" id="salary" value="" />
```

```
=
```

```
<span id="tax"></span>
```

```
<input type="button" value="Calculate Tax" onClick="doTax()" />
```

```
<script type="text/javascript">
```

```
function doTax(){
```

```
    var salaryObj = document.getElementById("salary");
```

```
    var salary = parseInt(salaryObj.value);
```

```
    var tax = 0;
```

```
    if (salary >= 500 && salary < 2000) {
```

```
        tax = salary * 0.1;
```

```
    } else if (salary >= 2000 && salary < 5000) {
```

```
        tax = salary * 0.15;
```

```
    } else if (salary >= 5000 && salary < 20000) {
```

```
        tax = salary * 0.2;
```

```
    } else {
```

```
        tax = salary * 0.3;
```

```
    }
```

```
    document.getElementById("tax").innerHTML = tax;
```

```
}
```

```
</script>
```



Switch

1. Crie um arquivo: **switch.html**

Insira um número 1, 2, 3

1 : Pay by Visa Card

2 : Pay by Master Card

3: Pay by Paypal

Otherwise Pay by face to face

break: termine o código para executar e depois saia.

continue: termine o código para executá-lo e continue o próximo código.

```
<script type="text/javascript">

    var num = 1;

    switch (num) {
        case 1:
            document.write("Pay by Visa Card");
            break; // termine o código para executar
        case 2:
            document.write("Pay by Master Card");
            break;
        case 3:
            document.write("Pay by Paypal");
            break;
        default:
            document.write("Pay by face to face");
    }

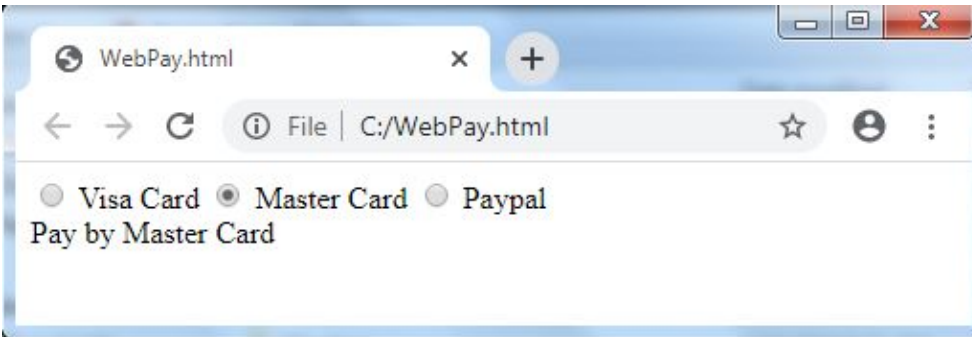
</script>
```

Resultado:

Pay by Visa Card

2. Crie um arquivo: **WebPay.html**

this: objeto do elemento atual



```
<input type="radio" name="card" value="1" onclick="doPay(this)"
/> Visa Card
<input type="radio" name="card" value="2" onclick="doPay(this)"
/> Master Card
<input type="radio" name="card" value="3" onclick="doPay(this)"
/> Paypal
<br>
<span id="result"></span>
```

```
<script type="text/javascript">
```

```
function doPay(obj){
    var num = parseInt(obj.value);
    var resultObj = document.getElementById("result");

    switch (num) {
        case 1:
            resultObj.innerHTML = "Pay by Visa Card";
            break;
        case 2:
            resultObj.innerHTML = "Pay by Master Card";
            break;
        case 3:
            resultObj.innerHTML = "Pay by Paypal";
```

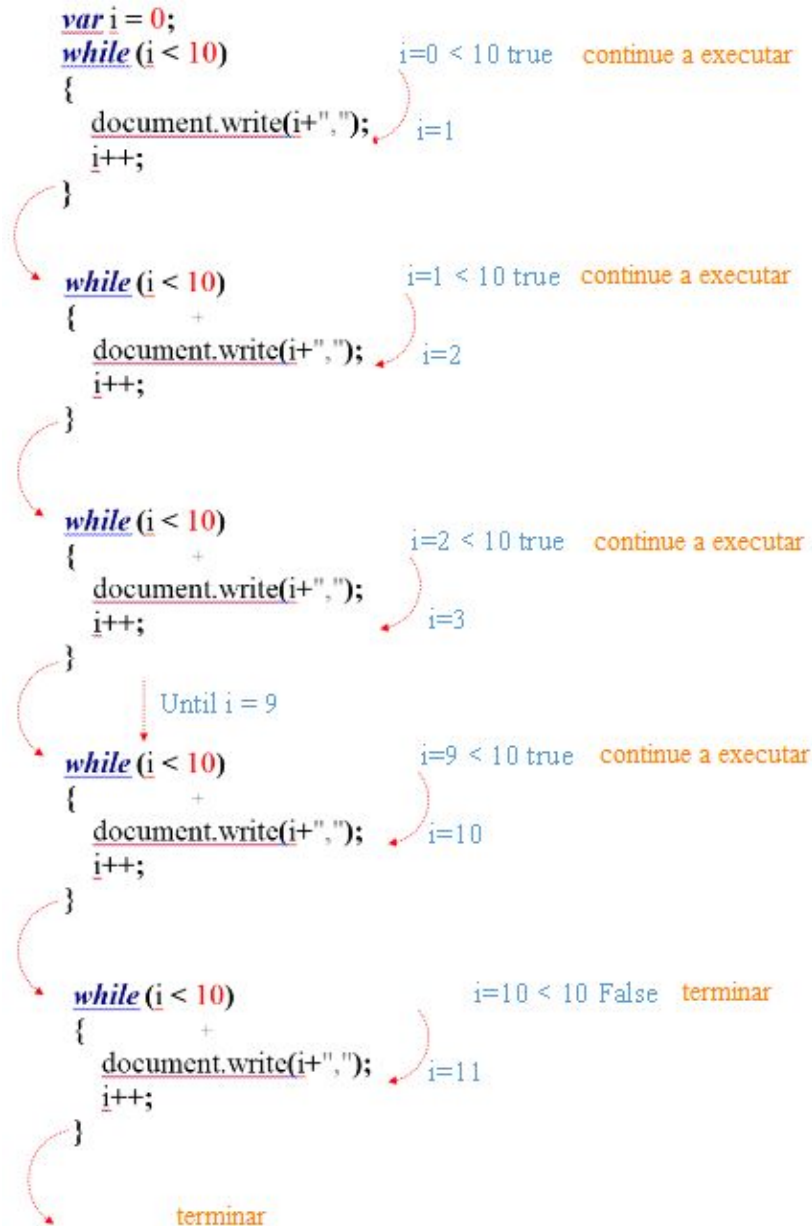
```
        break;  
    default:  
        resultObj.innerHTML = "Pay by face to face";  
    }  
}  
</script>
```

While Loop

while(expressão){

}

se a expressão continuar execute 1, caso contrário, sai do loop



1. Crie um arquivo: **WhileLoop.html**

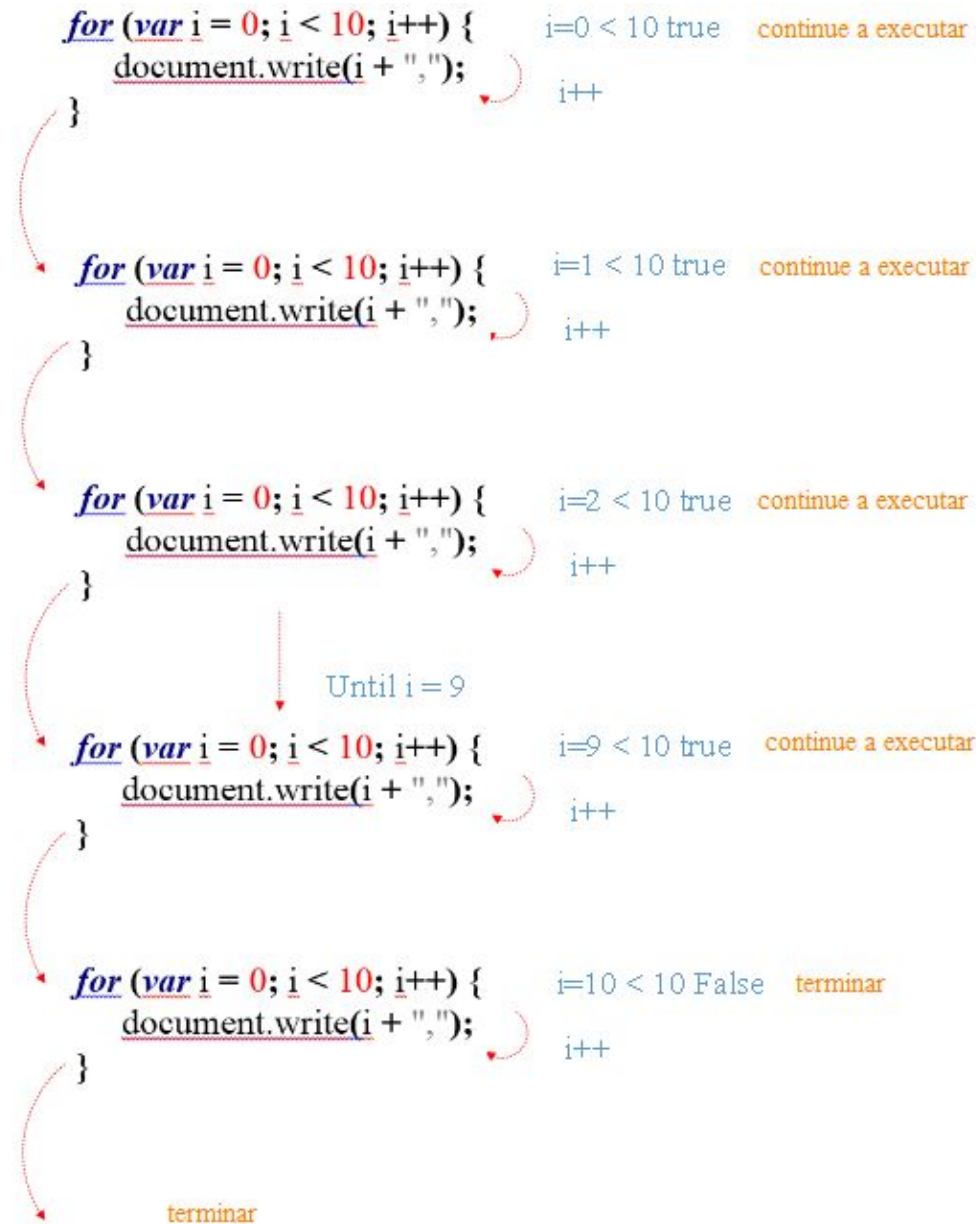
```
<script type="text/javascript">  
  
  var i = 0;  
  while (i < 10)  
  {  
    document.write(i+",");  
    i++;  
  }  
  
</script>
```

Resultado:

0,1,2,3,4,5,6,7,8,9,

For Loop

for (variável de inicialização; condição; variável de atualização){
}



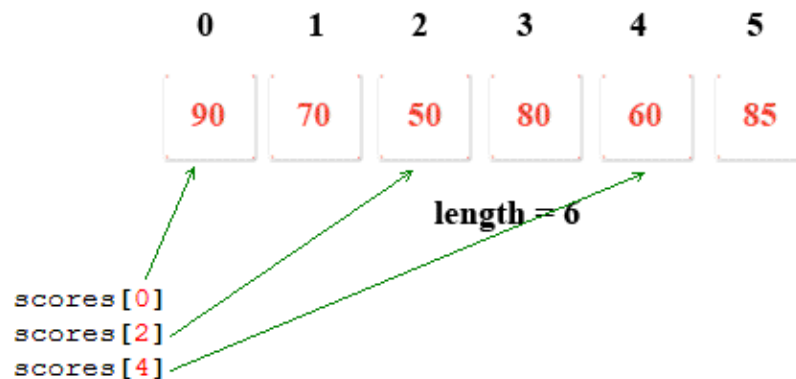
1. Crie um arquivo: **ForLoop.html**

```
<script type="text/javascript">  
  
  for (var i = 0; i < 10; i++) {  
    document.write(i + ",");  
  }  
  
</script>
```

Resultado:

0,1,2,3,4,5,6,7,8,9,

Matriz Unidimensional (One-Dimensional Array)



1. Crie um arquivo: **OneArray.html**

```
<script type="text/javascript">  
  
  // Definição e inicialização de array unidimensional  
  var scores = [ 90, 70, 50, 80, 60, 85 ];  
  
  document.write(scores[0] + "<br>");  
  document.write(scores[2] + "<br>");  
  document.write(scores[4] + "<br>");  
  
</script>
```

Resultado:

90
70
50
60

2. imprimir todas as pontuações do array

Scores.length: a contagem total da matriz

```
<script type="text/javascript">

    var scores = [ 90, 70, 50, 80, 60, 85 ];

    // imprime todas as pontuações do array
    for (var i = 0; i < scores.length; i++) {
        document.write(scores[i] + ",");
    }

</script>
```

Resultado:

90,70,50,80,60,85,

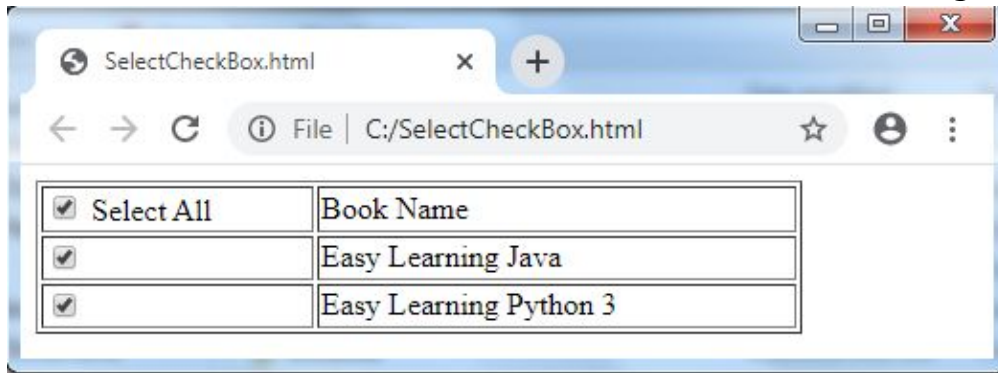
3. imprimir todas as pontuações do array

```
<script type="text/javascript">  
  
  // Defina dinamicamente uma matriz unidimensional  
  var scores = new Array();  
  scores[0] = 90;  
  scores[1] = 70;  
  scores[2] = 50;  
  scores[3] = 80;  
  scores[4] = 60;  
  scores[5] = 85;  
  
  // imprime todas as pontuações do array  
  for (var i = 0; i < scores.length; i++) {  
    document.write(scores[i] + ",");  
  }  
  
</script>
```

Resultado:

90,70,50,80,60,85,

Selecionar Tudo Caixa de Seleção



1. Crie um arquivo: **SelectCheckBox.html**

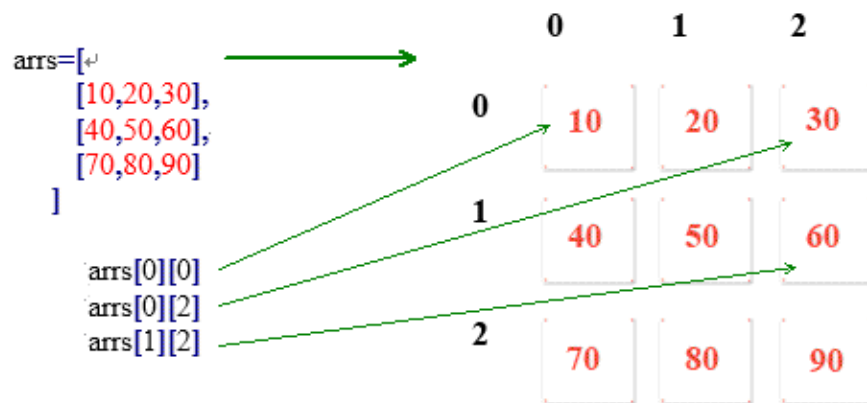
document.getElementsByName(): obter uma matriz de elementos com o mesmo nome

checked: se a caixa de seleção estiver marcada

```
<table width="400" border="1">
  <tr>
    <td><input type="checkbox" onclick="checkAll(this)" /> Select
    All</td>
    <td>Book Name</td>
  </tr>
  <tr>
    <td><input type="checkbox" name="book" /></td>
    <td>Easy Learning Java</td>
  </tr>
  <tr>
    <td><input type="checkbox" name="book" /></td>
    <td>Easy Learning Python 3</td>
  </tr>
</table>
<script language="JavaScript">
  function checkAll(obj)
  {
    var bookArray=document.getElementsByName("book");
    for(var i=0;i<bookArray.length;i++)
```

```
    {  
      bookArray[i].checked=obj.checked;  
    }  
  }  
</script>
```

Matriz Bidimensional



1. Crie um arquivo: **TwoArray.html**

```
<script type="text/javascript">

// Definição e inicialização de array bidimensional
var arrs =[
    [ 10, 20, 30 ],
    [ 40, 50, 60 ],
    [ 70, 80, 90 ]
];

document.write(arrs[0][0] + "<br>");
document.write(arrs[0][2] + "<br>");
document.write(arrs[1][2] + "<br>");

</script>
```

Resultado:

10
30
60

2. Crie um arquivo: **TwoArray2.html**

i: row index, j: column index

arrs[i][j]

	i	0	1	2
j	0	10	20	30
	1	40	50	60
	2	70	80	90

```
<script type="text/javascript">
```

```
var arrs =[
  [ 10, 20, 30 ],
  [ 40, 50, 60 ],
  [ 70, 80, 90 ]
];

// i: índice da linha, j: índice da coluna
for (var i = 0; i < arrs.length; i++) {
  for (var j = 0; j < arrs[i].length; j++) {
    document.write(arrs[i][j]);
    document.write(" ");
  }
  document.write("<br>");
}
```

```
</script>
```

Jogo Find Dog

1. Crie botões de acordo com mapas de matriz bidimensional

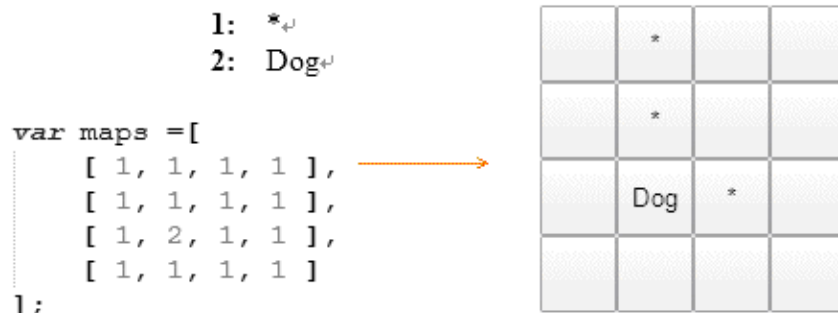
```
var maps =[  
  [ 1, 1, 1, 1 ],  
  [ 1, 1, 1, 1 ],  
  [ 1, 2, 1, 1 ],  
  [ 1, 1, 1, 1 ]  
];
```



```
<style>  
  input{  
    width:40px;  
    height:40px;  
  }  
</style>  
  
<span id="result"></span>  
  
<script type="text/javascript">  
  var maps =[  
    [ 1, 1, 1, 1 ],  
    [ 1, 1, 1, 1 ],  
    [ 1, 2, 1, 1 ],  
    [ 1, 1, 1, 1 ]  
  ];  
  var mapHTML = "";  
  for (var i = 0; i < maps.length; i++) {  
    for (var j = 0; j < maps[i].length; j++) {  
      mapHTML += "<input type='button' value=' ' />";  
    }  
    mapHTML += "<br>";  
  }  
</script>
```

```
}  
document.getElementById("result").innerHTML = mapHTML;  
</script>
```


2. Preencha o nome da matriz no botão e adicione um evento de clique de botão, se o nome do botão for **1** torna-se *****, se o nome do botão for **2** torna-se **Dog**



1: *

2: Dog

```

<style>
  input{
    width:40px;
    height:40px;
  }
</style>

<span id="result"></span>

<script type="text/javascript">
  var maps =[
    [ 1, 1, 1, 1 ],
    [ 1, 1, 1, 1 ],
    [ 1, 2, 1, 1 ],
    [ 1, 1, 1, 1 ]
  ];

  var mapHTML = "";
  for (var i = 0; i < maps.length; i++) {
    for (var j = 0; j < maps[i].length; j++) {
      mapHTML += "<input type='button' value=' '
onClick='doButtonClick(this,"+maps[i][j]+"' />";

```

```

    }
    mapHTML += "<br>";
}
document.getElementById("result").innerHTML = mapHTML;

function doButtonClick(obj,value){
    if(value == 1){
        obj.value = "*";
    }else if(value == 2){
        obj.value = "Dog";
    }
}
</script>

```

3. Imprimir números aleatórios 0-9

Math.random(): retorna um número aleatório entre 0 e 1.

Math.floor(): arredonda um número para baixo para o inteiro mais próximo

```

<script type="text/javascript">
    for(var i=0;i<10;i++){
        var num = Math.floor(Math.random() * 10)
        document.write(num + ", ");
    }
</script>

```

Resultado:

9, 3, 9, 0, 4, 1, 0, 3, 4, 5,

4. Gere aleatoriamente 1- 2, inicialize a matriz bidimensional, preencha o nome da matriz no botão e adicione um evento de clique de botão, se o nome do botão for **1** torna-se *****, se o nome do botão for **2** torna-se **Dog**

	*	*	
	Dog	Dog	
*	Dog		
Dog			

1: *

2: Dog

```
<style>
  input{
    width:40px;
    height:40px;
  }
</style>

<span id="result"></span>

<script type="text/javascript">
  var maps =[
    [ 0, 0, 0, 0 ],
    [ 0, 0, 0, 0 ],
    [ 0, 0, 0, 0 ],
    [ 0, 0, 0, 0 ]
  ];

  for (var i = 0; i < maps.length; i++) {
    for (var j = 0; j < maps[i].length; j++) {
      var num = Math.floor(Math.random()*2)+1
      maps[i][j] = num;
    }
  }
</script>
```

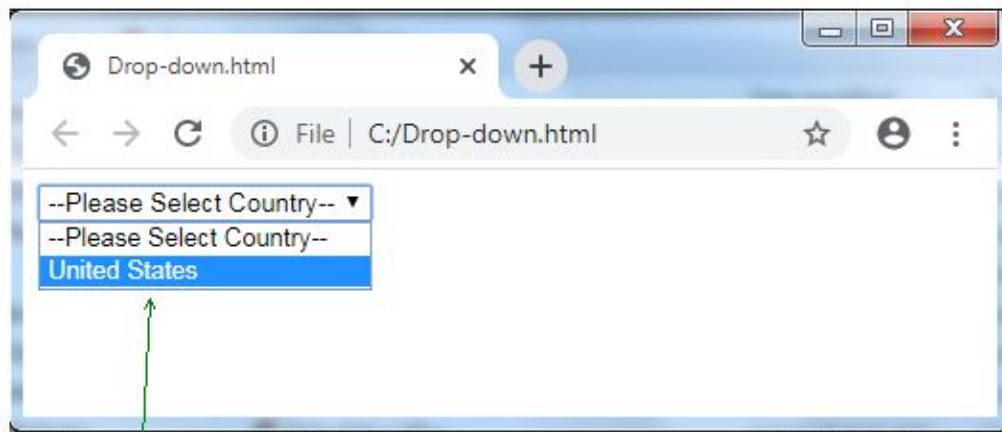
```
var mapHTML = "";
for (var i = 0; i < maps.length; i++) {
    for (var j = 0; j < maps[i].length; j++) {
        mapHTML += "<input type='button' value=' '
onClick='doButtonClick(this,\"+maps[i][j]+")' />";
    }
    mapHTML += "<br>";
}
document.getElementById("result").innerHTML = mapHTML;

function doButtonClick(obj,value){
    if(value == 1){
        obj.value = "*";
    }else if(value == 2){
        obj.value = "Dog";
    }
}
</script>
```

Lista Suspensa de Ligação Secundária

1. Crie um arquivo: **Drop-down.html**

window.onload: executa uma função depois que uma página foi carregada
document.createElement(): crie um elemento html



```
<select id="country">
  <option value="">--Please Select Country--</option>
  <option value="United States">United States</option>
</select>
```

```
var countryObj=document.getElementById("country");

option=document.createElement("option");
option.value="United States";
option.text="United States";

countryObj.options[countryObj.options.length]=option;
```

```
<select id="country">
  <option value="">--Please Select Country--</option>
</select>

<script type="text/javascript">

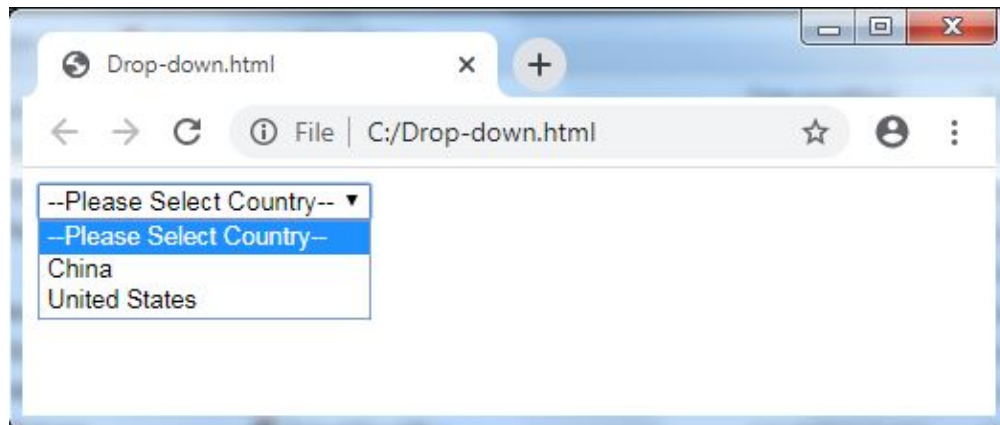
  window.onload=function()
  {
    var countryObj=document.getElementById("country");

    option=document.createElement("option");
    option.text="United States";
    option.value="United States";

    countryObj.options[countryObj.options.length]=option;
  }

</script>
```

2. Adicionar mais **options** por **countryArray**



```
<select id="country">
  <option value="">--Please Select Country--</option>
</select>

<script type="text/javascript">

  var countryArray=["China", "United States"];

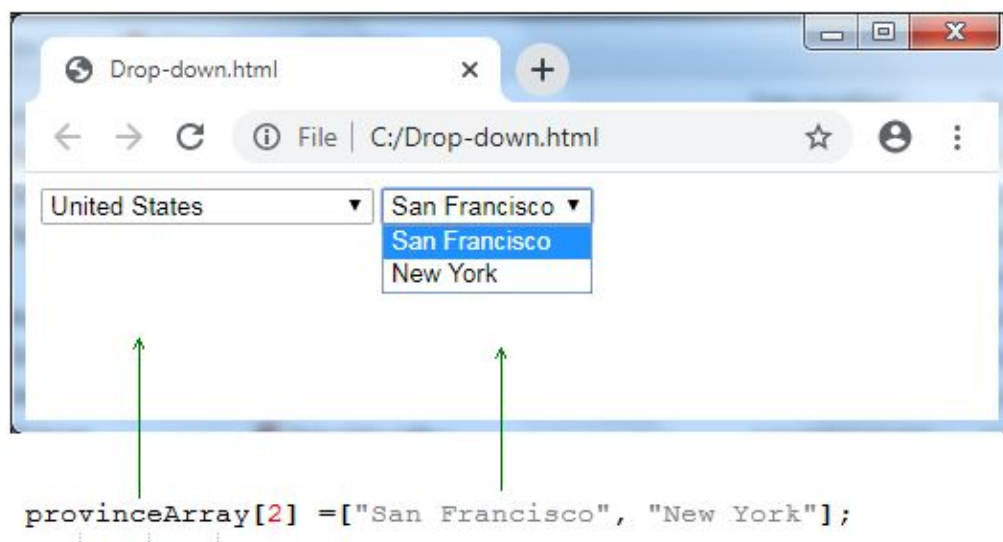
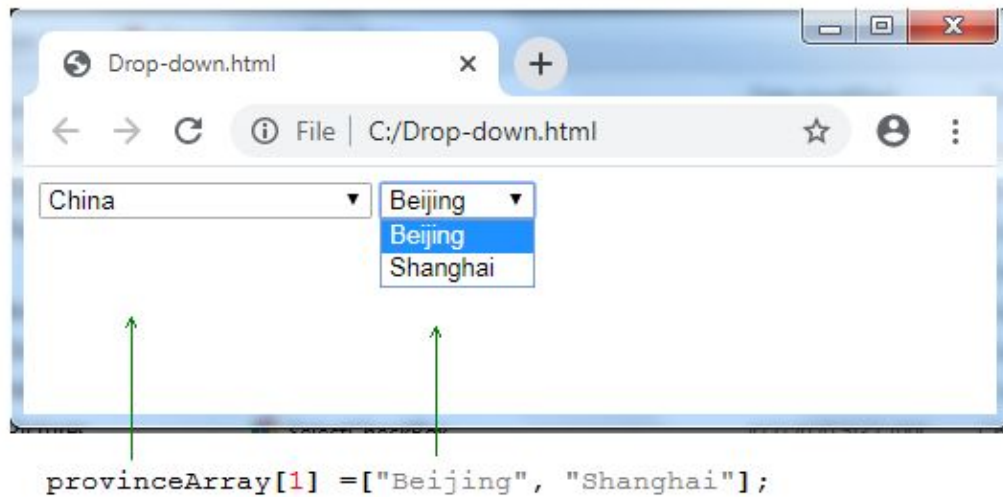
  window.onload=function()
  {
    var countryObj=document.getElementById("country");

    for(var i=0;i<countryArray.length;i++)
    {
      var option=document.createElement("option");
      option.text=countryArray[i];
      option.value=countryArray[i];
      countryObj.options[countryObj.options.length]=option;
    }
  }

</script>
```

3. Adicionar **options** de **province**

selectedIndex: define ou retorna o índice da opção selecionada em uma lista suspensa.




```
<select id="country" onchange="doCountryChange(this)" >
  <option value="">--Please Select Country--</option>
</select>
```

```
<select id="province" >
</select>
```

```
<script type="text/javascript">
  var countryArray=["China", "United States"];

  var provinceArray = new Array();
  provinceArray[0] =[];
  provinceArray[1] =["Beijing", "Shanghai"];
  provinceArray[2] =["San Francisco", "New York"];
```

```
window.onload=function()
{
  var country=document.getElementById("country");
  for(var i=0;i<countryArray.length;i++)
  {
    var option=document.createElement("option");
    option.text=countryArray[i];
    option.value=countryArray[i];
    country.options[country.options.length]=option;
  }
}
```

```
function doCountryChange(obj)
{
  var provinceObj=document.getElementById("province");
  provinceObj.options.length=0; // clear all
  var index=obj.selectedIndex;
  for(var i=0;i<provinceArray[index].length;i++)
  {
    var option=document.createElement("option");
```

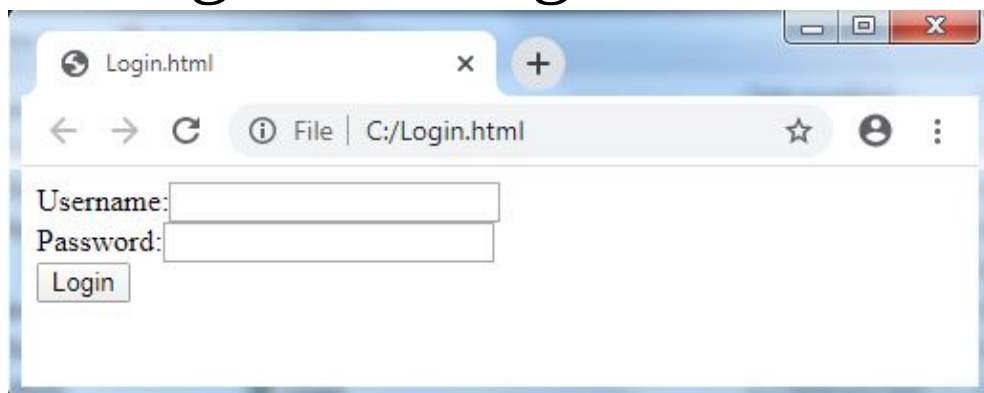
```
    option.text=provinceArray[index][i];  
    option.value=provinceArray[index][i];  
    provinceObj.options[provinceObj.options.length]=option;  
  }  
}  
</script>
```

Evento (**Event**)

Lista de evento:

Onblur	elemento perde o foco
OnChange	muda o conteúdo
OnClick	clique do mouse em um objeto
Ondblclick	clique duas vezes com o mouse em um objeto
Onerror	Ocorreu um erro ao carregar um documento ou imagem
Onfocus	elemento obtém foco
Onkeydown	A tecla de um teclado é pressionada
Onkeypress	A tecla de um teclado é pressionada
Onkeyup	A tecla de um teclado é liberada
Onload	uma página ou imagem é carregada
Onmousedown	um botão do mouse é pressionado
Onmousemove	mouse é movido
Onmouseout	o mouse é movido para fora de um elemento
Onmouseover	o mouse é movido sobre um elemento
Onmouseup	botão do mouse é liberado
Onreset	botão de reset foi clicado
Onresize	janela ou moldura é redimensionada
Onselect	o texto é selecionado
Onsubmit	botão enviar foi clicado
Onunload	página de saída

Página de Login da Web



1. Crie um arquivo: **Login.html**

```
Username:<input type="text" id="username" value="" /> <br>
Password:<input type="password" id="pwd" value="" /> <br>
<input type="button" value="Login" onclick="checkLogin()" />
```

```
<script type="text/javascript">
```

```
    function checkLogin()
```

```
    {
```

```
        var usernameObj=document.getElementById("username");
```

```
        var pwdObj=document.getElementById("pwd");
```

```
        if(usernameObj.value=="")
```

```
        {
```

```
            alert("Please input username !");
```

```
            return;
```

```
        }
```

```
        if(pwdObj.value=="")
```

```
        {
```

```
            alert("Please input password !");
```

```
            return;
```

```
        }
```

```
        alert("login successfull !");
```

```
}  
</script>
```

2. Crie um arquivo: **Login.html**

document.onkeypress: o evento ocorre quando o usuário pressiona uma tecla.

```
Username:<input type="text" id="username" value="" />
<br>
Password:<input type="password" id="pwd" value="" />
<br>
<input type="button" value="Login" onclick="checkLogin()" />

<script type="text/javascript">
    document.onkeypress=function(event)
    {
        var ext=window.event?window.event:event;
        var key=ext.keyCode?ext.keyCode:ext.which;
        if(key==13) // pressione a tecla Enter será invocado
        {
            checkLogin();
        }
    }

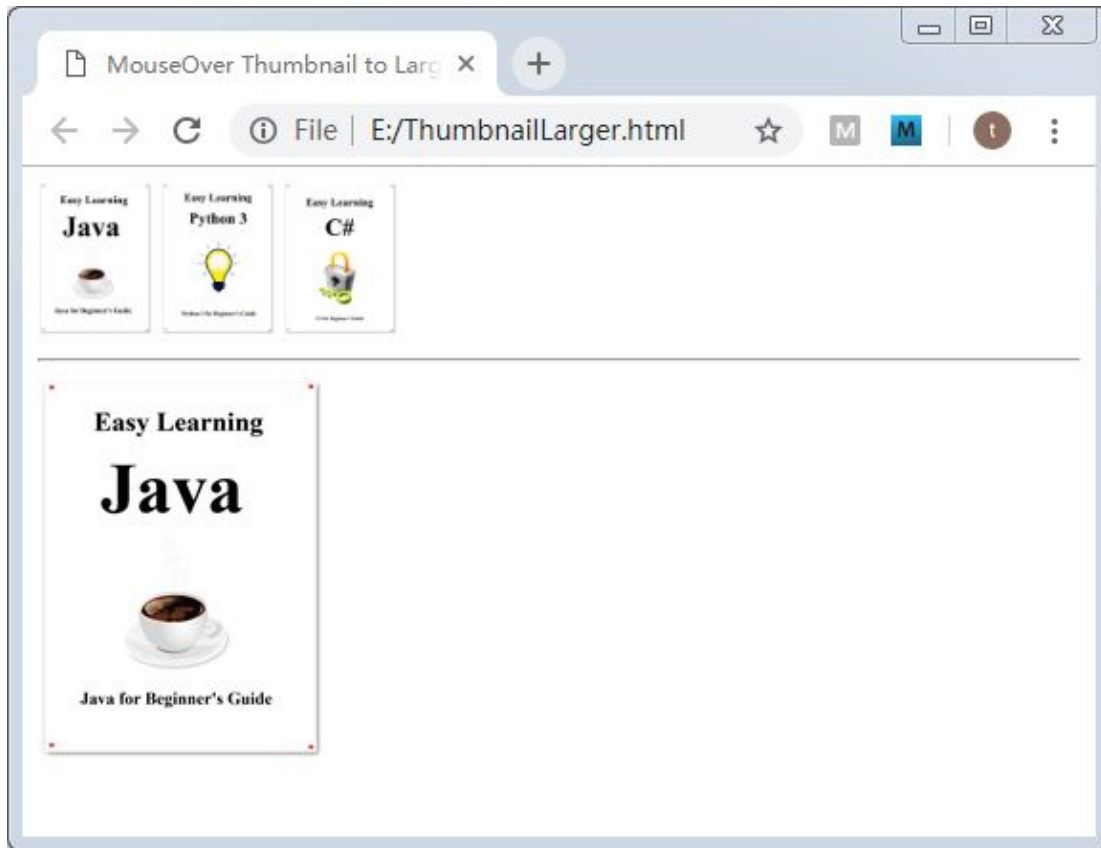
    function checkLogin()
    {
        var usernameObj=document.getElementById("username");
        var pwdObj=document.getElementById("pwd");

        if(usernameObj.value=="")
        {
            alert("Please input username !");
            return;
        }

        if(pwdObj.value=="")
        {
            alert("Please input password !");
        }
    }
}
```

```
        return;  
    }  
  
    alert("login successfull !");  
}  
</script>
```

MouseOver Miniatura Para Maior



1. Crie um arquivo: **ThumbnailLarger.html**

****: tag define uma imagem em uma página HTML.

```



<hr>
<img id="image" src="" />
<script type="text/javascript">
    function zoomImage(obj)
```

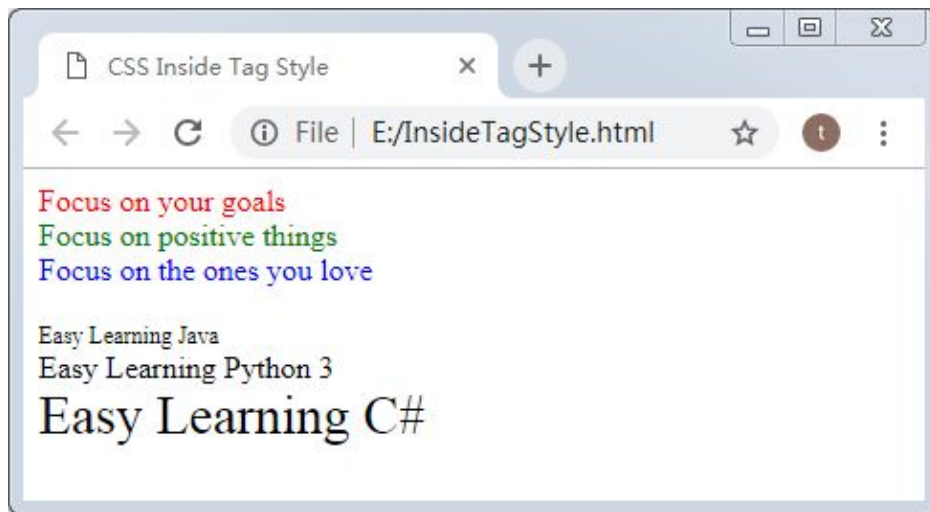


```
{  
  document.getElementById("image").src = obj.src;  
}  
</script>
```

CSS Dentro do Estilo da Tag

CSS é uma linguagem que descreve o estilo de um documento HTML.

1. Crie um arquivo: **InsideTagStyle.html**



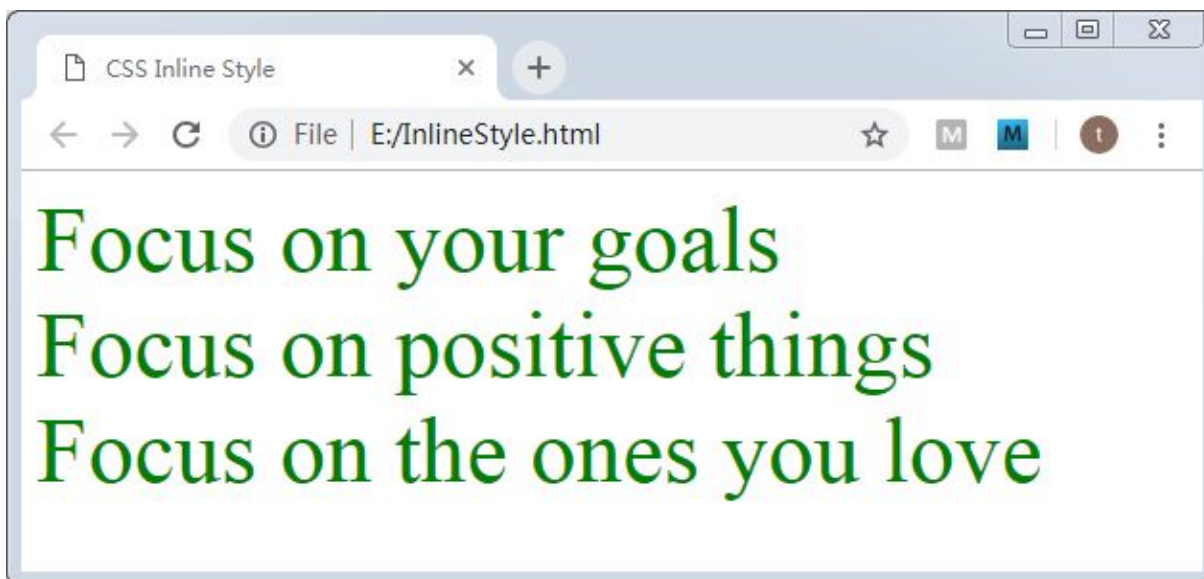
```
<div style="color:red">Focus on your goals</div>
<div style="color:green">Focus on positive things</div>
<div style="color:blue">Focus on the ones you love</div>

<br>

<div style="font-size:12px">Easy Learning Java</div>
<div style="font-size:16px">Easy Learning Python 3</div>
<div style="font-size:28px">Easy Learning C#</div>
```

Estilo CSS Inline

1. Crie um arquivo: **InsideStyle.html**

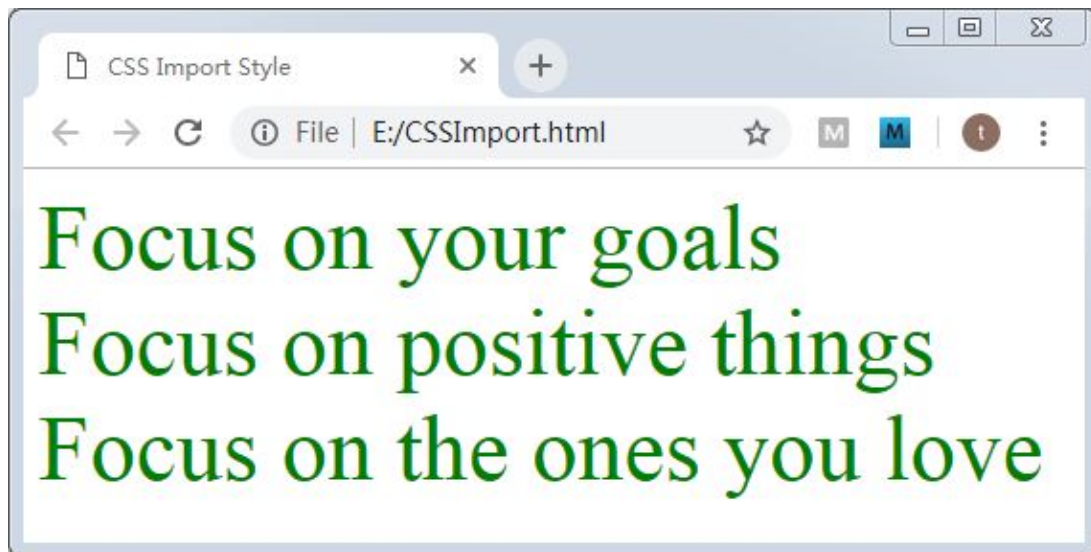


The .class selector: seleciona elementos com um atributo de classe específico.

```
<style>
  .fontStyle{
    color:green;
    font-size:48px;
  }
</style>

<div class="fontStyle">Focus on your goals</div>
<div class="fontStyle">Focus on positive things</div>
<div class="fontStyle">Focus on the ones you love</div>
```

Estilo de Importação CSS



1. Crie um arquivo: **style.css**

```
@CHARSET "UTF-8";  
  
div{  
    color:green;  
    font-size:48px;  
}
```

2. Crie um arquivo: **CSSImport.html**

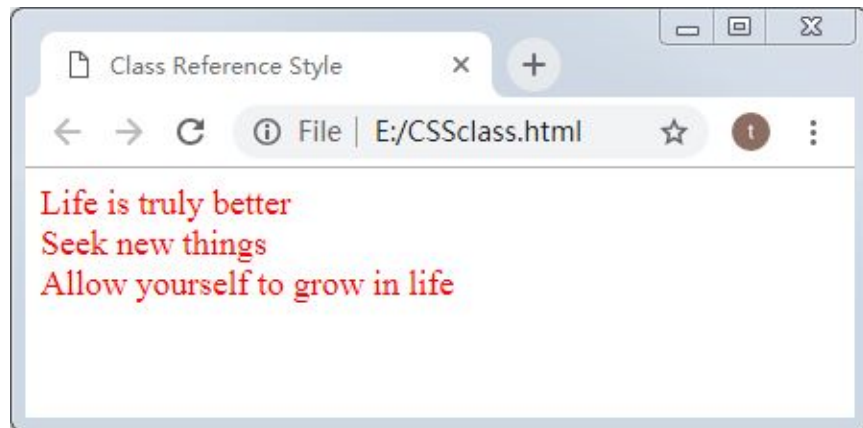
<link>: tag define um link entre um documento e um recurso externo **style.css**.

```
<link rel="stylesheet" type="text/css" href="style.css" />  
  
<div >Focus on your goals</div>  
<div >Focus on positive things</div>
```

`<div >`**Focus on the ones you love**`</div>`

Estilo de Referência de Classe

1. Crie um arquivo: **CSSclass.html**



```
<style>
  .classStyleName{
    color:red;
    font-size:18px;
  }
</style>

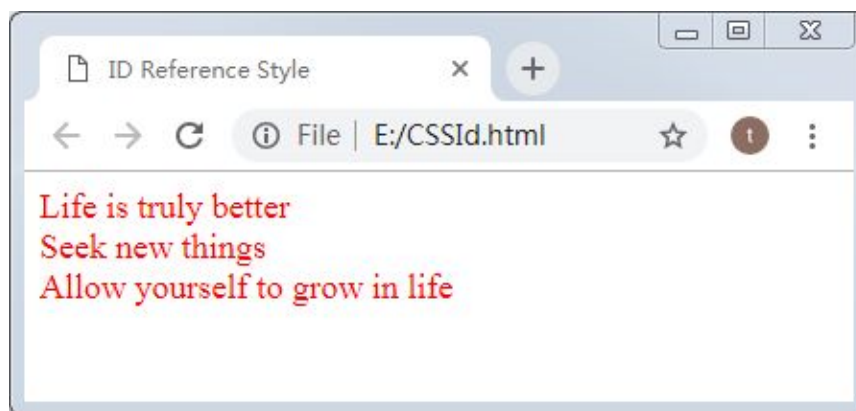
<span class="classStyleName" >Life is truly better</span>
<br>

<span class="classStyleName" >Seek new things</span>
<br>

<span class="classStyleName" >Allow yourself to grow in life</span>
```

Estilo de Referência de ID

1. Crie um arquivo: **CSSId.html**



The #id selector: estiliza o elemento com o id especificado.

```
<style>
  #IdStyleName{
    color:red;
    font-size:18px;
  }
</style>

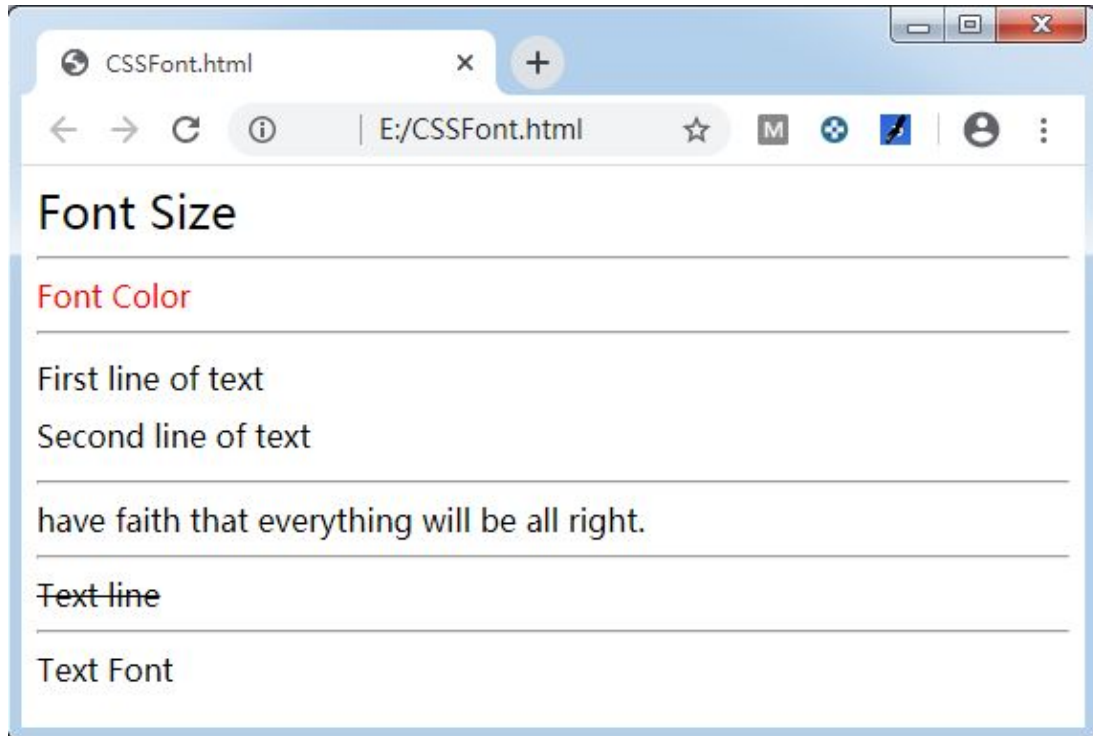
<span id="IdStyleName" >Life is truly better</span>
<br>

<span id="IdStyleName" >Seek new things</span>
<br>

<span id="IdStyleName" >Allow yourself to grow in life</span>
```

Estilo de Fonte CSS

1. Crie um arquivo: **CSSFont.html**



font-size:	especifica o tamanho do texto dentro
line-height:	especifica a altura de uma linha
font-weight:	1.bold 2.normal
text-transform:	1.capitalize 2.uppercase 3.lowercase
text-decoration:	1.underline 2.overline 3.none
font-family:	"Times New Roman"


```
<style>
#fontSize{
  font-size:24px;
}
#fontColor{
  color:#ff0000;
}
#lineHeight{
  line-height: 30px;
}
#textTransform{
  text-transform: none;
}
#textDecoration{
  text-decoration: line-through;
}
#fontFamily{
  font-family: sans-serif;
}
</style>
```

```
<span id="fontSize">Font Size</span>
```

```
<hr>
```

```
<span id="fontColor">Font Color</span>
```

```
<hr>
```

```
<span id="lineHeight">
```

```
  First line of text
```

```
  <br>
```

```
  Second line of text
```

```
</span>
```

```
<hr>
```

```
<span id="textTransform">have faith that everything will be all  
right.</span>
```

```
<hr>
```

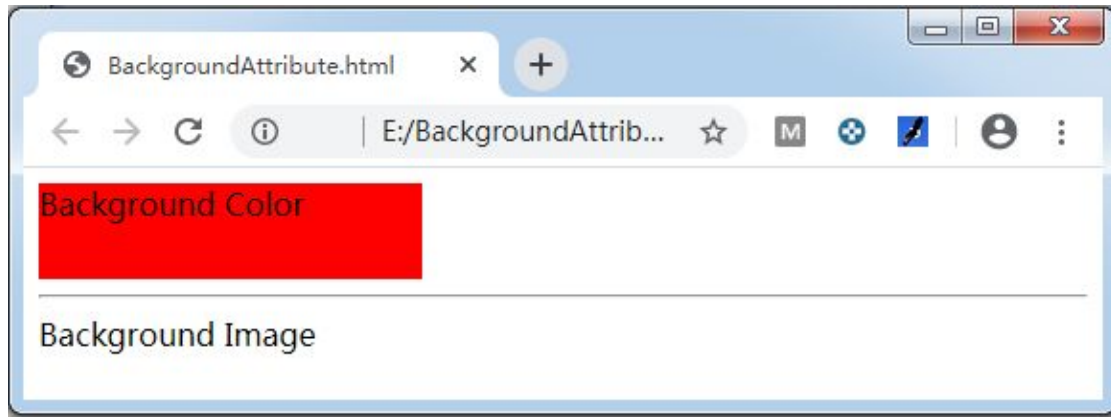
```
<span id="textDecoration">Text line</span>
```

```
<hr>
```

```
<span id="fontFamily">Text Font</span>
```

CSS Estilo de Fundo

1. Crie um arquivo: **BackgroundAttribute.html**



background-color: #FFFFFF;	cor de fundo
background-image: url();	url da imagem
background-repeat: no-repeat;	1.no-repeat 2.no-repeat-x 3.no-repeat-y
background-position:	1.left 2.top

```
<style>
#backgroundColor{
    background-color: #ff0000;
    width:200px;
    height:50px;
}

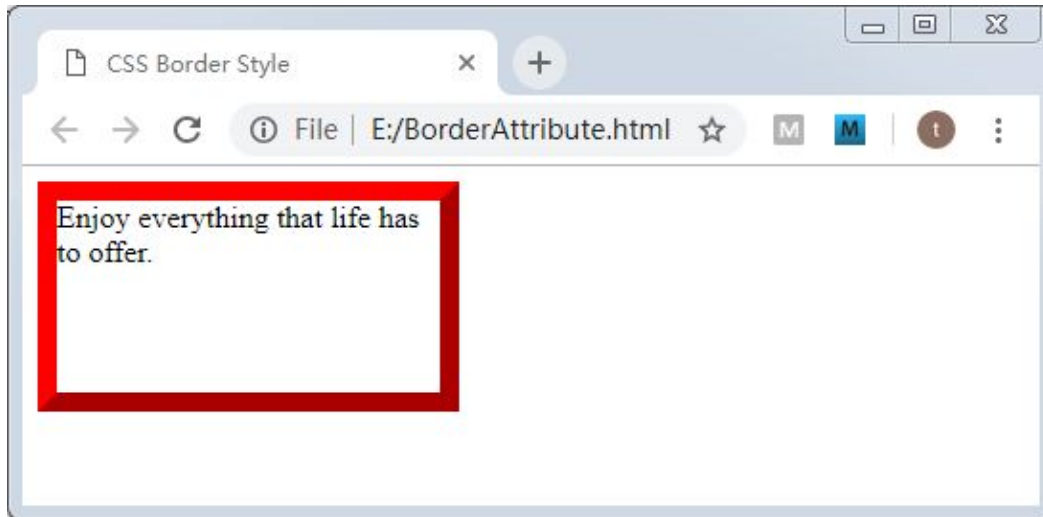
#backgroundImage{
    background-image: url("images/cat.gif");
    background-repeat: no-repeat;
    background-position: right;
}
</style>

<div id="backgroundColor">Background Color</div>
<hr>
<div id="backgroundImage">Background Image</div>
```



CSS Estilo de Borda

1. Crie um arquivo: **BorderAttribute.html**



border-style: 1.dotted 2.dashed 3.solid 4.double
5.groove 6.ridge 7.inset 8.outset

```
<style>
```

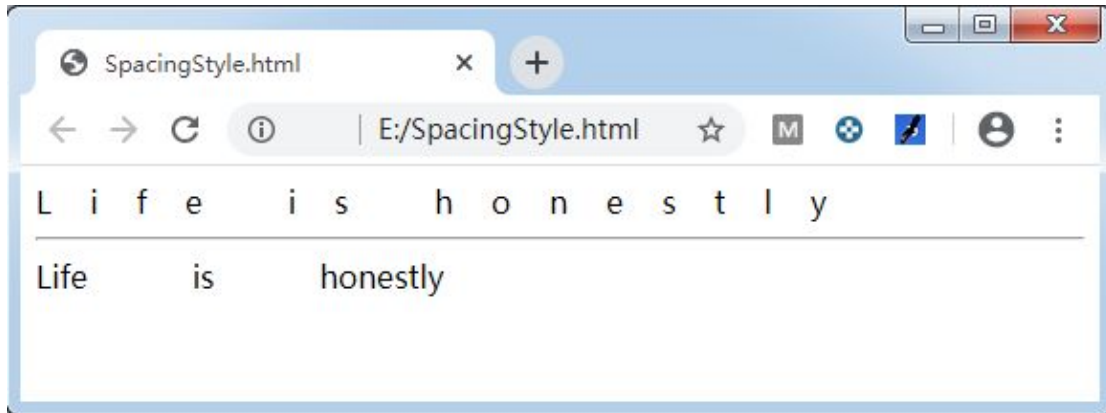
```
#borderStyle{  
  width:200px;  
  height:100px;  
  border-color:#ff0000;  
  border-width:10px;  
  border-style:outset;  
}
```

```
</style>
```

```
<div id="borderStyle">Enjoy everything that life has to offer.</div>
```

Estilo de Espaçamento CSS

1. Crie um arquivo: **SpacingStyle.html**



letter-spacing: aumenta ou diminui o espaço entre os caracteres em um texto.

word-spacing: aumenta ou diminui o espaço em branco entre as palavras.

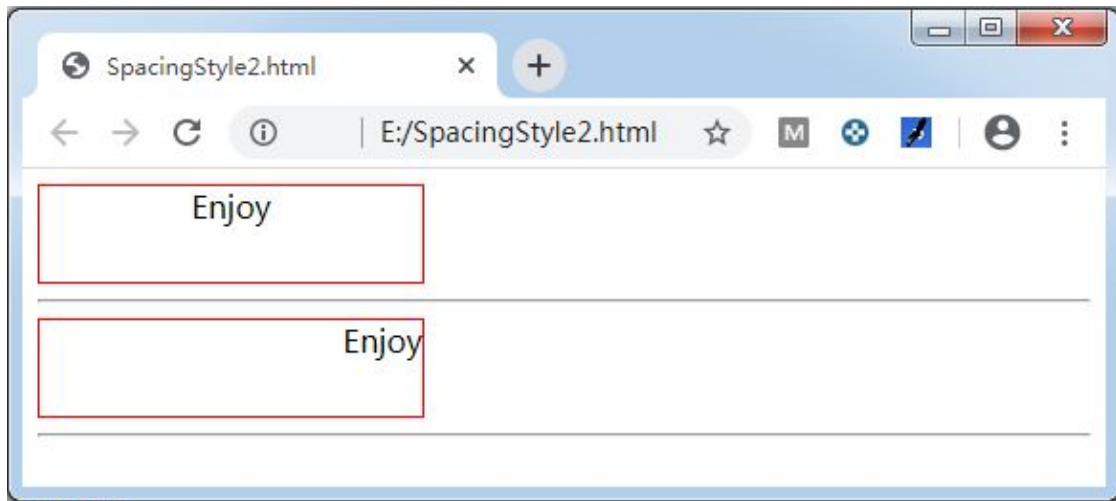
```
<style>
  #letterSpacing{
    letter-spacing: 20px;
  }

  #wordSpacing{
    word-spacing: 50px;
  }
</style>

<div id="letterSpacing">Life is honestly</div>
<hr>

<div id="wordSpacing">Life is honestly</div>
```

2. Crie um arquivo: **SpacingStyle2.html**



text-align: 1.justify 2.left 3.right 4.center

```
<style>
  #textAlignCenter{
    text-align:center;
    width:200px;
    height:50px;
    border: 1px solid #ff0000;
  }

  #textAlignRight{
    text-align:right;
    width:200px;
    height:50px;
    border: 1px solid #ff0000;
  }
</style>

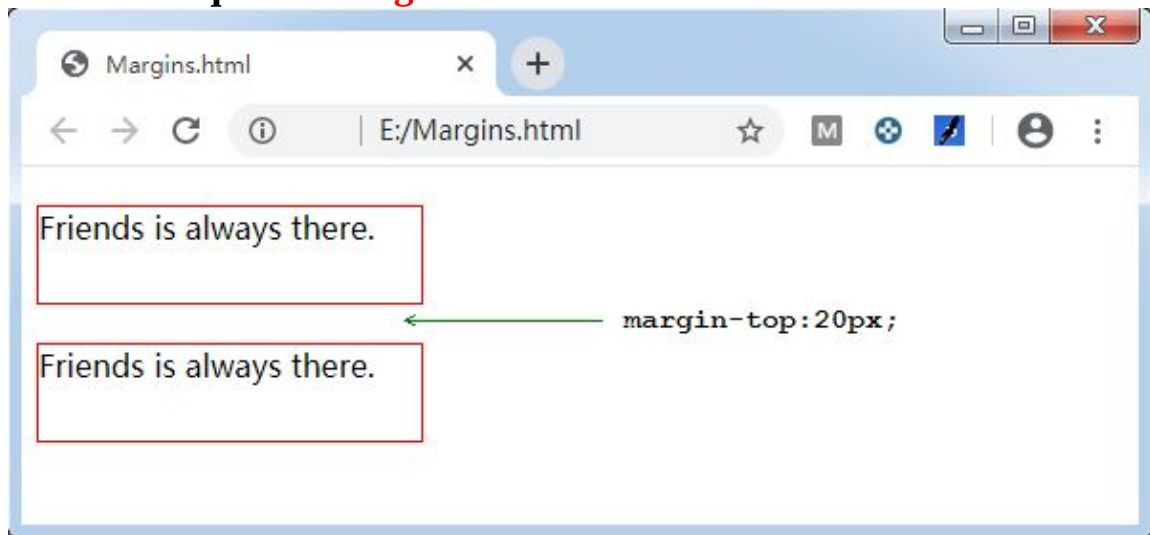
<div id="textAlignCenter">Enjoy</div>
<hr>

<div id="textAlignRight">Enjoy</div>
```

<hr>

Margens Internas e Externas de CSS

1. Crie um arquivo: **Margins.html**



margin: Outer Margins

- 1 margin-top
- 2 margin-bottom
- 3 margin-left
- 4 margin-right

```
<style>
  #marginDiv{
    width:200px;
    height:50px;
    border:1px solid #ff0000;
    margin-top:20px;
  }
</style>

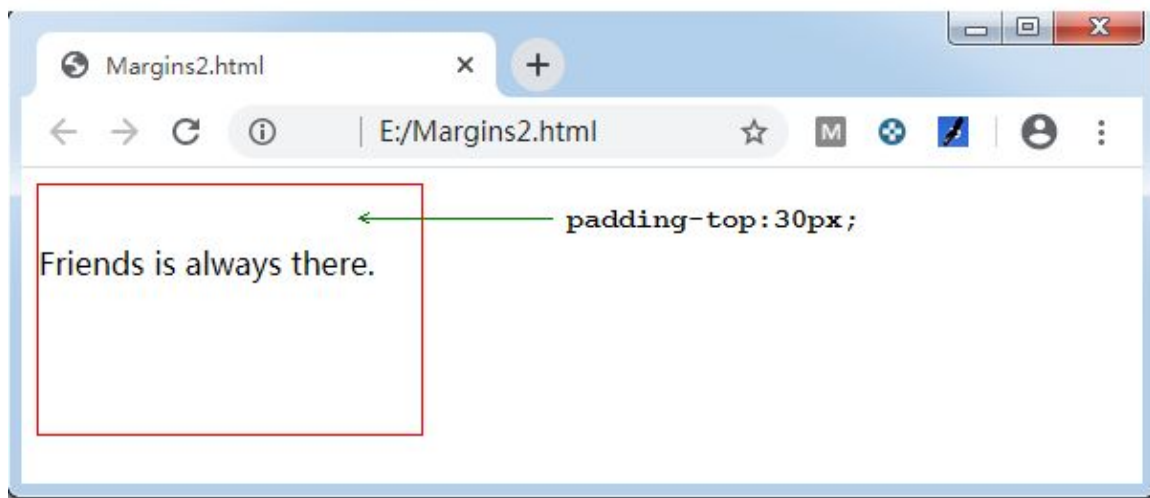
<div id="marginDiv">
  Friends is always there.
</div>

<div id="marginDiv">
```

Friends is always there.

</div>

2. Crie um arquivo: **Margins2.html**



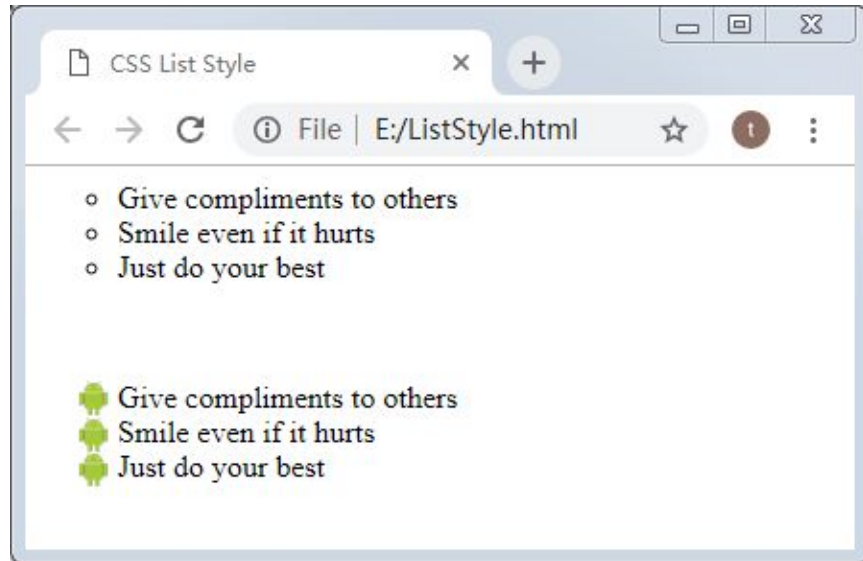
padding: Inner Margins
padding-top
padding-bottom
padding-left
padding-right

```
<style>
  #paddingDiv{
    width:200px;
    height:100px;
    border:1px solid #ff0000;
    padding-top:30px;
  }
</style>

<div id="paddingDiv">
  Friends is always there.
</div>
```

Estilo de Lista CSS

1. Crie um arquivo: **ListStyle.html**



list-style-type: 1.disc 2.circle 3.square 4.decimal 5.lower-roman
6.upper-roman 7.lower-alpha 8.upper-alpha
list-style-position: 1.outside 2.inside
list-style-image: url(..); picture

```
<style>
#list{
  list-style-type: circle;
  list-style-position:outside;
}

#listImage{
  list-style-type: circle;
  list-style-position:outside;
  list-style-image: url(images/icon.jpg);
}
</style>
```

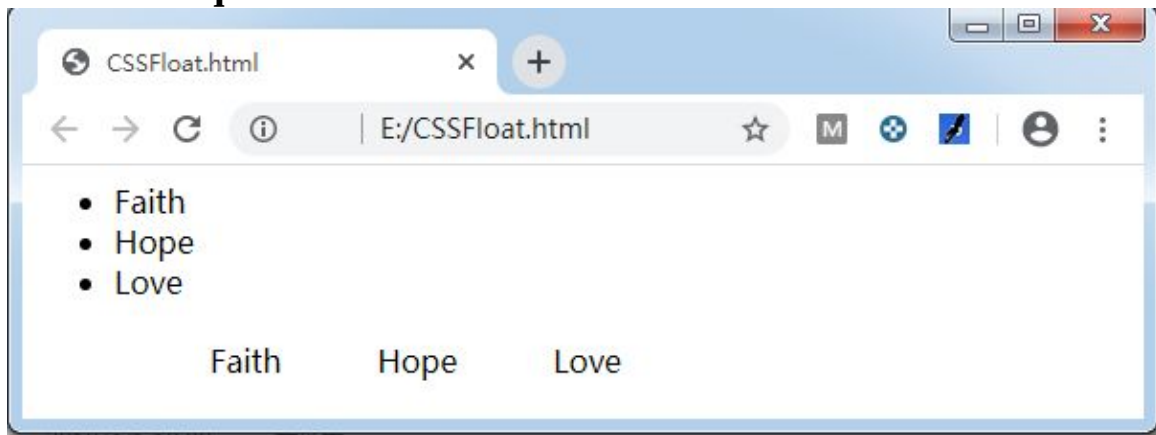
```
<ul id="list">
  <li>Give compliments to others</li>
  <li>Smile even if it hurts</li>
  <li>Just do your best</li>
</ul>
```

```
<br>
```

```
<ul id="listImage">
  <li>Give compliments to others</li>
  <li>Smile even if it hurts</li>
  <li>Just do your best</li>
</ul>
```

CSS Float

1. Crie um arquivo: **CSSFloat.html**



float: especifica como um elemento deve flutuar.

```
<style>
  #floatStyle li{
    float:left;
    padding-left:50px;
    list-style:none;
  }
</style>

<ul>
  <li>Faith</li>
  <li>Hope</li>
  <li>Love</li>
</ul>

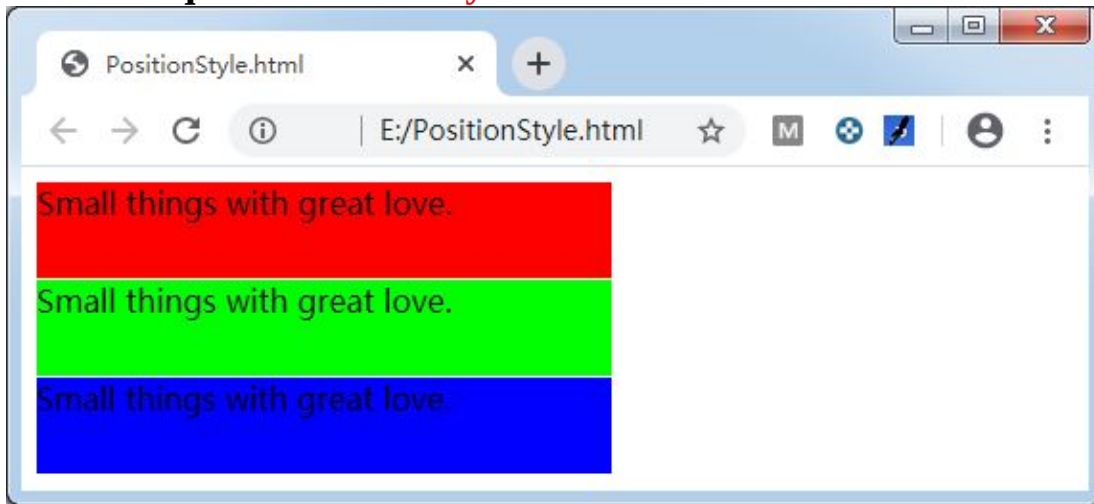
<br>

<ul id="floatStyle">
  <li>Faith</li>
  <li>Hope</li>
  <li>Love</li>
</ul>
```



Estilo de Posição CSS

1. Crie um arquivo: **PositionStyle.html**

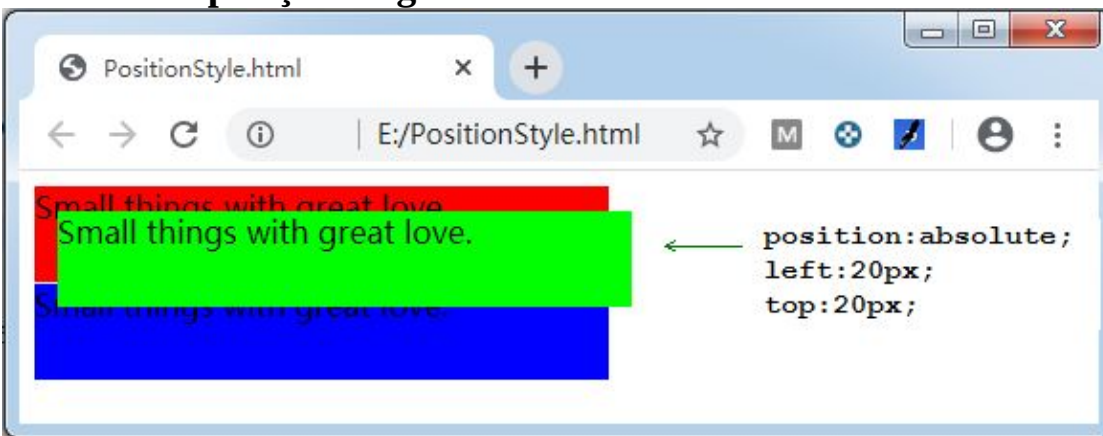


```
<style>
  #redBox{
    width:300px;
    height:50px;
    background-color:#ff0000;
    margin-top:1px;
  }
  #greenBox{
    width:300px;
    height:50px;
    background-color:#00ff00;
    margin-top:1px;
  }
  #blueBox{
    width:300px;
    height:50px;
    background-color:#0000ff;
    margin-top:1px;
  }
</style>
<div id="redBox">Small things with great love.</div>
```



```
<div id="greenBox">Small things with great love.</div>  
<div id="blueBox">Small things with great love.</div>
```

2. Alterar a posição do greenBox: **absolute**



```
<style>
#redBox{
  width:300px;
  height:50px;
  background-color:#ff0000;
  margin-top:1px;
}
#greenBox{
  width:300px;
  height:50px;
  background-color:#00ff00;
  margin-top:1px;
  position:absolute;
  left:20px;
  top:20px;
}
#blueBox{
  width:300px;
  height:50px;
  background-color:#0000ff;
  margin-top:1px;
}

</style>
<div id="redBox">Small things with great love.</div>
```

```
<div id="greenBox">Small things with great love.</div>  
<div id="blueBox">Small things with great love.</div>
```

3. Alterar a posição do greenBox: **relative**



```
<style>
  #redBox{
    width:300px;
    height:50px;
    background-color:#ff0000;
    margin-top:1px;
  }
  #greenBox{
    width:300px;
    height:50px;
    background-color:#00ff00;
    margin-top:1px;
    position:relative;
    left:20px;
    top:20px;
  }
  #blueBox{
    width:300px;
    height:50px;
    background-color:#0000ff;
    margin-top:1px;
  }
</style>
<div id="redBox">Small things with great love.</div>
```

```
<div id="greenBox">Small things with great love.</div>  
<div id="blueBox">Small things with great love.</div>
```

Estilo de Visibilidade CSS

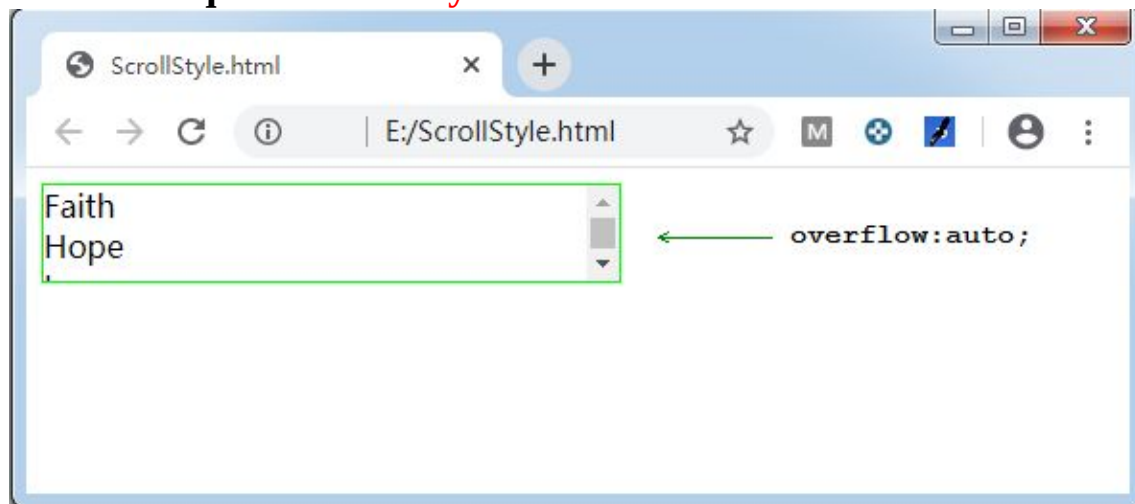
1. Crie um arquivo: **VisibilityStyle.html**



```
<style>
#redBox{
  width:400px;
  height:50px;
  background-color:#ff0000;
  margin-top:1px;
  visibility: visible
}
#greenBox{
  width:400px;
  height:50px;
  background-color:#00ff00;
  margin-top:1px;
  visibility: hidden
}
#blueBox{
  width:400px;
  height:50px;
  background-color:#0000ff;
  margin-top:1px;
  visibility: visible
}
</style>
```

```
<div id="redBox">If you're feeling helpless, help someone.</div>  
<div id="greenBox">If you're feeling helpless, help someone.</div>  
<div id="blueBox">If you're feeling helpless, help someone.</div>
```

2. Crie um arquivo: **ScrollStyle.html**

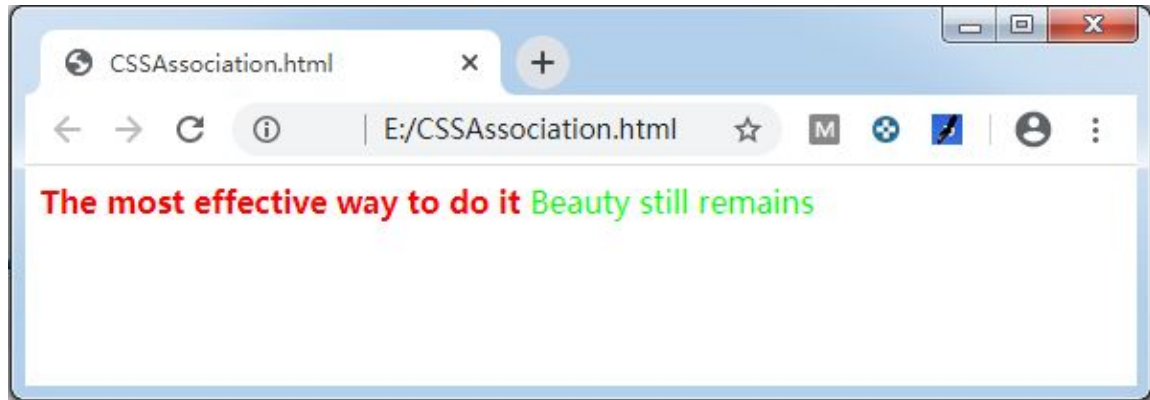


```
<style>
  #greenBox{
    width:300px;
    height:50px;
    border:1px solid #00ff00;
    margin-top:1px;
    overflow:auto;
  }
</style>

<div id="greenBox">
  Faith <br>
  Hope <br>
  Love <br>
  Joy <br>
  Peace
</div>
```


Estilo de Associação CSS

1. Crie um arquivo: **CSSAssociation.html**



```
<style>
div{
  color:#ff0000;
}

div span{
  color:#00ff00;
}
</style>

<div>
  <b>The most effective way to do it</b>
  <span> Beauty still remains</span>
</div>
```

2. Crie um arquivo: **CSSAssociation.html**

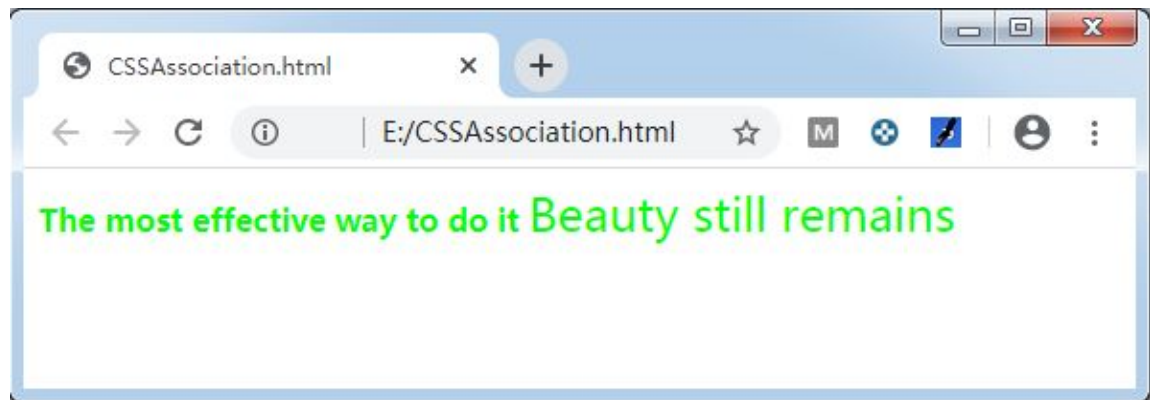


```
<style>
  .div1{
    font-size:24px;
  }

  .div1 span{
    color:#0000ff;
  }
</style>

<div class="div1">
  <b>The most effective way to do it</b>
  <span> Beauty still remains</span>
</div>
```

3. Crie um arquivo: **CSSAssociation.html**



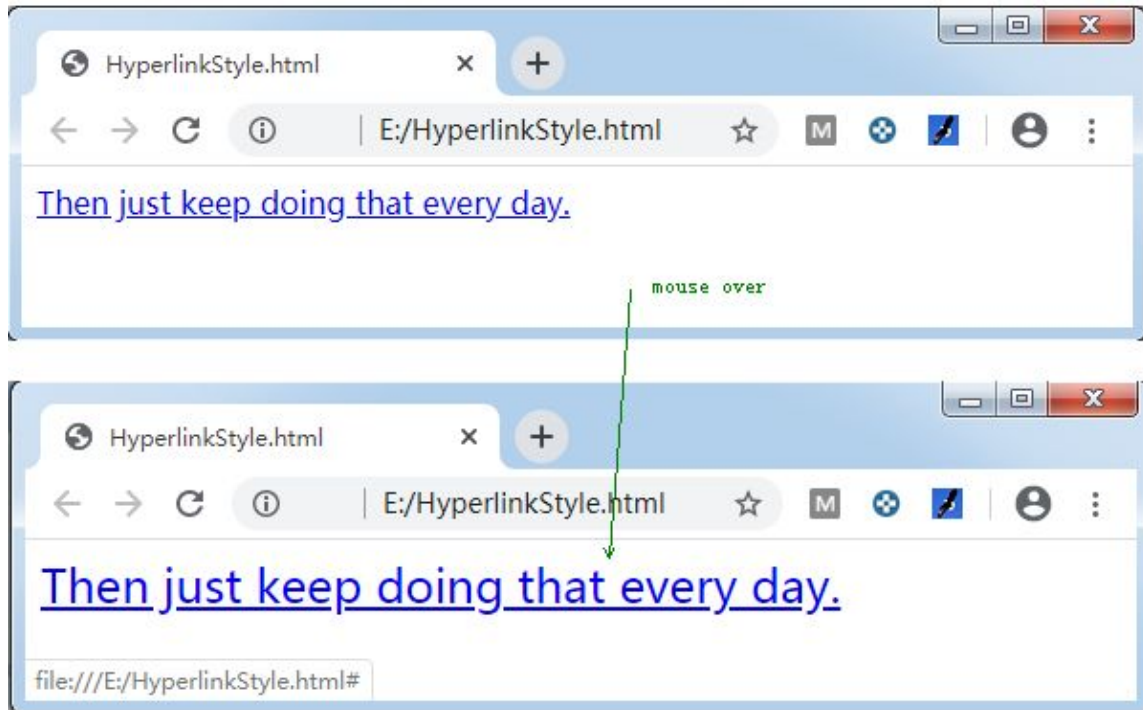
```
<style>
  #div2{
    color:#00ff00;
  }

  #div2 span{
    font-size:24px;
  }
</style>

<div id="div2">
  <b>The most effective way to do it</b>
  <span> Beauty still remains</span>
</div>
```

Estilo de Hiperlink CSS

1. Crie um arquivo: **HyperlinkStyle.html**



a:link	quando o hiperlink não foi clicado
a:visited	depois que o hiperlink é clicado
a:hover	quando o cursor se move para o hiperlink
a:active	quando o hiperlink é clicado

```
<style>
a:link{ }
a:visited{ color:#0000ff; }
a:hover{font-size:24px;}
a:active{color:#00ff00;}

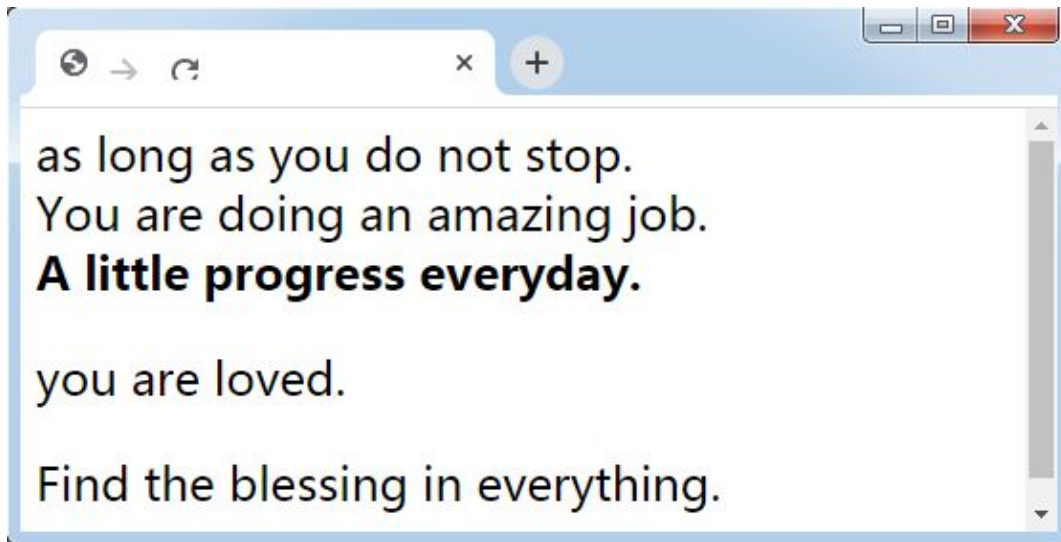
input.focus{background-color: #0000ff;}
</style>

<a href="#">Then just keep doing that every day.</a>
```



CSS Estilo União

1. Crie um arquivo: **CSSUnion.html**



```
<style>
  span,div,className,#idName{
    font-size:24px;
  }
</style>

<span>as long as you do not stop.</span>

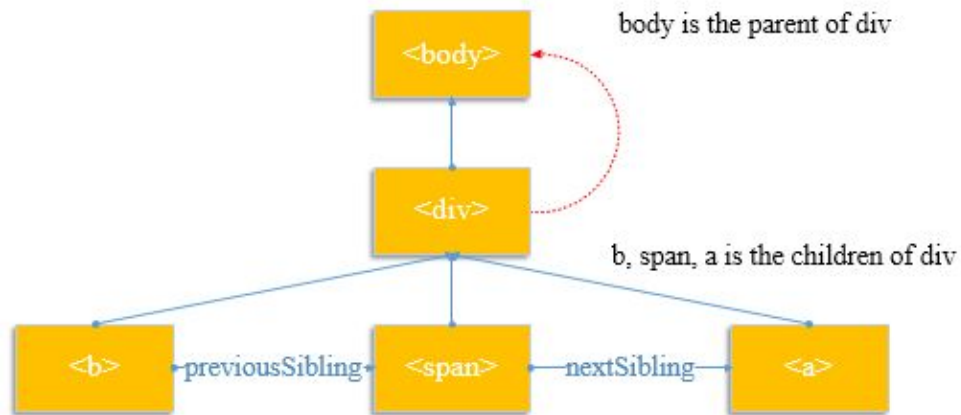
<div>You are doing an amazing job.</div>

<b class="className">A little progress everyday.</b>

<p id="idName">you are loved.</p>

<div><p>Find the blessing in everything.</p></div>
```

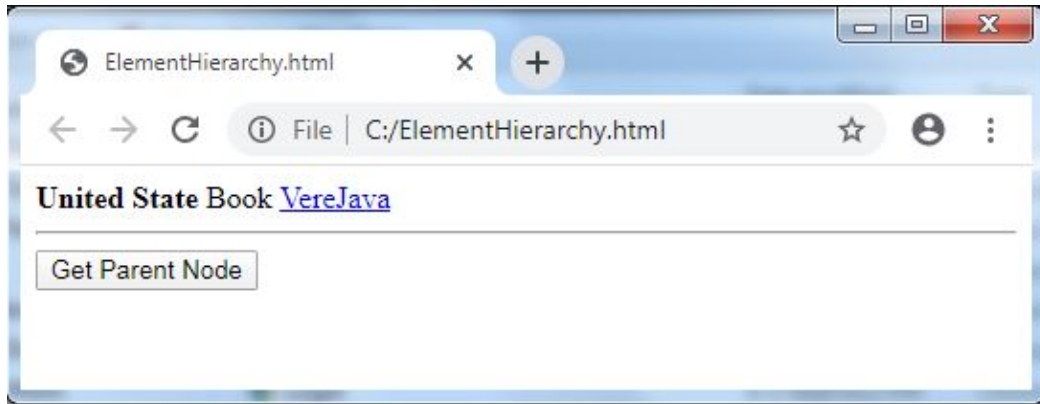
Hierarquia de Elemento



```
<body>
  <div id="div">
    <b>United State</b>
    <span>Book</span>
    <a href="#" >VereJava</a>
  </div>
</body>
```

1. Altere **ElementHierarchy.html** para obter os nós pais.

obj.parentNode: obter nós pais do nó atual



****: tag especifica texto em negrito.

<a>: tag define um hyperlink, que é usado para vincular de uma página a outra.

```
<body>
  <div id="div">
    <b>United State</b>
    <span>Book</span>
    <a href="#" >VereJava</a>
  </div>
  <hr>
  <input type="button" value="Get Parent Node"
onclick="doGetParentNode()" />
</body>

<script type="text/javascript">
  function doGetParentNode()
  {
    var divObj=document.getElementById("div");
    alert(divObj.parentNode.nodeName);
  }
</script>
```



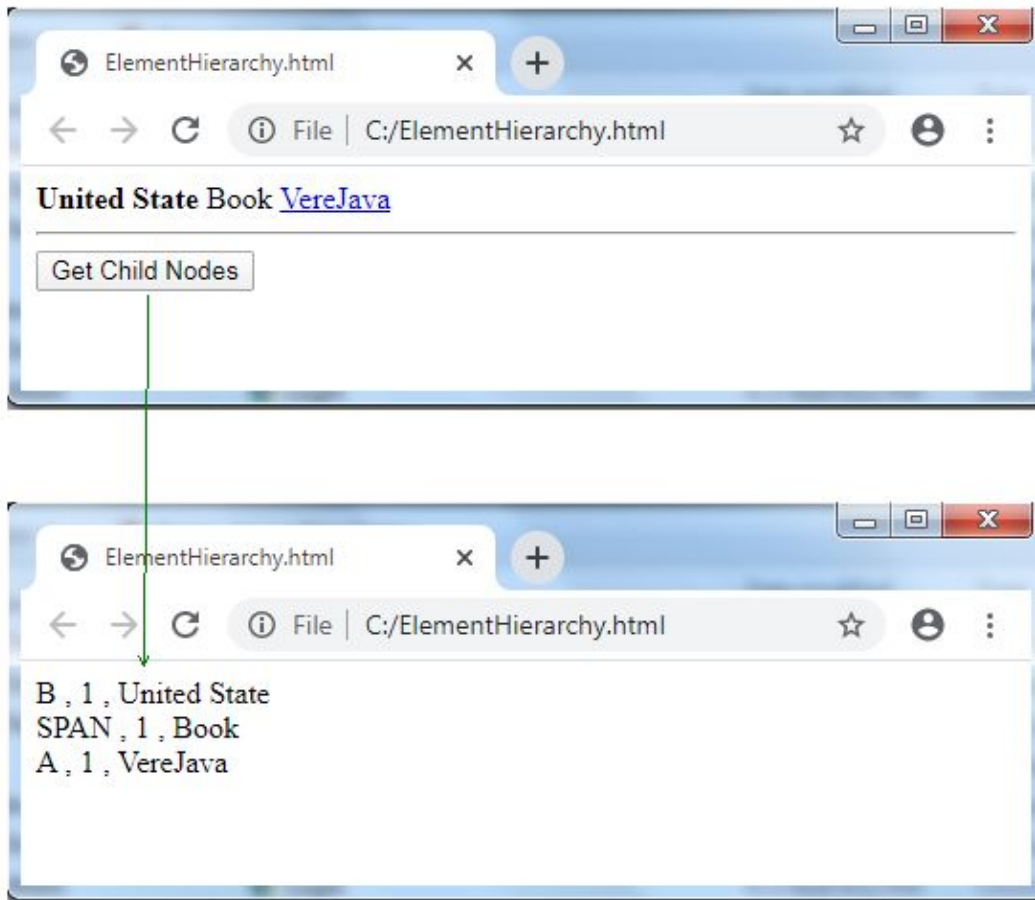

2. Alterar **ElementHierarchy.html** obter todos os nós filhos

obj.childNodes: obter todos os nós filhos do nó atual

obj.nodeName: obter o nome do nó pai

obj.nodeType: obter o tipo do nó pai

obj.innerHTML: obter o conteúdo html do nó pai



```
<body>
  <div id="div">
    <b>United State</b>
    <span>Book</span>
    <a href="#" >VereJava</a>
  </div>
  <hr>
  <input type="button" value="Get Child Nodes"
onclick="doGetChildNodes()" />
</body>

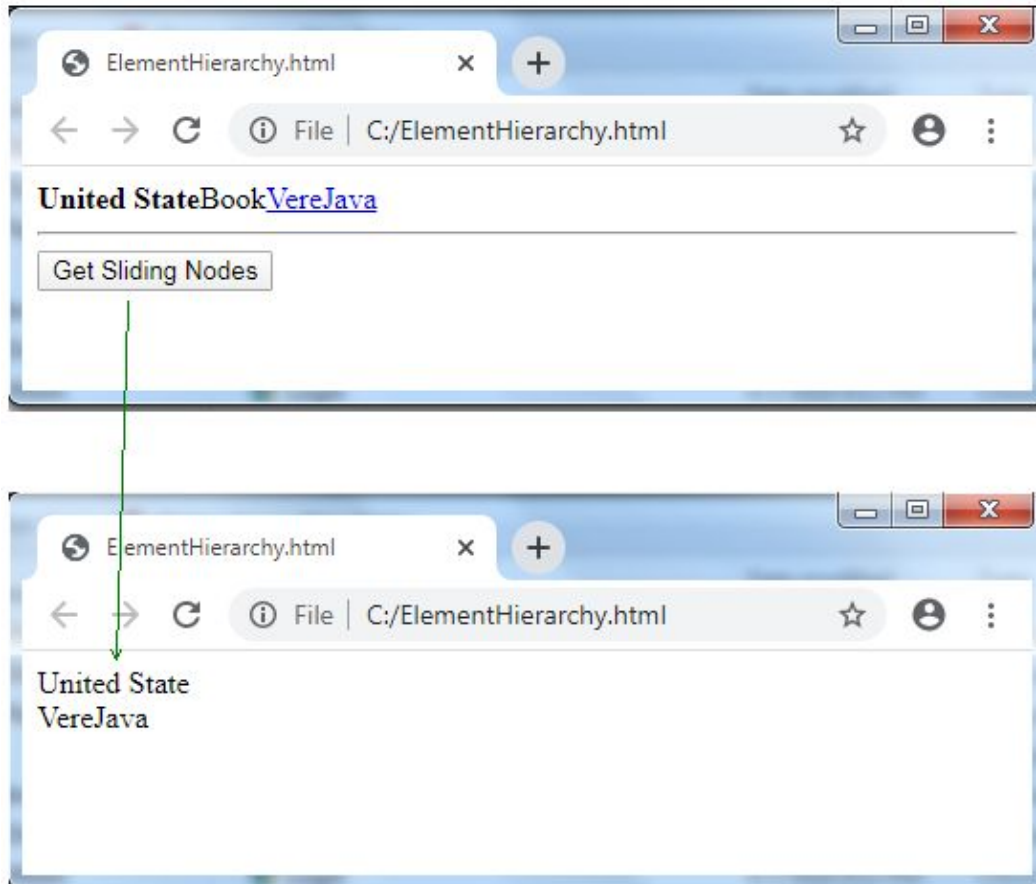
<script type="text/javascript">

  function doGetChildNodes()
  {
    var div=document.getElementById("div");
    var childNodes=div.childNodes;
    for(var i=0;i<childNodes.length;i++)
    {
      if(childNodes[i].nodeType==1)
      {
        document.write(childNodes[i].nodeName+" , ");
        document.write(childNodes[i].nodeType+" , ");
        document.write(childNodes[i].innerHTML+"<br>");
      }
    }
  }
</script>
```

3. Alterar **ElementHierarchy.html** obter o nó anterior e o próximo nó do livro

obj. previousSibling: obter o nó anterior do nó atual

obj. nextSibling: obter o próximo nó do nó atual



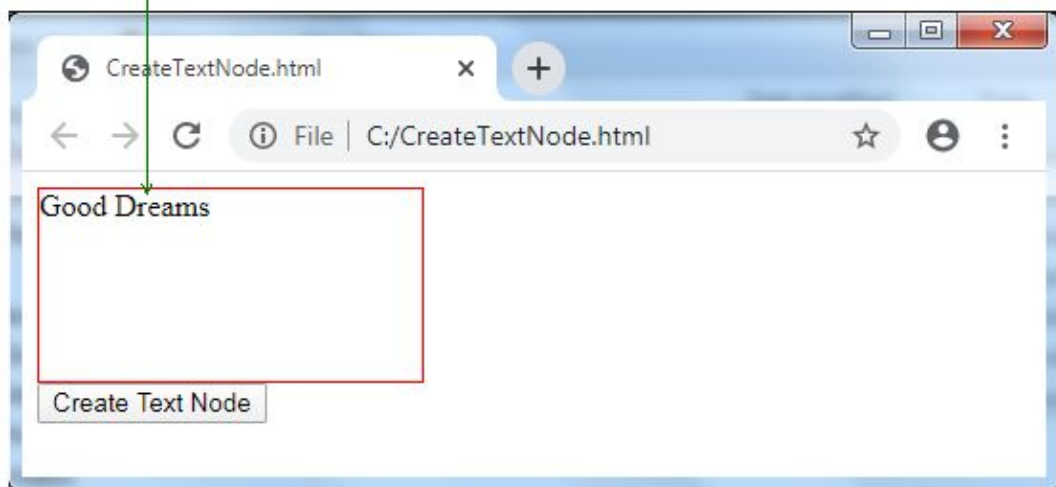
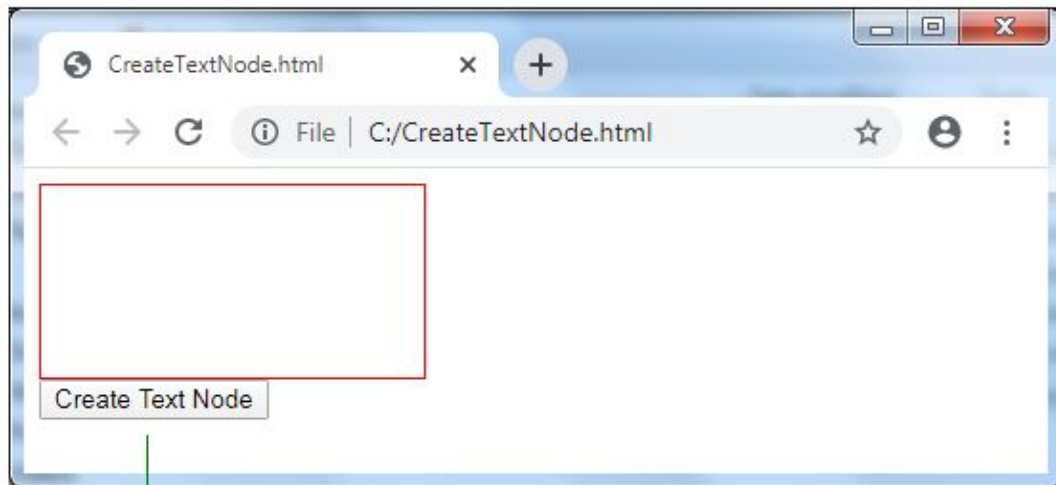
```
<body>
  <div id="div">
    <b>United State</b><span id="book">Book</span><a href="#"
  >VereJava</a>
  </div>
  <hr>
  <input type="button" value="Get Sliding Nodes"
onclick="doGetSliding()" />
</body>

<script type="text/javascript">

  function doGetSliding()
  {
    var bookObj=document.getElementById("book");
    document.write(bookObj.previousSibling.innerHTML+"<br>");
    document.write(bookObj.nextSibling.innerHTML);
  }

</script>
```

Criar Nó de Texto



1. Crie um arquivo: **CreateTextNode.html**

obj. createTextNode: cria um nó de texto com o texto especificado

obj. appendChild: anexa um nó como o último filho de um nó

```
<style>
  div{
    width:200px;
    height:100px;
    border:1px solid #ff0000;
  }
</style>

<div id="div"></div>
<input type="button" value="Create Text Node"
onclick="doCreateTextNode()" />

<script type="text/javascript">

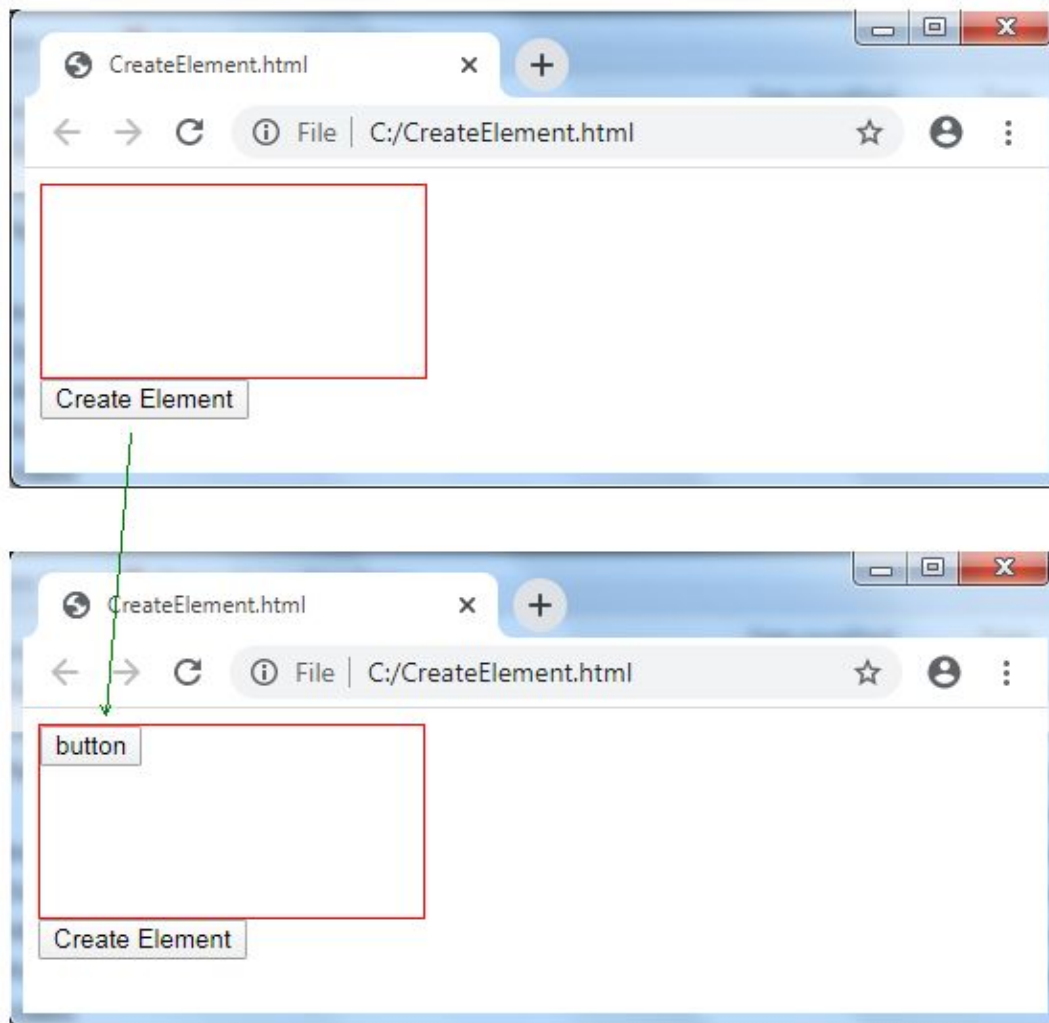
  function doCreateTextNode()
  {
    var divObj = document.getElementById("div");
    var newNode=document.createTextNode("Good Dreams");
    divObj.appendChild(newNode);
  }

</script>
```

2. Crie um arquivo: **CreateElement.html**

obj. createElement: cria um nó de elemento com o nome especificado

obj. appendChild: anexa um nó como o último filho de um nó




```
<style>
  div{
    width:200px;
    height:100px;
    border:1px solid #ff0000;
  }
</style>

<div id="div">
</div>

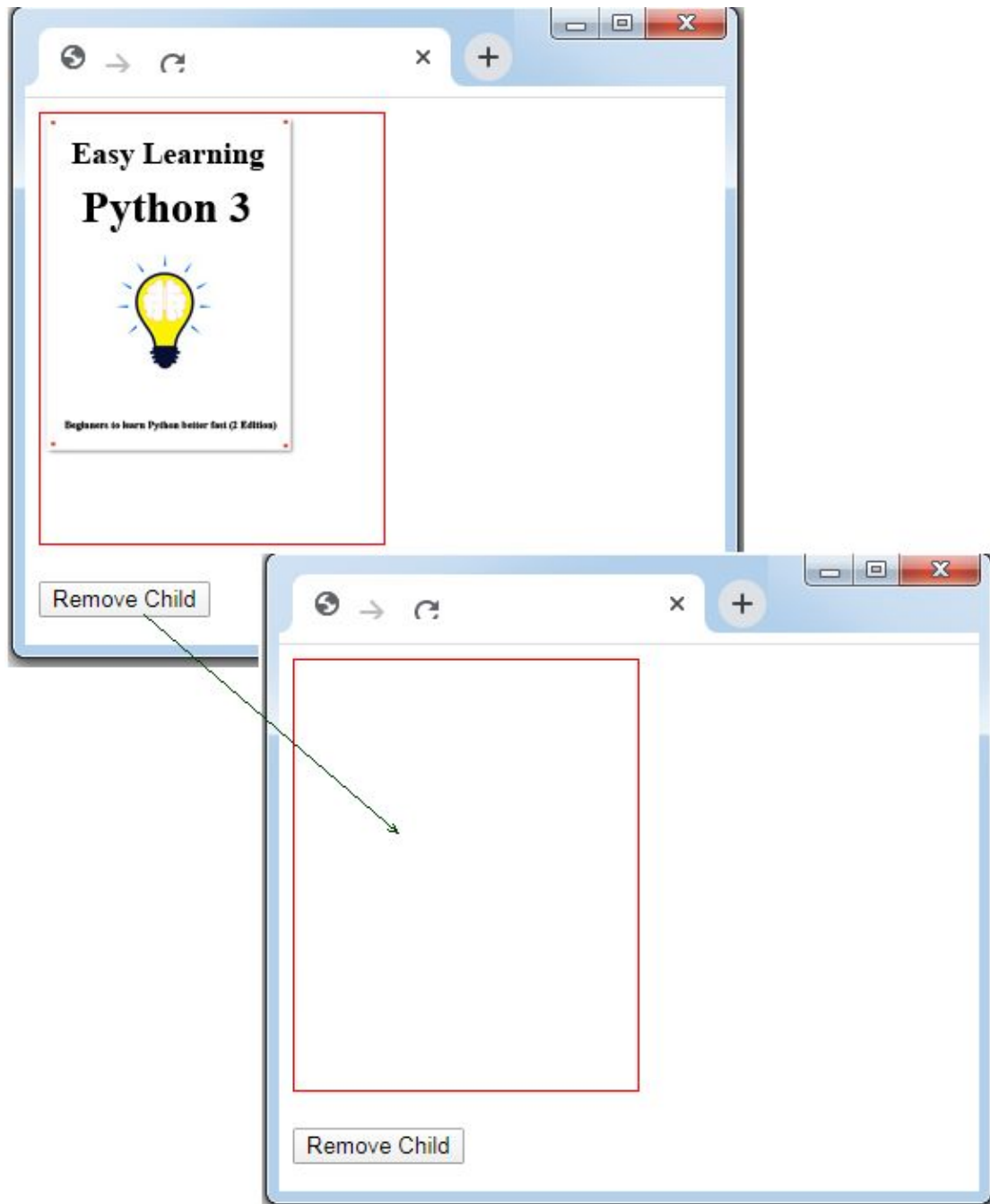
<input type="button" value="Create Element"
onclick="doCreateElement()" />

<script type="text/javascript">

  function doCreateElement()
  {
    var divObj = document.getElementById("div");
    var buttonElement=document.createElement("input");
    buttonElement.type="button";
    buttonElement.value="button";
    divObj.appendChild(buttonElement);
  }

</script>
```

Excluir Nó



1. Crie um arquivo: **DeleteNode.html**

obj. removeChild: remove um nó filho especificado

```
<style>
  div{
    width:200px;
    height:250px;
    border:1px solid #ff0000;
  }
</style>

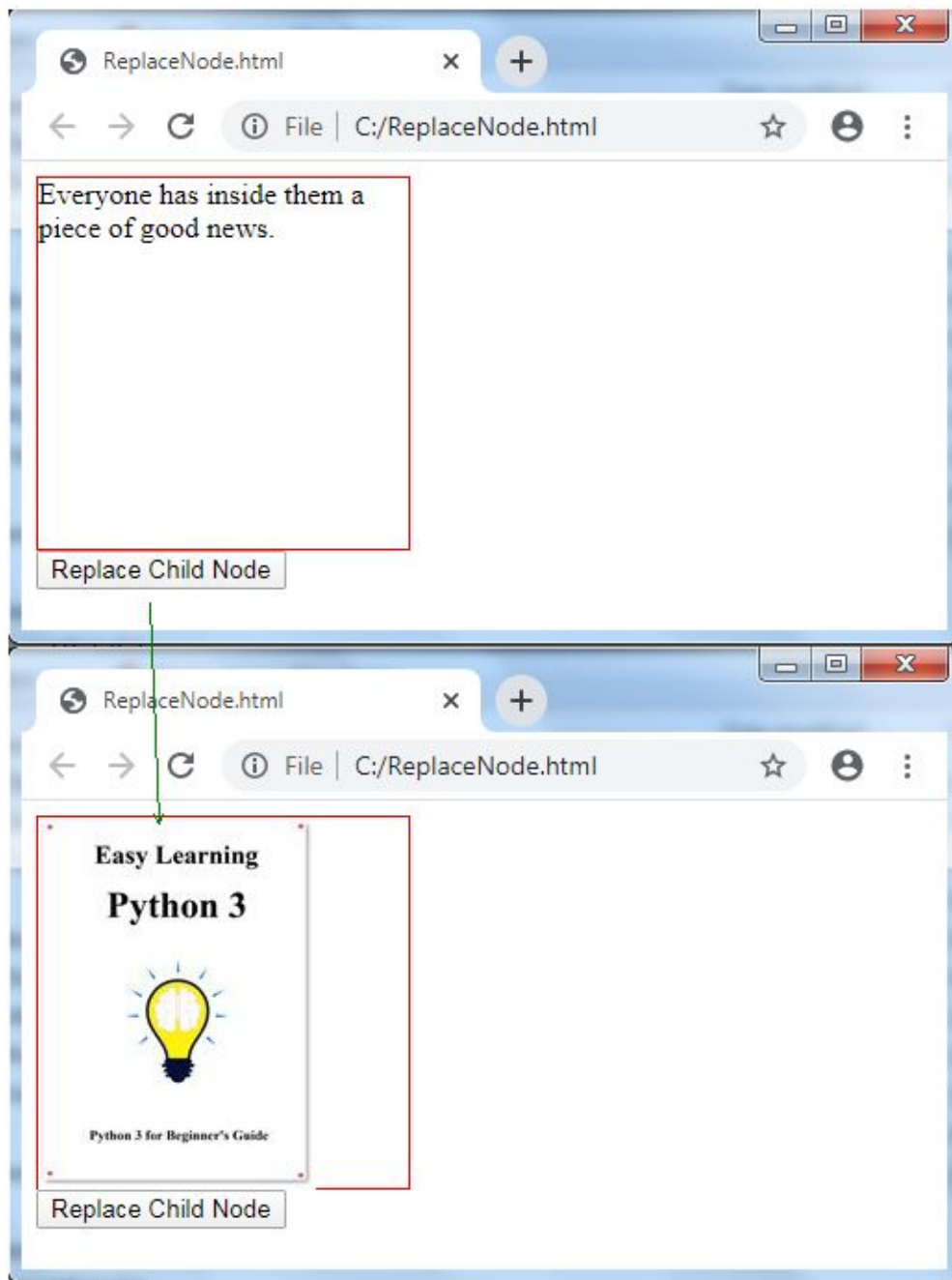
<div id="div1">
  
</div>
<br>
<input type="button" value="Remove Child"
onclick="doRemoveChild()" />

<script type="text/javascript">

  function doRemoveChild()
  {
    var div=document.getElementById("div1");
    var img=document.getElementById("image");
    div.removeChild(img);
  }

</script>
```

Substituir Nó



1. Crie um arquivo: **ReplaceNode.html**

obj. replaceChild: substitui um nó filho por um novo nó

```
<style>
  #div1{
    width:200px;
    height:200px;
    border:1px solid #ff0000;
  }
</style>

<div id="div1">Everyone has inside them a piece of good news.</div>
<input type="button" value="Replace Child Node"
onclick="doReplaceChild()" />

<script type="text/javascript">

  function doReplaceChild()
  {
    var imgNode=document.createElement("img");
    imgNode.src="python.jpg";
    var div1=document.getElementById("div1");
    div1.replaceChild(imgNode,div1.childNodes[0]);
  }

</script>
```

Adicionar Exemplo de Contato

[illegible]

1. Crie um arquivo: **AddContactExample.html**

```
<style>
  #div{
    border:1px solid #ff0000;
    width:550px;
  }
</style>

<a href="javascript:void(0)" onclick="addContact()" >Add
Contact</a>
<div id="div">
  <div id="contact">
    Name:<input type="text" name="user" />
    Telephone:<input type="text" name="telephone" />
    <a href="javascript:void(0)" onclick="deleteNode(this)"
>Delete</a>
  </div>
</div>

<script type="text/javascript">

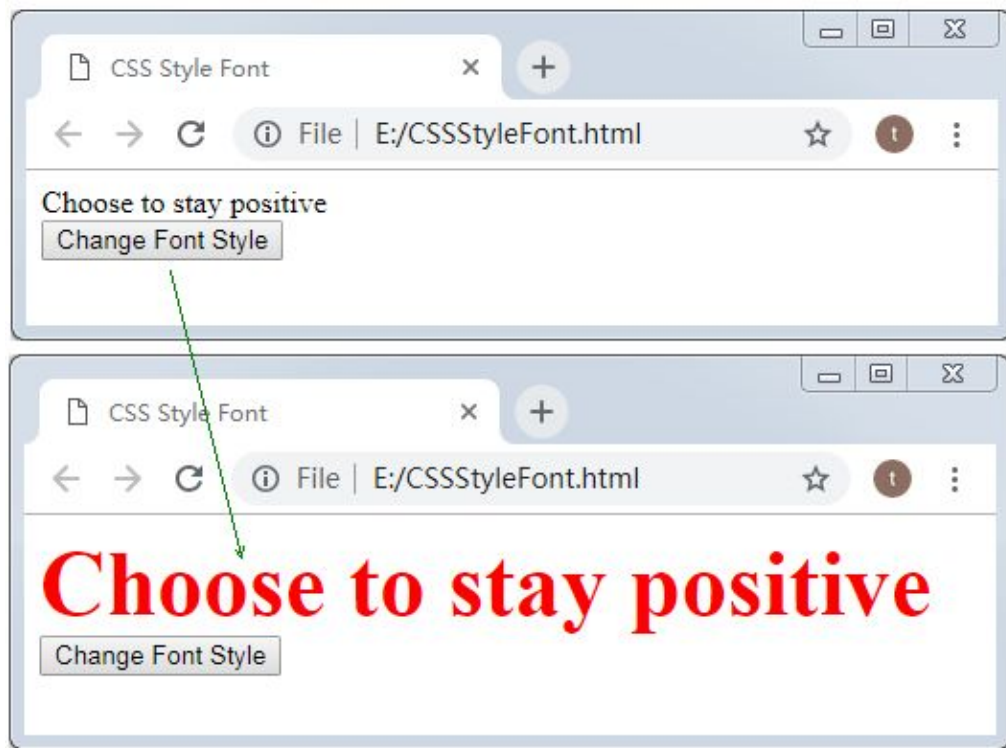
  function addContact()
  {
    var contact=document.getElementById("contact");
    var newNone=contact.cloneNode(true);

    var div=document.getElementById("div");
    div.appendChild(newNone);
  }

  function deleteNode(obj)
  {
    if(obj.parentNode.parentNode.childNodes.length>1)
      obj.parentNode.removeNode(true);
  }
}
```

</script>

Fonte de Estilo CSS



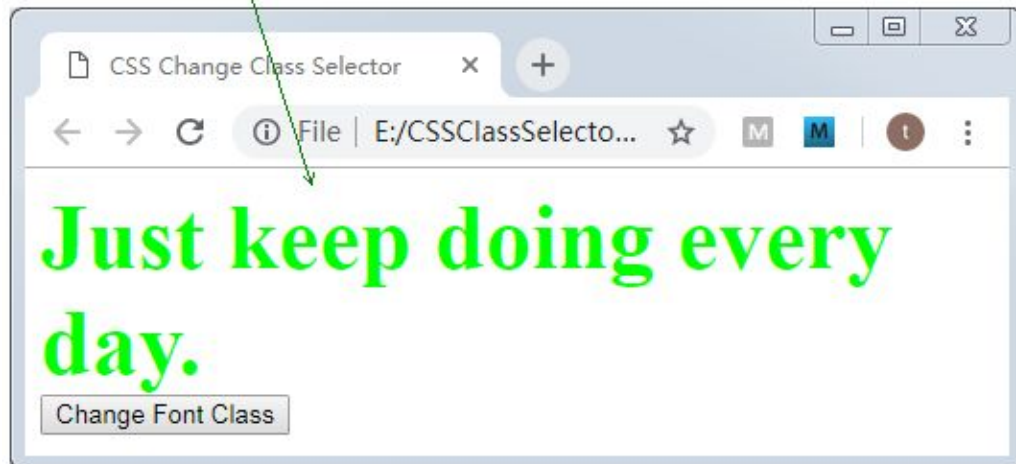
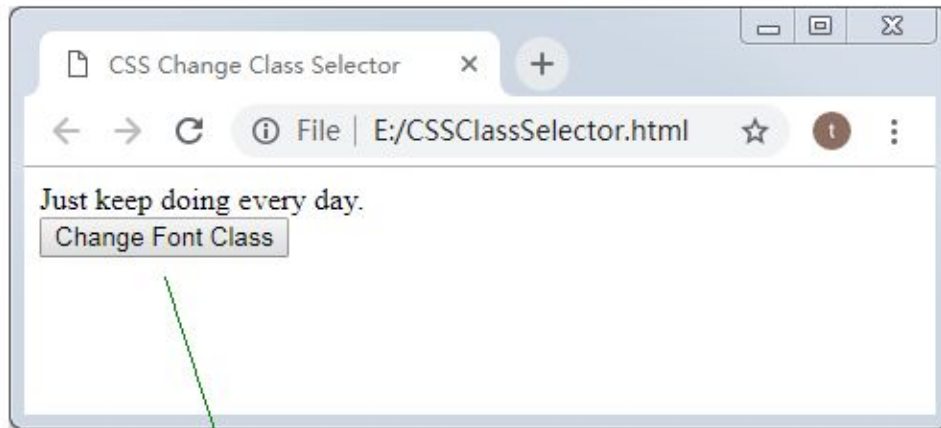
1. Crie um arquivo: **CSSStyleFont.html**

```
<div id="div">Choose to stay positive</div>
<input type="button" value="Change Font Style"
onclick="doChangeFont()" />

<script type="text/javascript">
  function doChangeFont()
  {
    var div=document.getElementById("div");
    div.style.fontSize=48+"px";
    div.style.fontWeight="bold";
    div.style.color="#ff0000";
  }
</script>
```



CSS Alterar Seletor de Classe



1. Crie um arquivo: **CSSClassSelector.html**

```
<style type="text/css">
  .smallFont{
    font-size:12px;
    color:#ff0000;
  }
  .bigFont{
    font-size:48px;
    font-weight:bold;
    color:#00ff00;
  }
</style>

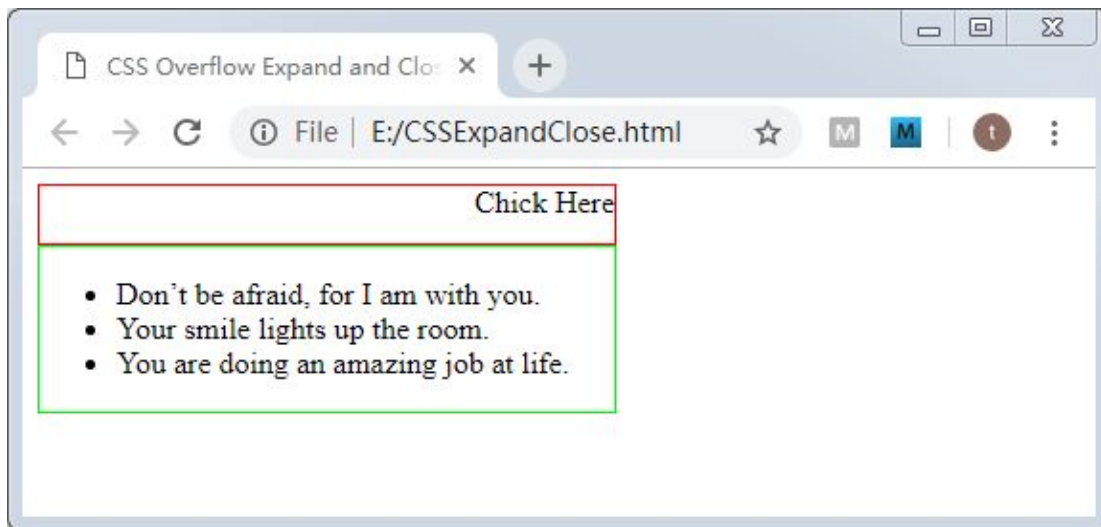
<div id="div">Just keep doing every day.</div>
<input type="button" value="Change Font Class"
onclick="changeFont()" />

<script type="text/javascript">

  function changeFont()
  {
    var div=document.getElementById("div");
    if(div.className=="smallFont")
    {
      div.className="bigFont";
    }
    else
    {
      div.className="smallFont";
    }
  }

</script>
```

Expandir e Fechar CSS Overflow



1. Crie um arquivo: **CSSExpandClose.html**

```
<style type="text/css">
.open{
    display :"" ;
}

.close{
    display :none ;
}

#div{
    width:300px;
    height:30px;
    border:1px solid #ff0000;
    text-align:right;
}

#div2{
    width:300px;
```

```
        border:1px solid #00ff00;
    }
</style>

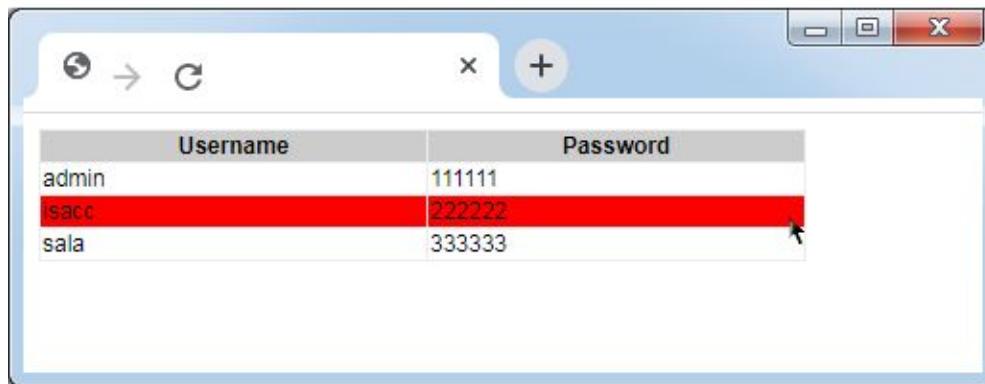
<div id="div" onclick="change()">Chick Here</div>
<div id="div2">
<ul>
    <li>Don't be afraid, for I am with you. </li>
    <li>Your smile lights up the room.</li>
    <li>You are doing an amazing job at life.</li>
</ul>
</div>

<script type="text/javascript">

    function change()
    {
        var div2=document.getElementById("div2");
        if(div2.className=="open")
        {
            div2.className="close";
        }
        else
        {
            div2.className="open";
        }
    }

</script>
```

Destaque de CSS Flutuante



Username	Password
admin	111111
sacc	222222
sala	333333

1. Crie um arquivo: **CSSHighlighting.html**

```
<style type="text/css">
  .over{
    background-color: #ff0000;
  }

  .down{
    background-color: #0000ff;
  }

  table{
    border:1px solid #eeeeee;
    width:400px;
    border-collapse: collapse;
  }

  table th{
    border:1px solid #eeeeee;
    background-color: #cccccc;
  }

  table td{
    border:1px solid #eeeeee;
```

```

    }
</style>

<table>
  <tr>
    <th>Username</th><th>Password</th>
  </tr>
  <tr onmousedown="doDown(this)" onmouseover="doOver(this)"
onmouseout="doOut(this)">
    <td>admin</td><td>111111</td>
  </tr>
  <tr onmousedown="doDown(this)" onmouseover="doOver(this)"
onmouseout="doOut(this)">
    <td>isacc</td><td>222222</td>
  </tr>
  <tr onmousedown="doDown(this)" onmouseover="doOver(this)"
onmouseout="doOut(this)">
    <td>sala</td><td>333333</td>
  </tr>
</table>

<script type="text/javascript">

  function doOver(obj){
    if(obj.className!="down"){
      obj.className="over";
    }
  }

  function doOut(obj){
    if(obj.className!="down"){
      obj.className="";
    }
  }

  function doDown(obj){

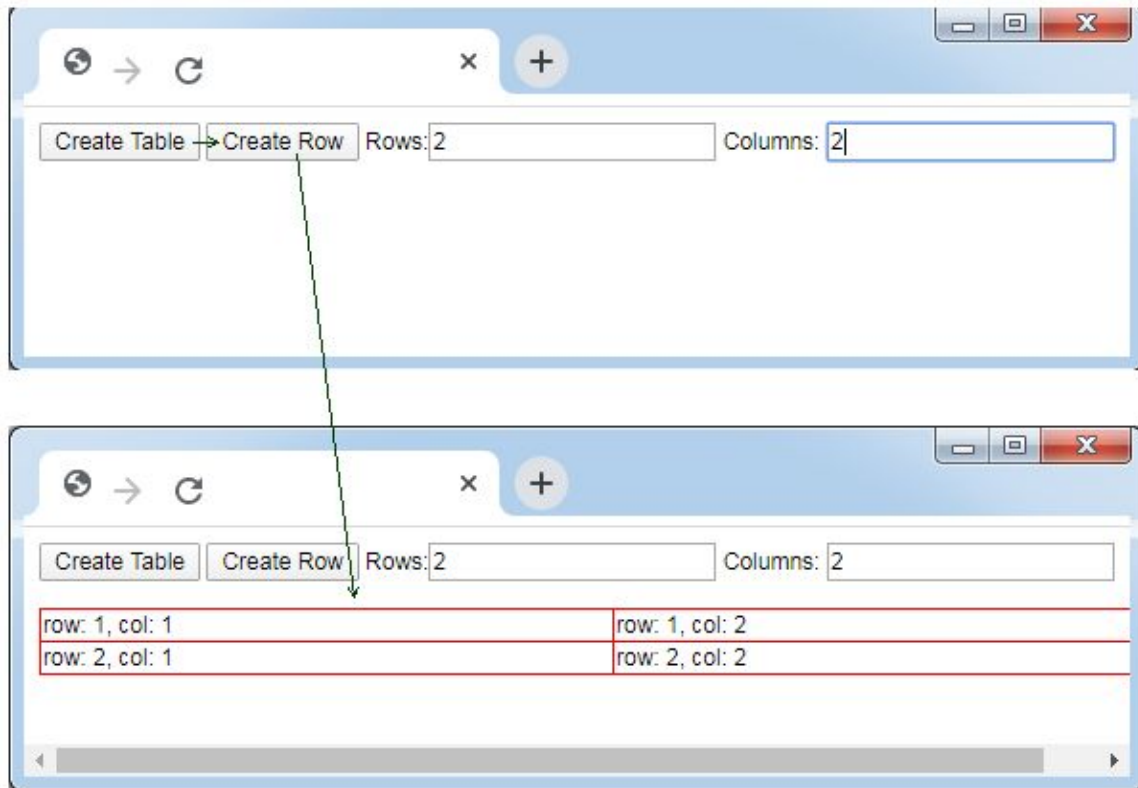
```



```
if(obj.className!="down"){  
    obj.className="down";  
}else{  
    obj.className="over";  
}  
}
```

```
</script>
```

Tabela Criar Colunas de Linhas



1. Crie um arquivo: **TableCreate RowsColumns.html**

table.insertRow: insere uma nova linha (<tr>) em uma determinada <table> e retorna uma referência à nova linha.

table.insertCell: insere uma nova coluna (<td>) em um determinado <tr>, retorna uma referência à nova coluna.

```
<style type="text/css">
  .tableClass{
    width:600px;
    border:1px solid #ff0000;
    border-collapse: collapse;
  }
  .tableClass td{
    border:1px solid #ff0000;
  }
</style>
```

```
<input type="button" value="Create Table"
onclick="doCreateTable()" />
<input type="button" value="Create Row" onclick="doCreateRow()"
/>
Rows:<input type="text" value="" id="row" /> Columns: <input
type="text" value="" id="col" />
<br>
<br>
<div id="div">

</div>
```

```
<script type="text/javascript">
    var table;
    function doCreateTable()
    {
        table=document.createElement("table");
        table.className="tableClass";

        document.getElementById("div").appendChild(table);
    }

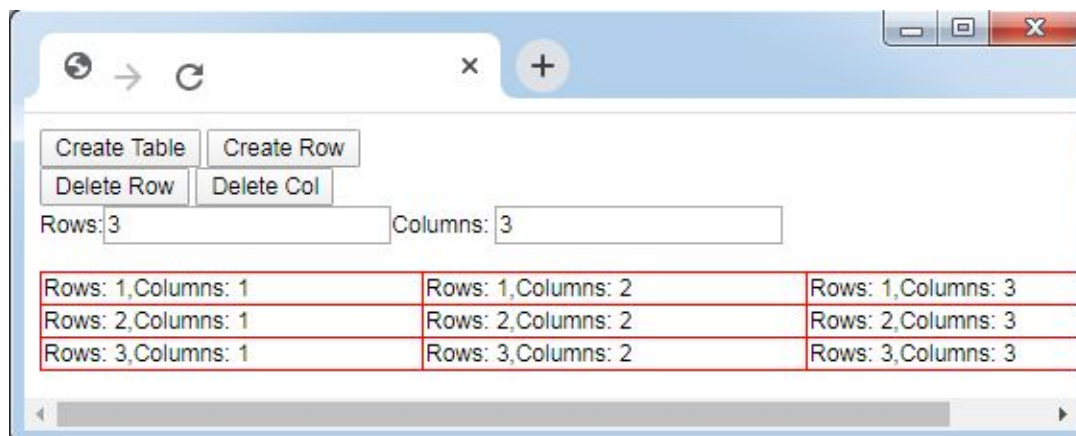
    function doCreateRow()
    {
        var rowNum=parseInt(document.getElementById("row").value);
        var colNum=parseInt(document.getElementById("col").value);

        for(var i=1;i<=rowNum;i++)
        {
            var row=table.insertRow(-1);
            for(var j=1;j<=colNum;j++)
            {
                var cell=row.insertCell(-1);
                cell.innerHTML="row: "+i+", col: "+j;
            }
        }
    }
</script>
```

```
}  
}
```

```
</script>
```

Excluir Coluna da Linha da Tabela



1. Crie um arquivo: **DeleteTableRowColumn.html**

table. deleteCell: exclui uma célula na linha da tabela atual.

```
<style type="text/css">
  .tableClass{
    width:600px;
    border:1px solid #ff0000;
    border-collapse: collapse;
  }

  .tableClass td{
    border:1px solid #ff0000;
  }
</style>
<input type="button" value="Create Table"
onclick="doCreateTable()" />
<input type="button" value="Create Row" onclick="doCreateRow()"
/>
<br>
<input type="button" value="Delete Row" onclick="doDeleteRow()"
/>
<input type="button" value="Delete Col" onclick="doDeleteCol()" />
```

```
<br>
Rows:<input type="text" value="" id="row" />Columns: <input
type="text" value="" id="col" />
<br>
<br>
<div id="div"> </div>
```

```
<script type="text/javascript">
    var table;
    function doCreateTable()
    {
        table=document.createElement("table");
        table.className="tableClass";
        document.getElementById("div").appendChild(table);
    }

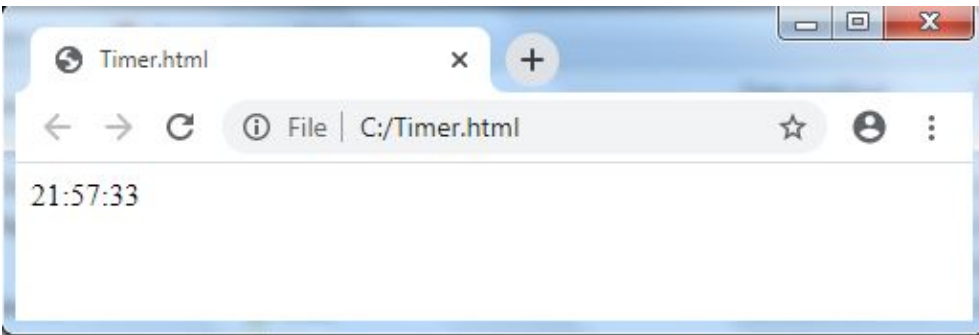
    function doCreateRow()
    {
        var rowNum=parseInt(document.getElementById("row").value);
        var colNum=parseInt(document.getElementById("col").value);

        for(var i=1;i<=rowNum;i++)
        {
            var row=table.insertRow(-1);
            for(var j=1;j<=colNum;j++)
            {
                var cell=row.insertCell(-1);
                cell.innerHTML="Rows: "+i+",Columns: "+j;
            }
        }
    }

    function doDeleteRow()
    {
        var rowNum=parseInt(document.getElementById("row").value);
```

```
    table.deleteRow(rowNum);  
}  
  
function doDeleteCol()  
{  
    var colNum=parseInt(document.getElementById("col").value);  
    var rows=table.rows;  
    for(var i=0;i<rows.length;i++)  
    {  
        rows[i].deleteCell(colNum);  
    }  
}  
</script>
```

Timer



1. Crie um arquivo: **Timer.html**

getFullYear(): Obtenha o ano como um número de quatro dígitos (yyyy)
getMonth(): Obtenha o mês como um número (0-11)
getDate(): Obtenha o dia como um número (1-31)
getHours(): Obtenha a hora (0-23)
getMinutes(): Obtenha o minuto (0-59)
getSeconds(): Obtenha o segundo (0-59)
getTime(): Obtenha o tempo (milissegundos desde 1 de janeiro de 1970)
setInterval(): continue chamando a função até que **clearInterval()** seja chamado.

```
<div id="dateDiv" ></div>

<script type="text/javascript">
    function doSetInterval()
    {
        var dateDiv=document.getElementById("dateDiv");
        var d=new Date();
        var dateString =
d.getHours()+":"+d.getMinutes()+":"+d.getSeconds()

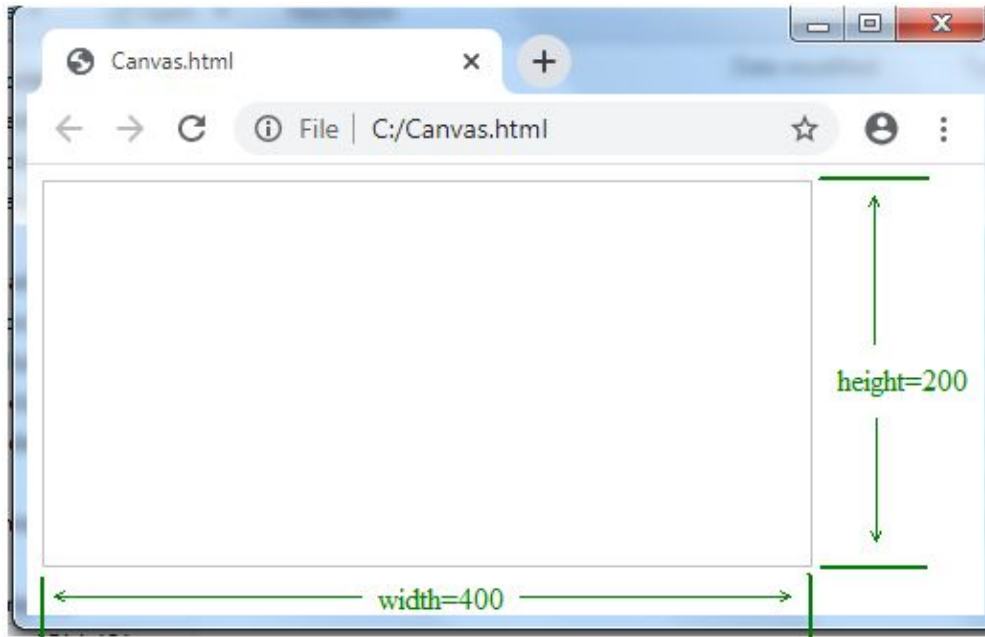
        dateDiv.innerHTML=dateString;
    }
</script>
```



```
setInterval("doSetInterval()",1000);  
</script>
```

Barra de Progresso da Tela

Canvas: O elemento HTML <canvas> é usado para desenhar gráficos em uma página da web.



1. Crie um arquivo: **Canvas.html**

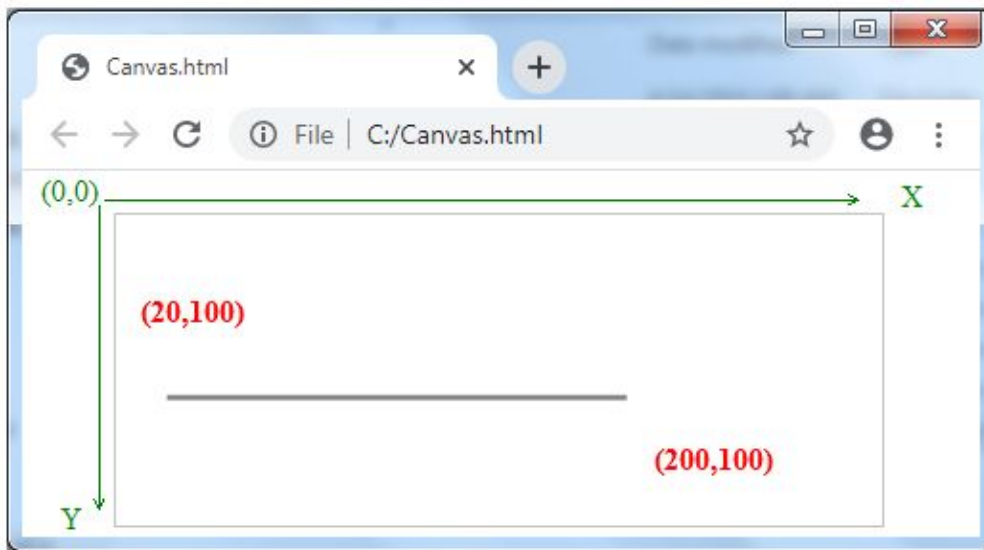
<style>: descreve como os elementos HTML devem ser exibidos na tela.

#id: O seletor #id define o estilo do elemento com o id especificado.

```
<style>
  #myCanvas{
    border:1px solid #c3c3c3;
  }
</style>
```

```
<canvas id="myCanvas" width="400" height="200"></canvas>
```

2. Desenhe uma Linha na Tela



canvas.getContext("2d"): retorna um objeto que fornece métodos para desenhar na tela.

```
<style>
  #myCanvas{
    border:1px solid #c3c3c3;
  }
</style>

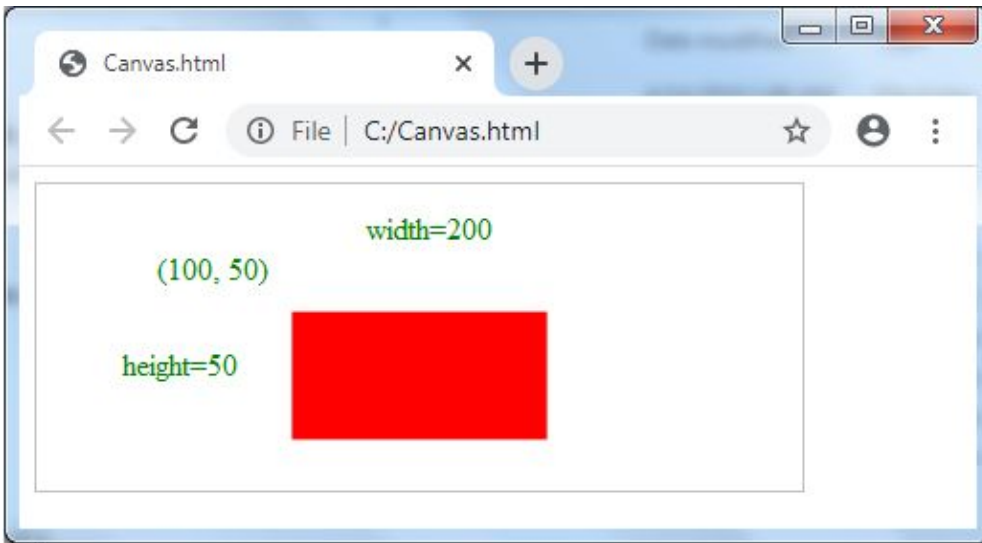
<canvas id="myCanvas" width="400" height="200"></canvas>

<script type="text/javascript">
  var canvas = document.getElementById("myCanvas");

  // Comece um caminho, crie um movimento de linha de (20.100) para
  (200.100):
  var context = canvas.getContext("2d");
  context.beginPath();
  context.moveTo(20,100);
  context.lineTo(200,100);
  context.stroke();
```

</script>

3. Desenhe um Retângulo na Tela



context.fillRect(x, y, width, height): desenhe um retângulo na tela.

context.fillStyle: a cor para preencher o retângulo (red: #ff0000, green: #00ff00, blue: #0000ff).

```
<style>
  #myCanvas{
    border:1px solid #c3c3c3;
  }
</style>
<canvas id="myCanvas" width="400" height="200"></canvas>
<script type="text/javascript">
  var canvas = document.getElementById("myCanvas");

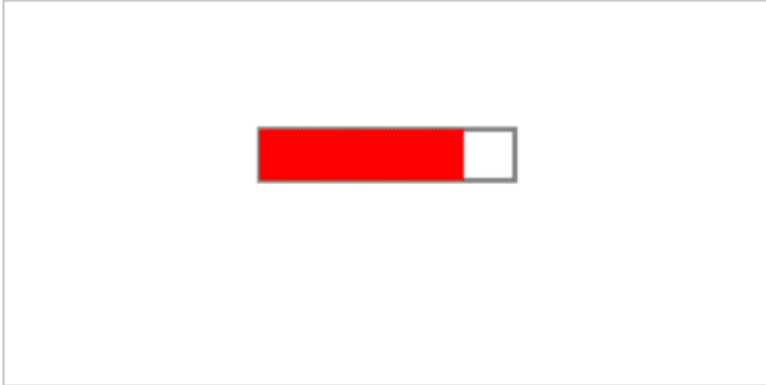
  // Comece um caminho, crie um retângulo
  var context = canvas.getContext("2d");
  context.fillStyle = "#ff0000";
  context.fillRect(100, 50, 100, 50);

</script>
```

4. Desenhar Animação da Barra de Progresso

4.1. Desenhe um retângulo vazio primeiro

4.2. Desenhe um retângulo sólido, a largura aumentando em 10 a cada 200 milissegundos até largura = 100



context. strokeRect(x, y, width, height): desenhe um retângulo de traço na tela.

```
<style>
  #myCanvas{
    border:1px solid #c3c3c3;
  }
</style>
<canvas id="myCanvas" width="400" height="200"></canvas>
<script type="text/javascript">
  var canvas = document.getElementById("myCanvas");
  var context = canvas.getContext("2d");

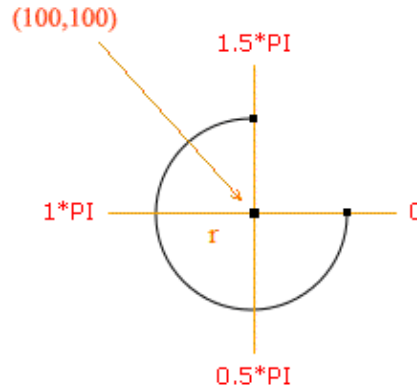
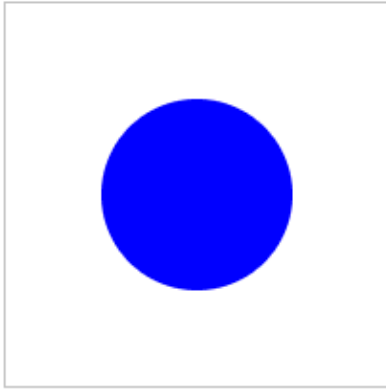
  context.strokeRect(100, 50, 100, 20);

  var time = setInterval("drawProgressBar()",200);
  var width = 0;
  function drawProgressBar(){
    context.fillStyle = "#ff0000";
    context.fillRect(100, 50, width, 20);
    width = width + 10;
    if(width>100){
      clearInterval(time);
    }
  }
</script>
```

```
}  
}  
</script>
```

Tela de Pintura Ball Bounce

1. Desenhe um círculo na tela center=(100, 100), r=50 from 0 to 2*PI.



1. Crie um arquivo: **ball.html**

context.arc(x,y,r,sAngle,eAngle): Gire 360 graus no sentido anti-horário com x, y como o centro e r como o raio.

```
<style>
  #myCanvas{
    border:1px solid #c3c3c3;
    position: absolute;
  }
</style>

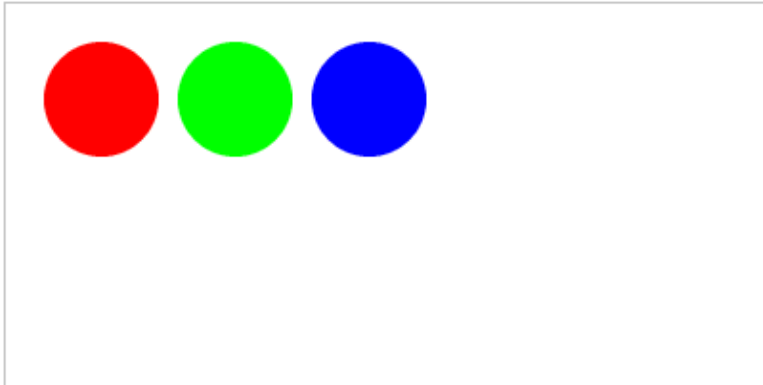
<canvas id="myCanvas" width="200" height="200"></canvas>

<script type="text/javascript">
  var canvas = document.getElementById("myCanvas");
  var context = canvas.getContext("2d");

  context.fillStyle = "#0000ff";
  context.beginPath();
  context.arc(100,100,50,0,2*Math.PI);
  context.fill();
```


</script>

2. Crie uma função **drawBall(x, y, r, color)** para desenhar 3 círculos na tela.



```
<style>
  #myCanvas{
    border:1px solid #c3c3c3;
    position: absolute;
  }
</style>

<canvas id="myCanvas" width="400" height="200"></canvas>

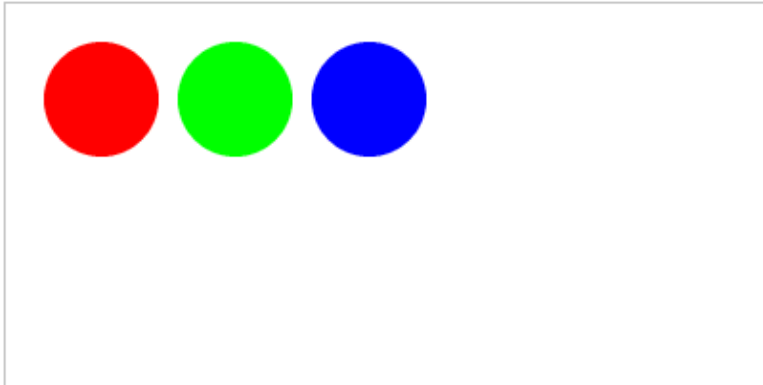
<script type="text/javascript">
  var canvas = document.getElementById("myCanvas");
  var context = canvas.getContext("2d");

  function drawBall(x, y, r, color){
    context.fillStyle = color;
    context.beginPath();
    context.arc(x,y,r,0,2*Math.PI);
    context.fill();
  }

  drawBall(50,50,30,"#ff0000");
  drawBall(120,50,30,"#00ff00");
  drawBall(190,50,30,"#0000ff");
```

</script>

3. Crie uma classe **Ball** e então crie um método **draw()** para desenhar 3 círculos na tela.



```
<style>
  #myCanvas{
    border:1px solid #c3c3c3;
    position: absolute;
  }
</style>

<canvas id="myCanvas" width="400" height="200"></canvas>

<script type="text/javascript">
  var canvas = document.getElementById("myCanvas");
  var context = canvas.getContext("2d");

  class Ball{
    constructor(x, y, r, color){
      this.x = x;
      this.y = y;
      this.r = r;
      this.color = color;
    }

    draw(){
      context.fillStyle = this.color;
      context.beginPath();
```

```
    context.arc(this.x,this.y,this.r,0,2*Math.PI);  
    context.fill();  
  }  
}
```

//new Ball o método construtor será chamado automaticamente

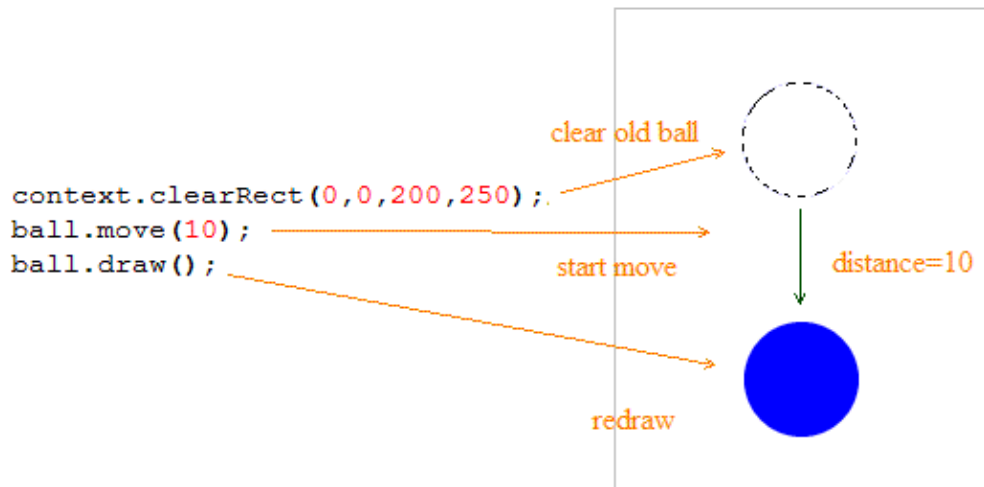
```
var redBall = new Ball(50,50,30,"#ff0000");  
redBall.draw();
```

```
var greenBall = new Ball(120,50,30,"#00ff00");  
greenBall.draw();
```

```
var blueBall = new Ball(190,50,30,"#0000ff");  
blueBall.draw();
```

```
</script>
```

4. Crie uma bola e depois desça.



```
<style>
```

```
#myCanvas{
  border:1px solid #c3c3c3;
  position: absolute;
}
```

```
</style>
```

```
<canvas id="myCanvas" width="200" height="250"></canvas>
```

```
<script type="text/javascript">
```

```
var canvas = document.getElementById("myCanvas");
var context = canvas.getContext("2d");
```

```
class Ball{
```

```
  constructor(x, y, r, color){
    this.x = x;
    this.y = y;
    this.r = r;
    this.color = color;
  }
```

```
  draw(){
```

```
    context.fillStyle = this.color;
    context.beginPath();
    context.arc(this.x,this.y,this.r,0,2*Math.PI);
```

```
        context.fill();
    }

    move(distance){
        this.y = this.y + distance;
    }
}

var ball = new Ball(100,50,30,"#0000ff");
ball.draw();

function continueRun(){
    context.clearRect(0,0,200,250);// tela clara
    ball.move(10);
    ball.draw();
}

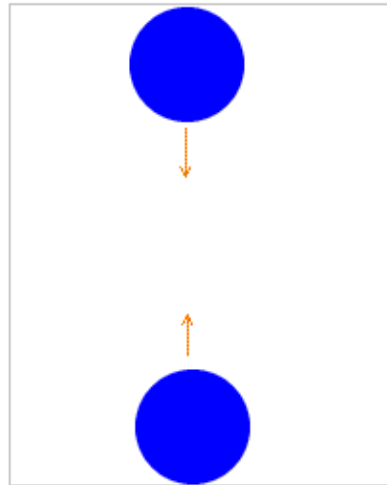
// Executar a cada 200 milissegundos
var time = setInterval("continueRun()",200);

</script>
```

5. A bola quica para cima e para baixo

If the ball hits the top and bounces downwards

```
if(ball.getY()-ball.getR() > 0 && direction == "up"){  
    ball.move(-10);  
    ball.draw();  
}else{  
    direction = "down"  
}
```



If the ball hits the bottom and bounces upwards

```
var direction = "down";  
if(ball.getY()+ball.getR() < canvasHeight && direction == "down"){  
    ball.move(10);  
    ball.draw();  
}else{  
    direction = "up"  
}
```



```
<style>
```

```
  #myCanvas{  
    border:1px solid #c3c3c3;  
    position: absolute;  
  }
```

```
</style>
```

```
<canvas id="myCanvas" width="200" height="250"></canvas>
```

```
<script type="text/javascript">
```

```
  var canvas = document.getElementById("myCanvas");  
  var context = canvas.getContext("2d");  
  var canvasWidth=200;  
  var canvasHeight=250;
```

```
  class Ball{
```

```
    constructor(x, y, r, color){  
      this.x = x;  
      this.y = y;  
      this.r = r;  
      this.color = color;  
    }
```

```
    draw(){  
      context.fillStyle = this.color;  
      context.beginPath();  
      context.arc(this.x,this.y,this.r,0,2*Math.PI);  
      context.fill();  
    }
```

```
    move(distance){  
      this.y = this.y + distance;  
    }
```

```
    getY(){
```

```

        return this.y;
    }

    getR(){
        return this.r;
    }
}

var ball = new Ball(100,50,30,"#0000ff");
ball.draw();

var direction = "down";
function continueRun(){
    context.clearRect(0,0,canvasWidth,canvasHeight);
    // Se a bola atingir o fundo e saltar para cima
    if(ball.getY()+ball.getR() < canvasHeight && direction == "down")
{
    ball.move(10);
    ball.draw();
} else{
    direction = "up"
}
    // Se a bola atingir o topo e quicar para baixo
    if(ball.getY()-ball.getR() > 0 && direction == "up"){
        ball.move(-10);
        ball.draw();
    } else{
        direction = "down"
    }
}

//Run every 200 milliseconds
var time = setInterval("continueRun()",200);

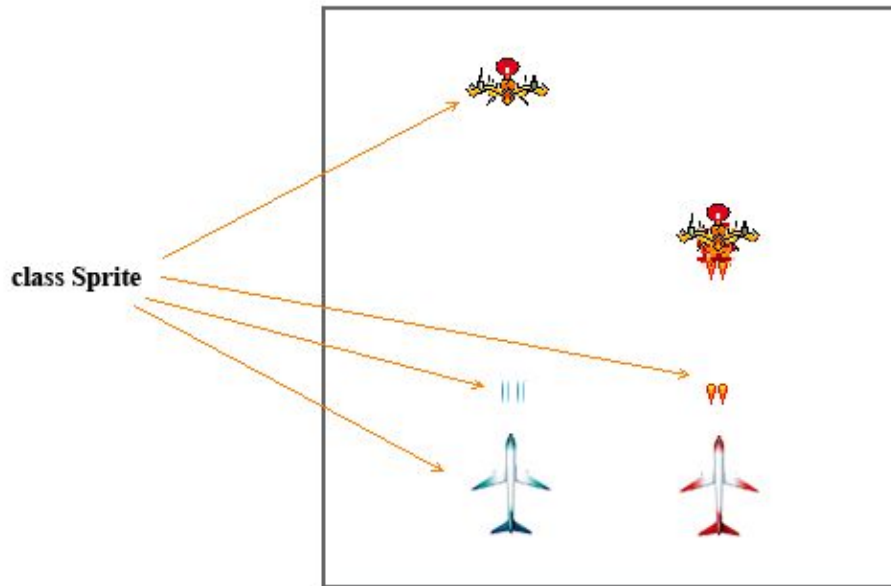
</script>

```

Criar Classe (**Create Class**)

Quase tudo é um objeto, com suas propriedades e métodos. Class é como um construtor de objeto para a criação de objetos.

No jogo, aviões(**planes**), balas(**bullets**), bombas(**bombs**) e aviões inimigos(**enemy planes**) são chamados de classe Sprite



1. Crie um arquivo: **class.html**

O nome do método do construtor deve ser: **constructor**

constructor: quando criar o objeto será chamado automaticamente

```
<script type="text/javascript">

// definindo uma classe Sprite
class Sprite{

    // método construtor
    constructor(name, color){
        this.name = name;
        this.color = color;
    }

    // método de definição, this: é o objeto atual
    draw(){
        var text = this.color + " " + this.name + " is drawn on canvas";
        document.write(text + "<br>");
    }
}

// criar o construtor do objeto será chamado automaticamente
blueSprite = new Sprite("Airplane", "Blue");
// Use propriedades e métodos de acesso sprite
blueSprite.draw();

redSprite = new Sprite("Bullet", "Blue");
redSprite.draw();

</script>
```

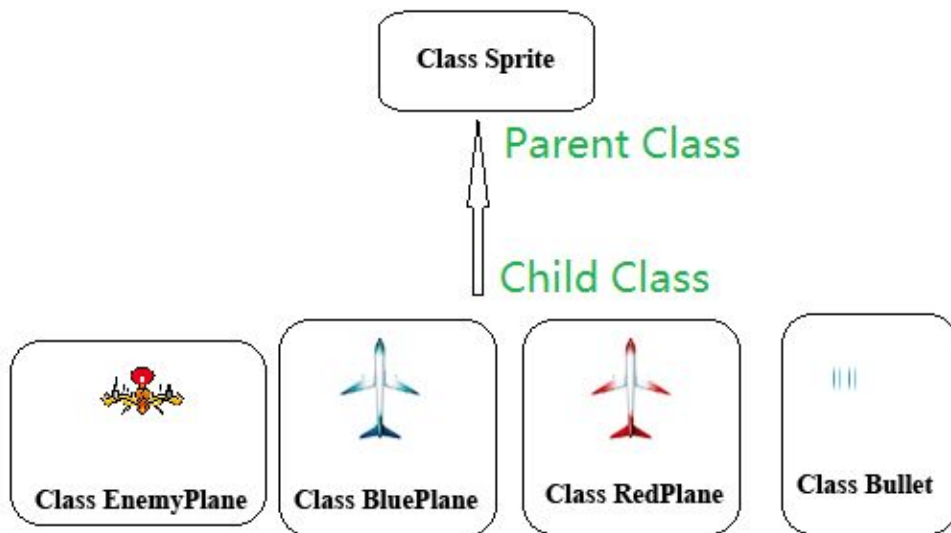
Resultado:

Blue Airplane is drawn on canvas
Blue Bullet is drawn on canvas

Inheritance

Todas as classes **EnemyPlane**, **BluePlane**, **RedPlane**, **Bullet** herdadas de **Sprite**

1. **Sprite** é chamado de classe pai
2. **EnemyPlane**, **BluePlane**, **RedPlane**, **Bullet** são chamados de classe infantil
3. A classe filha pode chamar as propriedades e métodos da classe pai



1. Crie um arquivo: **Inheritance.html**

```
<script type="text/javascript">
```

```
class Sprite{
```

```
    constructor(name, color){
        this.name = name;
        this.color = color;
    }
```

```
    draw(){
```

```
        var text = this.color + " "+this.name+" is drawn on canvas";
        document.write(text+"<br>");
```

```
    }  
  }  
  
  // BluePlane é filho, estende é herdado, Sprite é pai  
  class BluePlane extends Sprite{  
  
  }  
  
  class RedPlane extends Sprite{  
  
  }  
  
  class Bullet extends Sprite{  
  
  }  
  
  blueSprite = new Sprite("Airplane","Blue");  
  blueSprite.draw();  
  
  redSprite = new Sprite("Bullet","Blue");  
  redSprite.draw();  
  
  redBullet= new Bullet("Bullet","Red")  
  redBullet.draw()  
  
</script>
```

Resultado:

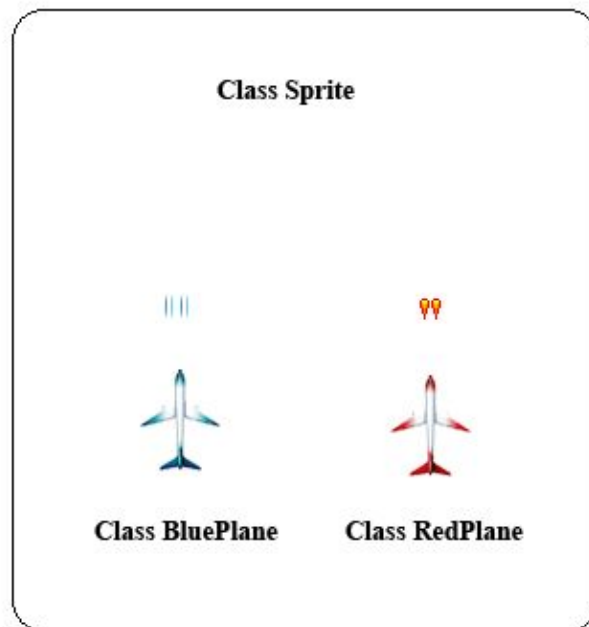
Blue Airplane is drawn on canvas
Red Airplane is drawn on canvas
Red Bullet is drawn on canvas

Sobrepôr (Override)

Sobrepôr (Override):

a classe filha substitui o método da classe pai,
a instância da classe filha irá chamar o método sobrescrito

Sprite pode disparar uma bala, mas BluePlane pode disparar uma bala azul e RedPlane pode disparar uma bala vermelha.




```
<script type="text/javascript">

  class Sprite{
    shoot(){
      document.write("fire bullet <br>");
    }
  }

  // BluePlane Inherit Sprite
  class BluePlane extends Sprite{
    shoot(){
      document.write("BluePlane fire blue bullet <br>");
    }
  }

  // RedPlane Inherit Sprite
  class RedPlane extends Sprite{
    shoot(){
      document.write("RedPlane fire red bullet <br>");
    }
  }

  bluePlane = new BluePlane()
  bluePlane.shoot();

  redPlane = new RedPlane()
  redPlane.shoot();

</script>
```

Resultado:

BluePlane fire blue bullet
RedPlane fire red bullet

Airplane Game

1. Desenhe uma imagem na tela.

1.1 coloque a imagem de 2 aviões no diretório

[images/blue_plane.png](#).

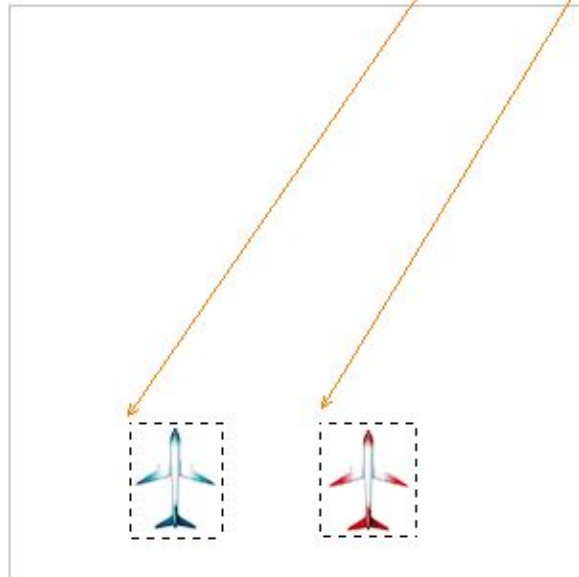
[images/red_plane.png](#)



1.2 desenhe [images/blue_plane.png](#), [images/red_plane.png](#) na tela.

```
var bluePlaneImage = new Image();
bluePlaneImage.src = "./images/blue_plane.png";
bluePlaneImage.onload = function() {
    context.drawImage(bluePlaneImage, 60, 220);
};

var redPlaneImage = new Image();
redPlaneImage.src = "./images/red_Plane.png";
redPlaneImage.onload = function() {
    context.drawImage(redPlaneImage, 160, 220);
};
```



Todas as imagens em **KidsLearningJavascriptImages.zip** podem ser baixadas neste link:

<http://en.verejava.com/download.jsp?id=1>

1. Crie um arquivo: **DrawImage.html**

Image.onload: evento ocorre quando uma imagem foi carregada.
drawImage(): desenha uma imagem na tela.

```
<style>
  #myCanvas{
    border:1px solid #c3c3c3;
    position: absolute;
  }
</style>

<canvas id="myCanvas" width="300" height="300"></canvas>

<script type="text/javascript">
  var canvas = document.getElementById("myCanvas");
  var context = canvas.getContext("2d");

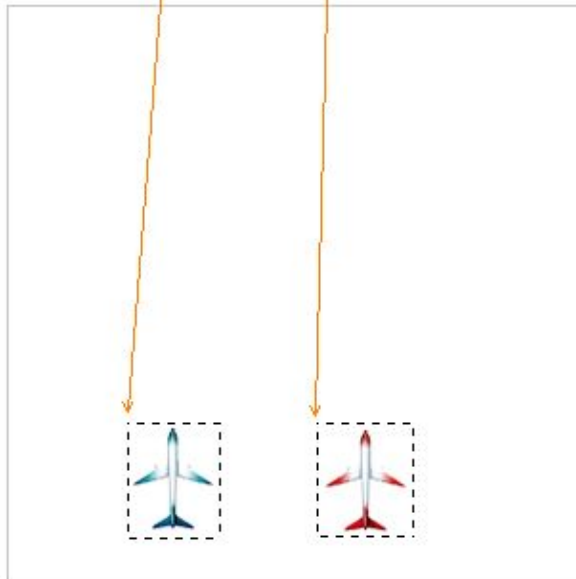
  var bluePlaneImage = new Image();
  bluePlaneImage.src = "./images/blue_plane.png";
  bluePlaneImage.onload = function(){
    context.drawImage(bluePlaneImage,60,220);
  };

  var redPlaneImage = new Image();
  redPlaneImage.src = "./images/red_Plane.png";
  redPlaneImage.onload = function(){
    context.drawImage(redPlaneImage,160,220);
  };
</script>
```

3. Crie a classe **Airplane** e, em seguida, crie o método **draw** para criar a imagem na tela.

```
var bluePlaneImage = new Image();
bluePlaneImage.src = "../images/blue_plane.png";
var bluePlane = new Airplane(bluePlaneImage, 60, 220);
bluePlaneImage.onload = function() {
    bluePlane.draw(context);
};

var redPlaneImage = new Image();
redPlaneImage.src = "../images/red_Plane.png";
var redPlane = new Airplane(redPlaneImage, 160, 220);
redPlaneImage.onload = function() {
    redPlane.draw(context);
};
```



```

<style>
  #myCanvas{
    border:1px solid #c3c3c3;
    position: absolute;
  }
</style>

<canvas id="myCanvas" width="300" height="300"></canvas>

<script type="text/javascript">
  var canvas = document.getElementById("myCanvas");
  var context = canvas.getContext("2d");

  class Airplane{
    constructor(img, x, y){
      this.x = x;
      this.y = y;
      this.img = img;
    }

    draw(context){
      context.drawImage(this.img,this.x,this.y);
    }
  }

  ////////// test //////////

  var bluePlaneImage = new Image();
  bluePlaneImage.src = "./images/blue_plane.png";
  var bluePlane = new Airplane(bluePlaneImage,60,220);
  bluePlaneImage.onload = function(){
    bluePlane.draw(context);
  };

  var redPlaneImage = new Image();

```

```
redPlaneImage.src = "./images/red_Plane.png";  
var redPlane = new Airplane(redPlaneImage,160,220);  
redPlaneImage.onload = function() {  
    redPlane.draw(context);  
};  
</script>
```

4. Evento Keypress

4.1 Se clicar nas teclas para cima, para baixo, para a esquerda e para a direita, a função **doKeyDown** será chamada.

```
canvas.addEventListener('keydown', doKeyDown, true);  
canvas.focus();  
window.addEventListener('keydown', doKeyDown, true);
```

```
function doKeyDown(e) {  
    var keyID = e.keyCode ? e.keyCode : e.which;  
    if(keyID === 37) { //left arrow  
        alert("left");  
        e.preventDefault();  
    }  
    if(keyID === 38) { //up arrow  
        alert("up");  
        e.preventDefault();  
    }  
    if(keyID === 39) { //right arrow  
        alert("right");  
        e.preventDefault();  
    }  
    if(keyID === 40) { //down arrow  
        alert("down");  
        e.preventDefault();  
    }  
}
```



```
<style>
  #myCanvas{
    border:1px solid #c3c3c3;
    position: absolute;
  }
</style>

<canvas id="myCanvas" width="300" height="300"></canvas>

<script type="text/javascript">
  var canvas = document.getElementById("myCanvas");
  var context = canvas.getContext("2d");

  class Airplane{

    constructor(img, x, y){
      this.x = x;
      this.y = y;
      this.img = img;
    }

    draw(context){
      context.drawImage(this.img,this.x,this.y);
    }
  }

  ////////// test //////////

  var bluePlaneImage = new Image();
  bluePlaneImage.src = "./images/blue_plane.png";
  var bluePlane = new Airplane(bluePlaneImage,60,220);
  bluePlaneImage.onload = function(){
    bluePlane.draw(context);
  };
</script>
```



```
var redPlaneImage = new Image();
redPlaneImage.src = "./images/red_Plane.png";
var redPlane = new Airplane(redPlaneImage,160,220);
redPlaneImage.onload = function(){
    redPlane.draw(context);
};

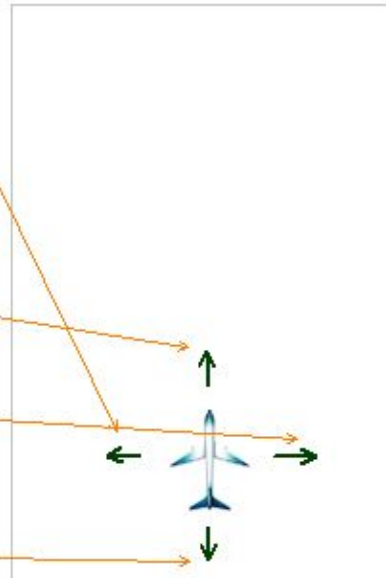
canvas.addEventListener('keydown', doKeyDown,true);
canvas.focus();
window.addEventListener('keydown', doKeyDown, true);

function doKeyDown(e) {
    var keyID = e.keyCode ? e.keyCode :e.which;
    if(keyID === 37) { //left arrow
        alert("left");
        e.preventDefault();
    }
    if(keyID === 38) { //up arrow
        alert("up");
        e.preventDefault();
    }
    if(keyID === 39) { //right arrow
        alert("right");
        e.preventDefault();
    }
    if(keyID === 40) { //down arrow
        alert("down");
        e.preventDefault();
    }
}
```

</script>

4. Clique na tecla Cima, Baixo, Esquerda e Direita para mover o avião na tela.

```
if(keyID === 37) { //left arrow
  bluePlane.move(context,-5, 0)
  e.preventDefault();
}
if(keyID === 38) { //up arrow
  bluePlane.move(context,0, -5)
  e.preventDefault();
}
if(keyID === 39) { //right arrow
  bluePlane.move(context,5, 0)
  e.preventDefault();
}
if(keyID === 40) { //down arrow
  bluePlane.move(context,0, 5)
  e.preventDefault();
}
```



```
<style>
  #myCanvas{
    border:1px solid #c3c3c3;
    position: absolute;
  }
</style>

<canvas id="myCanvas" width="300" height="300"></canvas>

<script type="text/javascript">
  var canvas = document.getElementById("myCanvas");
  var context = canvas.getContext("2d");
  var canvasWidth = 300;
  var canvasHeight = 300;

  class Airplane{

    constructor(img, x, y){
      this.x = x;
      this.y = y;
      this.img = img;
    }

    draw(context){
      context.drawImage(this.img,this.x,this.y);
    }

    move(context,distanceX,distanceY){
      this.x = this.x + distanceX;
      this.y = this.y + distanceY;
      context.drawImage(this.img,this.x,this.y);
    }
  }

  ////////// test //////////
```

```
var bluePlaneImage = new Image();
bluePlaneImage.src = "./images/blue_plane.png";
var bluePlane = new Airplane(bluePlaneImage,100,220);
bluePlaneImage.onload = function(){
    bluePlane.draw(context);
};
```

```
canvas.addEventListener('keydown', doKeyDown,true);
canvas.focus();
window.addEventListener('keydown', doKeyDown, true);
```

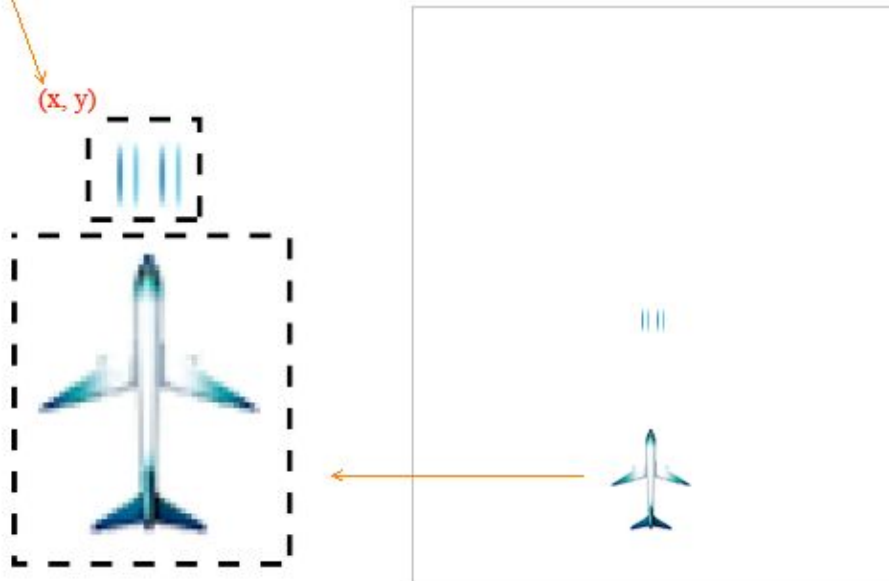
```
function doKeyDown(e) {
    context.clearRect(0,0,canvasWidth,canvasHeight);
    var keyID = e.keyCode ? e.keyCode :e.which;
    if(keyID === 37) { //left arrow
        bluePlane.move(context,-5, 0)
        e.preventDefault();
    }
    if(keyID === 38) { //up arrow
        bluePlane.move(context,0, -5)
        e.preventDefault();
    }
    if(keyID === 39) { //right arrow
        bluePlane.move(context,5, 0)
        e.preventDefault();
    }
    if(keyID === 40) { //down arrow
        bluePlane.move(context,0, 5)
        e.preventDefault();
    }
}
```

```
</script>
```

5. O avião dispara uma bala azul.

5.1 As balas disparadas do centro superior do avião

```
var x = bluePlane.getX()+bluePlane.getWidth()/2-bullet.getWidth()/2;  
var y = bluePlane.getY() - bullet.getHeight();  
bullet.setX(x);  
bullet.setY(y);  
bullet.setVisible(true);
```



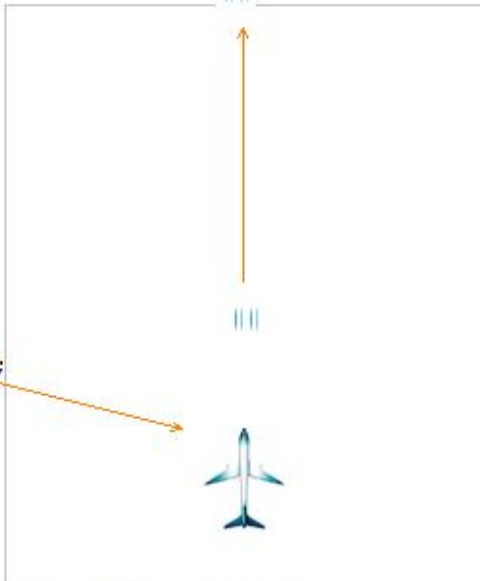
5.2 Inicie um **setInterval("reDraw()", 200)** para manter o marcador movendo-se para cima, quando o marcador sair da tela, o marcador precisa ser destruído **set visible = false**, o marcador pode ser reemitido.

```
class Bullet extends Sprite{
    move(context,distanceX,distanceY){
        this.x = this.x + distanceX;
        this.y = this.y + distanceY;
        if(this.y + this.height <=0){
            this.setVisible(false)
        }
    }
}

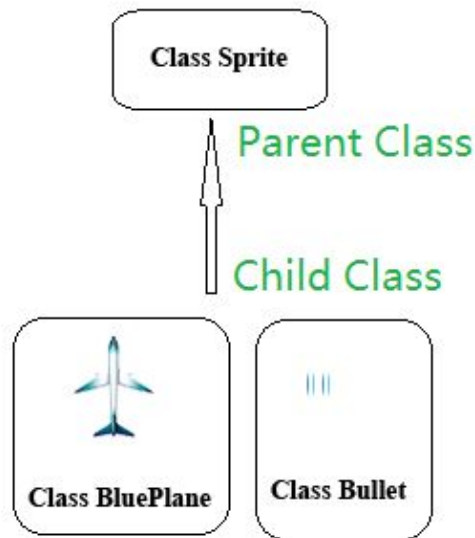
function bulletMove(){
    if(bullet.isVisible()){
        bullet.move(context,0,-5);
    }
    bullet.draw(context);
}

function reDraw(){
    context.clearRect(0,0,canvasWidth,canvasHeight);
    bulletMove();
    bluePlane.draw(context);
}

window.onload=function(){
    //Run reDraw() every 200 milliseconds
    var time = setInterval("reDraw()",200);
}
```



5.3 Avião e marcador herdam de Sprite da reutilização.



```
<style>
  #myCanvas{
    border:1px solid #c3c3c3;
    position: absolute;
  }
</style>
<span id="dd"></span>
<br>
<canvas id="myCanvas" width="300" height="300"></canvas>

<script type="text/javascript">
  var canvas = document.getElementById("myCanvas");
  var context = canvas.getContext("2d");
  var canvasWidth = 300;
  var canvasHeight = 300;

  class Sprite{
    constructor(img, x, y){
      this.x = x;
      this.y = y;
      this.img = img;
```

```
}

draw(context){
    context.drawImage(this.img,this.x,this.y);
}

move(context,distanceX,distanceY){
    this.x = this.x + distanceX;
    this.y = this.y + distanceY;
}

isVisible(){
    return this.visible;
}

setVisible(visible){
    this.visible = visible;
}

getX(){
    return this.x;
}

setX(x){
    this.x=x;
}

getY(){
    return this.y;
}

setY(y){
    this.y=y;
}

getHeight(){
    return this.height;
}
```



```

    setHeight(height){
        this.height = height;
    }

    getWidth(){
        return this.width;
    }

    setWidth(width){
        this.width=width;
    }
}

class Airplane extends Sprite{

}

class Bullet extends Sprite{

    move(context,distanceX,distanceY){
        this.x = this.x + distanceX;
        this.y = this.y + distanceY;
        if(this.y + this.height <=0){
            this.setVisible(false)
        }
    }
}

////////// test //////////

var bluePlaneImage = new Image();
bluePlaneImage.src = "./images/blue_plane.png";
var bluePlane = new Airplane(bluePlaneImage,100,220);
bluePlaneImage.onload = function() {
    bluePlane.setWidth(bluePlaneImage.width);
    bluePlane.setHeight(bluePlaneImage.height);
}

```

```
    bluePlane.draw(context);  
};  
  
var bulletImage = new Image();  
bulletImage.src = "./images/blue_bullet.png";  
var bullet = new Bullet(bulletImage,-100,-100);  
bulletImage.onload = function(){  
    bullet.setWidth(bulletImage.width);  
    bullet.setHeight(bulletImage.height);  
    bullet.setVisible(false);  
    bullet.draw(context);  
};
```

```
canvas.addEventListener('keydown', doKeyDown,true);  
canvas.focus();  
window.addEventListener('keydown', doKeyDown, true);
```

```
function doKeyDown(e) {  
    var keyID = e.keyCode ? e.keyCode :e.which;  
    if(keyID === 37) { //left arrow  
        bluePlane.move(context,-5, 0)  
        e.preventDefault();  
    }  
    if(keyID === 38) { //up arrow  
        bluePlane.move(context,0, -5)  
        e.preventDefault();  
    }  
    if(keyID === 39) { //right arrow  
        bluePlane.move(context,5, 0)  
        e.preventDefault();  
    }  
    if(keyID === 40) { //down arrow  
        bluePlane.move(context,0, 5)  
        e.preventDefault();  
    }  
}
```

```

    }
    if(keyID === 13) { //enter arrow
        if(!bullet.isVisible()){
            var x = bluePlane.getX()+bluePlane.getWidth()/2-
bullet.getWidth()/2;
            var y = bluePlane.getY() - bullet.getHeight();
            bullet.setX(x);
            bullet.setY(y);
            bullet.setVisible(true);
        }
        e.preventDefault();
    }
    reDraw();
}

function bulletMove(){
    if(bullet.isVisible()){
        bullet.move(context,0,-5);
    }
    bullet.draw(context);
}

function reDraw(){
    context.clearRect(0,0,canvasWidth,canvasHeight);
    bulletMove();
    bluePlane.draw(context);
}

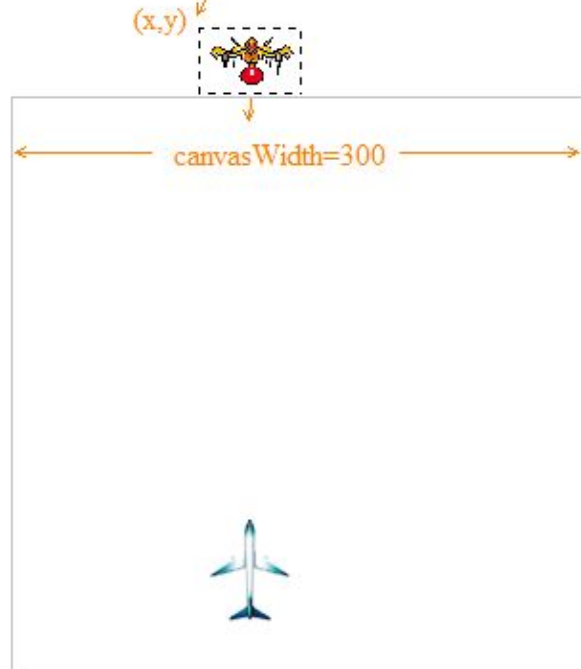
window.onload=function(){
    // Execute reDraw() a cada 200 milissegundos
    var time = setInterval("reDraw()",200);
}
</script>

```

6. O avião inimigo continua se movendo para baixo.

6.1 O avião inimigo começa aleatoriamente do topo da tela e desce

```
function enemyPlaneMove() {  
    if(!enemyPlane.isVisible()) {  
        var x = Math.floor(Math.random()*canvasWidth)+1;  
        var y = -enemyPlane.getHeight();  
        enemyPlane.setX(x);  
        enemyPlane.setY(y);  
        enemyPlane.setVisible(true);  
        enemyPlane.draw(context);  
    }  
    if(enemyPlane.isVisible()) {  
        enemyPlane.move(context,0,3);  
    }  
    enemyPlane.draw(context);  
}
```

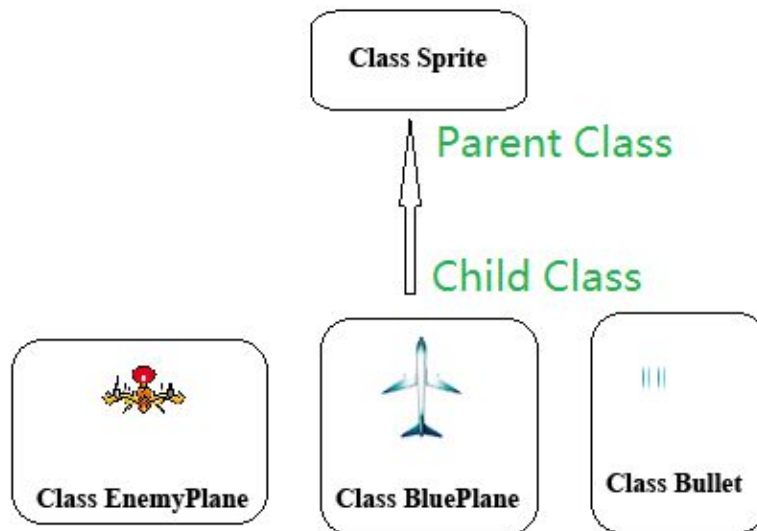


6.2 Mantenha o avião inimigo se movendo para baixo, quando o avião inimigo se mover para fora da tela, o avião inimigo precisa ser destruído set **visible = false**, o avião inimigo pode ser reemitido.



```
class EnemyPlane extends Sprite{  
    move(context,distanceX,distanceY){  
        this.x = this.x + distanceX;  
        this.y = this.y + distanceY;  
        if(this.y >= canvasHeight){  
            this.setVisible(false);  
        }  
    }  
}
```

6.3 Avião, bala e avião inimigo são herdados de Sprite da reutilização.



```
<style>
  #myCanvas{
    border:1px solid #c3c3c3;
    position: absolute;
  }
</style>
<span id="dd"></span>
<br>
<canvas id="myCanvas" width="300" height="300"></canvas>

<script type="text/javascript">
  var canvas = document.getElementById("myCanvas");
  var context = canvas.getContext("2d");
  var canvasWidth = 300;
  var canvasHeight = 300;
```

```
class Sprite{
    constructor(img, x, y){
        this.x = x;
        this.y = y;
        this.img = img;
    }

    draw(context){
        context.drawImage(this.img,this.x,this.y);
    }

    move(context,distanceX,distanceY){
        this.x = this.x + distanceX;
        this.y = this.y + distanceY;
    }

    isVisible(){
        return this.visible;
    }

    setVisible(visible){
        this.visible = visible;
    }

    getX(){
        return this.x;
    }

    setX(x){
        this.x=x;
    }
}
```

```

    getY(){
        return this.y;
    }
    setY(y){
        this.y=y;
    }

    getHeight(){
        return this.height;
    }

    setHeight(height){
        this.height = height;
    }

    getWidth(){
        return this.width;
    }

    setWidth(width){
        this.width=width;
    }
}

class Airplane extends Sprite{

class Bullet extends Sprite{

    move(context,distanceX,distanceY){
        this.x = this.x + distanceX;
        this.y = this.y + distanceY;
        if((this.y + this.height <=0){
            this.setVisible(false)
        }
    }
}

```



```

    }
}

class EnemyPlane extends Sprite{

    move(context,distanceX,distanceY){
        this.x = this.x + distanceX;
        this.y = this.y + distanceY;
        if(this.y >= canvasHeight){
            this.setVisible(false);
        }
    }
}

```

```

////////// test //////////

```

```

var bluePlaneImage = new Image();
bluePlaneImage.src = "./images/blue_plane.png";
var bluePlane = new Airplane(bluePlaneImage,100,220);
bluePlaneImage.onload = function(){
    bluePlane.setWidth(bluePlaneImage.width);
    bluePlane.setHeight(bluePlaneImage.height);
    bluePlane.draw(context);
};

var bulletImage = new Image();
bulletImage.src = "./images/blue_bullet.png";
var bullet = new Bullet(bulletImage,-100,-100);
bulletImage.onload = function(){
    bullet.setWidth(bulletImage.width);
    bullet.setHeight(bulletImage.height);
    bullet.setVisible(false);
}

```

```
bullet.draw(context);
};

var enemyImage = new Image();
enemyImage.src = "./images/enemy.png";
var enemyPlane = new EnemyPlane(enemyImage,-100,-100);
enemyImage.onload = function(){
    enemyPlane.setWidth(enemyImage.width);
    enemyPlane.setHeight(enemyImage.height);
    enemyPlane.setVisible(false);
    enemyPlane.draw(context);
};

canvas.addEventListener('keydown', doKeyDown,true);
canvas.focus();
window.addEventListener('keydown', doKeyDown, true);
```

```
function doKeyDown(e) {
    var keyID = e.keyCode ? e.keyCode :e.which;
    if(keyID === 37) { //left arrow
        bluePlane.move(context,-5, 0)
        e.preventDefault();
    }
    if(keyID === 38) { //up arrow
        bluePlane.move(context,0, -5)
        e.preventDefault();
    }
    if(keyID === 39) { //right arrow
```

```

        bluePlane.move(context,5, 0)
        e.preventDefault();
    }
    if(keyID === 40) { //down arrow
        bluePlane.move(context,0, 5)
        e.preventDefault();
    }
    if(keyID === 13) { //enter arrow
        if(!bullet.isVisible()){
            var x = bluePlane.getX()+bluePlane.getWidth()/2-
bullet.getWidth()/2;
            var y = bluePlane.getY() - bullet.getHeight();
            bullet.setX(x);
            bullet.setY(y);
            bullet.setVisible(true);
        }
        e.preventDefault();
    }
    bluePlane.draw(context);
}

function bulletMove(){
    if(bullet.isVisible()){
        bullet.move(context,0,-5);
    }
    bullet.draw(context);
}

```

```

function enemyPlaneMove(){
    if(!enemyPlane.isVisible()){
        var x = Math.floor(Math.random()*canvasWidth)+1;
    }
}

```

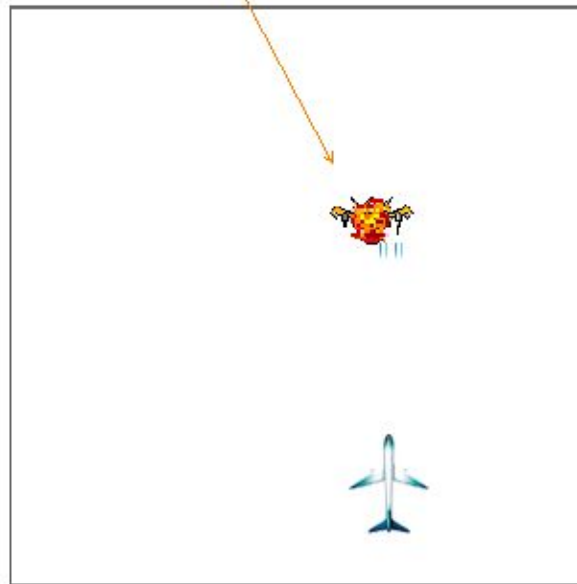
```
    var y = -enemyPlane.getHeight();
    enemyPlane.setX(x);
    enemyPlane.setY(y);
    enemyPlane.setVisible(true);
    enemyPlane.draw(context);
}
if(enemyPlane.isVisible()){
    enemyPlane.move(context,0,3);
}
enemyPlane.draw(context);
}

function reDraw(){
    context.clearRect(0,0,canvasWidth,canvasHeight);
    bulletMove();
    enemyPlaneMove();
    bluePlane.draw(context);
}

window.onload=function(){
    // Execute reDraw() a cada 200 milissegundos
    var time = setInterval("reDraw()",200);
}
</script>
```

7. A bala colidiu com o avião inimigo e explodiu.

```
function collide(){  
    if(bullet.isVisible() && enemyPlane.isVisible()){  
        if(bullet.collideWith(enemyPlane)){  
            bullet.setX(-100);  
            bullet.setY(-100);  
            bullet.setVisible(false);  
            enemyPlane.setVisible(false);  
        }  
    }  
}
```

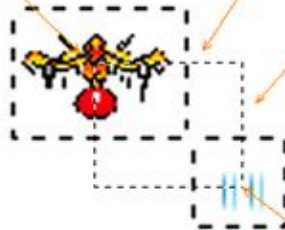


7.1 bala colidiu com o avião inimigo

O valor absoluto da diferença entre as coordenadas centrais das duas imagens é menor que a soma da metade de sua largura e altura, pensamos que elas têm uma interseção e colidiram

```
collideWith(sprite) {  
    var centerX = this.x + this.getWidth()/2;  
    var centerY = this.y + this.getHeight()/2;  
    var spriteCenterX = sprite.getX() + sprite.getWidth()/2;  
    var spriteCenterY = sprite.getY() + sprite.getHeight()/2;  
  
    if(Math.abs(centerX - spriteCenterX) < (this.width/2 + sprite.getWidth()/2)  
        &&  
        Math.abs(centerY - spriteCenterY) < (this.height/2 + sprite.getHeight()/2))  
        return true  
    else  
        return false  
}
```

(centerX, centerY)



(spriteCenterX, spriteCenterY)

```
<style>
  #myCanvas{
    border:1px solid #c3c3c3;
    position: absolute;
  }
</style>
<span id="dd"></span>
<br>
<canvas id="myCanvas" width="300" height="300"></canvas>

<script type="text/javascript">
  var canvas = document.getElementById("myCanvas");
  var context = canvas.getContext("2d");
  var canvasWidth = 300;
  var canvasHeight = 300;

  class Sprite{

    constructor(img, x, y){
      this.x = x;
      this.y = y;
      this.img = img;
    }

    draw(context){
      context.drawImage(this.img,this.x,this.y);
    }

    move(context,distanceX,distanceY){
      this.x = this.x + distanceX;
      this.y = this.y + distanceY;
    }
  }
</script>
```

```

collideWith(sprite){
    var centerX = this.x + this.getWidth()/2;
    var centerY = this.y + this.getHeight()/2;
    var spriteCenterX = sprite.getX() + sprite.getWidth()/2;
    var spriteCenterY = sprite.getY() + sprite.getHeight()/2;

    if(Math.abs(centerX - spriteCenterX) < (this.width/2 +
sprite.getWidth()/2)
        &&
        Math.abs(centerY - spriteCenterY) < (this.height/2 +
sprite.getHeight()/2))
        return true
    else
        return false
    }

isVisible(){
    return this.visible;
}

setVisible(visible){
    this.visible = visible;
}

getX(){
    return this.x;
}

setX(x){

```



```
        this.x=x;
    }

    getY(){
        return this.y;
    }
    setY(y){
        this.y=y;
    }

    getHeight(){
        return this.height;
    }


    setHeight(height){
        this.height = height;
    }

    getWidth(){
        return this.width;
    }

    setWidth(width){
        this.width=width;
    }
}

class Airplane extends Sprite{

}

class Bullet extends Sprite{

    move(context,distanceX,distanceY){
```

```

        this.x = this.x + distanceX;
        this.y = this.y + distanceY;
        if((this.y + this.height <=0){
            this.setVisible(false)
        }
    }
}

```

```

class EnemyPlane extends Sprite{

```

```

    move(context,distanceX,distanceY){
        this.x = this.x + distanceX;
        this.y = this.y + distanceY;
        if((this.y >= canvasHeight){
            this.setVisible(false);
        }
    }
}

```

```

////////// test //////////

```

```

var bluePlaneImage = new Image();
bluePlaneImage.src = "./images/blue_plane.png";
var bluePlane = new Airplane(bluePlaneImage,100,220);
bluePlaneImage.onload = function(){
    bluePlane.setWidth(bluePlaneImage.width);
    bluePlane.setHeight(bluePlaneImage.height);
    bluePlane.draw(context);
};

```

```

var bulletImage = new Image();
bulletImage.src = "./images/blue_bullet.png";

```

```
var bullet = new Bullet(bulletImage,-100,-100);
bulletImage.onload = function(){
    bullet.setWidth(bulletImage.width);
    bullet.setHeight(bulletImage.height);
    bullet.setVisible(false);
    bullet.draw(context);
};

var enemyImage = new Image();
enemyImage.src = "./images/enemy.png";
var enemyPlane = new EnemyPlane(enemyImage,-100,-100);
enemyImage.onload = function(){
    enemyPlane.setWidth(enemyImage.width);
    enemyPlane.setHeight(enemyImage.height);
    enemyPlane.setVisible(false);
    enemyPlane.draw(context);
};

canvas.addEventListener('keydown', doKeyDown,true);
canvas.focus();
window.addEventListener('keydown', doKeyDown, true);
```

```
function doKeyDown(e) {
    var keyID = e.keyCode ? e.keyCode : e.which;
    if(keyID === 37) { //left arrow
        bluePlane.move(context,-5, 0)
        e.preventDefault();
    }
}
```

```

    if(keyID === 38) { //up arrow
        bluePlane.move(context,0, -5)
        e.preventDefault();
    }
    if(keyID === 39) { //right arrow
        bluePlane.move(context,5, 0)
        e.preventDefault();
    }
    if(keyID === 40) { //down arrow
        bluePlane.move(context,0, 5)
        e.preventDefault();
    }
    if(keyID === 13) { //enter arrow
        if(!bullet.isVisible()){
            var x = bluePlane.getX()+bluePlane.getWidth()/2-
bullet.getWidth()/2;
            var y = bluePlane.getY() - bullet.getHeight();
            bullet.setX(x);
            bullet.setY(y);
            bullet.setVisible(true);
        }
        e.preventDefault();
    }
    bluePlane.draw(context);
}

function bulletMove(){
    if(bullet.isVisible()){
        bullet.move(context,0,-5);
    }
    bullet.draw(context);
}

```

```

function enemyPlaneMove(){
  if(!enemyPlane.isVisible()){
    var x = Math.floor(Math.random()*canvasWidth)+1;
    var y = -enemyPlane.getHeight();
    enemyPlane.setX(x);
    enemyPlane.setY(y);
    enemyPlane.setVisible(true);
    enemyPlane.draw(context);
  }
  if(enemyPlane.isVisible()){
    enemyPlane.move(context,0,3);
  }
  enemyPlane.draw(context);
}

```

```

function collide(){
  if(bullet.isVisible() && enemyPlane.isVisible()){
    if(bullet.collideWith(enemyPlane)){
      bullet.setX(-100);
      bullet.setY(-100);
      bullet.setVisible(false);
      enemyPlane.setVisible(false);
    }
  }
}

```

```

function reDraw(){
  context.clearRect(0,0,canvasWidth,canvasHeight);
  bulletMove();
  enemyPlaneMove();
  bluePlane.draw(context);
  collide();
}

```

```

window.onload=function(){

```

```
//Execute redraw() a cada 200 milissegundos  
var time = setInterval("redraw()",200);  
}  
</script>
```

Obrigado por aprender

<https://www.amazon.com.br/dp/B08QTZ3N9M>



Se você gostou deste livro e encontrou algum benefício em ler isso, gostaria de ouvir sua opinião e espero que você possa levar algum tempo para publicar uma resenha na Amazon. Seus comentários e suporte nos ajudará a melhorar bastante no futuro e tornar este livro ainda melhor.

Você pode seguir este link agora.

<http://www.amazon.com.br/review/create-review?&asin=B08P5R7TKK>

Revisões de diferentes países só precisam modificar o nome de domínio amazon no link:

www.amazon.co.uk

www.amazon.de

www.amazon.fr

www.amazon.es

www.amazon.it

www.amazon.ca

www.amazon.nl

www.amazon.in

www.amazon.co.jp

www.amazon.com.br

www.amazon.com.mx

www.amazon.com.au

Desejo a todos o melhor em seu futuro sucesso!