**Lógica de programação II**

- desenhando gráficos com canvas

- relembrando:

- tudo que está dentro das tags <script></script>

- pertence ao mundo JS

- tudo que está fora, seja antes ou depois pertence ao HTML

- a função document permite dentro do mundo JS retornar algo no mundo HTML

- document.write ( );

- escreve algo do mundo JS no mundo HTML

- <canvas></canvas>

- é uma tag do HTML que permite que desenhemos algo

- por padrão ela é branca

- para definir (delimitar) a área que o canvas vai ocupar dentro do navegador utilizamos dois atributos

width (largura)

height (altura)

- document.querySelector ( )

- permite utilizar algo do HTML no JS

- podemos utilizar no JS tanto aspas simples (apóstrofo), quanto aspas duplas (aspas)

***var tela = document.querySelector('canvas');***

***var pincel = tela.getContext('2d');***

- a função querySelector pega a tag ‘canvas’ e passa como valor na variável tela

- criamos a variável pincel que recebe a tela como valor

- getContext se refere ao contexto no qual escreveremos na tela

***pincel.fillRect(0, 0, 800, 600);***

- pincel vai preencher um retângulo

fill (preencher)

rect (retângulo)

- precisamos informar ao JS onde inicia o preenchimento do retângulo

0, 0 (vai iniciar no x e y na posição zero)

- eixo x (horizontal)

- eixo y (vertical)

- a partir do 0 indicamos que ele vai preencher 800 de largura e 600 de altura

***pincel.fillStyle = 'grey';***

- indicamos a cor do preenchimento (no caso cinza)

- o fillStyle vem antes do fillRect

- diferente de fillRect ( ), querySelector ( ), getContext ( )

- fillStyle não é uma função e sim uma propriedade (equivalente a uma propriedade, pois recebe um valor)

- o fillRect() precisa de algumas informações para conseguir desenhar o retângulo, que estarão na ordem: posição inicial X do “pincel”, posição inicial Y do “pincel”, posição final X do “pincel”, posição final Y do “pincel”.

- as informações sempre têm que aparecer na ordem correta para que o JavaScript interprete corretamente onde o desenho começa e onde termina.

- para realizar um comentário no JS **/\* texto a ser comentado \*/**

**<canvas width="900" height="600"></canvas>**

**<script>**

**var tela = document.querySelector('canvas');**

**var pincel = tela.getContext('2d');**

**pincel.fillStyle = 'grey';**

**pincel.fillRect(0, 0, 900, 600);**

**pincel.fillStyle = 'purple';**

**pincel.fillRect (0, 0, 300, 600);**

**pincel.fillStyle = 'violet';**

**pincel.fillRect (300, 0, 300, 600);**

**pincel.fillStyle = 'pink';**

**pincel.fillRect (600, 0, 300, 600);**

**</script>**

- abaixo o resultado do código

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente

pincel.beginPath ()

- informamos a pincel o incio (begin) do caminho (path)

pincel.moveTo(450, 300);

- o pincel ira mover se para o eixo x (450) e y (300)

pincel.lineTo(300, 600);

- o pincel fará uma linha x (300) e y (600)

pincel.lineTo(600, 600);

- o pincel fará uma linha x (600) e y (600)

pincel.fill();

- preenche todo o desenho com a cor definida anteriormente

***<canvas width="900" height="600"></canvas>***

***<script>***

***var tela = document.querySelector('canvas');***

***var pincel = tela.getContext('2d');***

***pincel.fillStyle = 'grey';***

***pincel.fillRect(0, 0, 900, 600);***

***pincel.fillStyle = 'purple';***

***pincel.fillRect (0, 0, 300, 600);***

***pincel.fillStyle = 'violet';***

***pincel.fillRect (300, 0, 300, 600);***

***pincel.fillStyle = 'pink';***

***pincel.fillRect (600, 0, 300, 600);***

***pincel.fillStyle = 'blue';***

***pincel.beginPath ();***

***pincel.moveTo(450, 300);***

***pincel.lineTo(300, 600);***

***pincel.lineTo(600, 600);***

***pincel.fill();***

***</script>***

- abaixo o resultado do código

Forma

Descrição gerada automaticamente

- beginPath ( );

- inicia um caminho

- fillStyle = ‘ cor’;

- define a cor

- pincel.arc

- desenha um circulo (um arco)

***pincel.fillStyle = 'darkblue';***

***pincel.beginPath();***

***pincel.arc (450, 300, 60, 0, 2 \* 3.14);***

***pincel.fill();***

Uma imagem contendo Forma

Descrição gerada automaticamente

- para que pudéssemos construir esta imagem, precisamos conhecer a função getContext(), fillRect(), lineTo(), beginPath(), a propriedade fillStyle, a esse conjunto de propriedades damos o nome de APIs, ou bibliotecas

- estamos manipulando a API específica para a criação de gráficos, nesse caso a canvas

- chamamos as instruções que as bibliotecas (APIs) deixam disponíveis para usarmos em nossos programas, de métodos

- os métodos moveTo() e lineTo() vistos na aula podem ser usados não apenas para desenhar triângulos, mas várias outras formas poligonais

- enquanto o programa receber instruções sucessivas para traçar linhas (com lineTo()) e forem dadas as coordenadas corretas em cada uma, é possível desenhar e preencher polígonos com a quantidade de lados que quisermos.

- existem diversos métodos diferentes para desenhar uma forma simples, como um retângulo, podemos usar fillRect(). Porém para formas com mais possibilidades de desenho, vamos precisar utilizar outros métodos.

- um triângulo pode ter diversas aberturas de ângulos, por exemplo, então devemos usar um método que permita traçar o ponto inicial e final de cada um dos lados.

***<canvas width="600" height="400"></canvas>***

***<script>***

***var tela = document.querySelector('canvas');***

***var pincel = tela.getContext('2d');***

***// cabeça***

***pincel.fillStyle = 'darkgreen';***

***pincel.fillRect(200,50,350,300);***

***// olhos***

***pincel.fillStyle = 'black';***

***pincel.fillRect(220, 110, 90, 90);***

***pincel.fillRect(440, 110, 90, 90);***

***// nariz***

***pincel.fillRect(340, 200, 70, 100);***

***// boca ou barba***

***pincel.fillRect(300, 240, 40, 110);***

***pincel.fillRect(410, 240, 40, 110);***

***</script>***

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente com confiança média

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

- para iniciar o path precisamos mover o pincel para a cordenada pincel.beginPath ( );

- após as cordenadas preenchemos com pincel.fill ( );

***<canvas width="600" height="400"></canvas>***

***<script>***

***var tela = document.querySelector('canvas');***

***var pincel = tela.getContext('2d');***

***pincel.fillStyle = 'black';***

***pincel.beginPath();***

***pincel.moveTo(50, 50);***

***pincel.lineTo(50, 400);***

***pincel.lineTo(400,400);***

***pincel.fill();***

***pincel.fillStyle = 'white';***

***pincel.beginPath();***

***pincel.moveTo(100, 175);***

***pincel.lineTo(100, 350);***

***pincel.lineTo(275,350);***

***pincel.fill();***

***</script>***

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente com confiança média

- bandeira do brasil

***<canvas width="600" height="400"></canvas>***

***<script>***

***var tela = document.querySelector('canvas');***

***var pincel = tela.getContext('2d');***

***pincel.fillStyle = 'green';***

***pincel.fillRect(0, 0, 600, 400);***

***pincel.fillStyle='yellow';***

***pincel.beginPath();***

***pincel.moveTo(300, 50);***

***pincel.lineTo(50, 200);***

***pincel.lineTo(550, 200);***

***pincel.fill();***

***pincel.fillStyle='yellow';***

***pincel.beginPath();***

***pincel.moveTo(300, 350);***

***pincel.lineTo(50, 200);***

***pincel.lineTo(550, 200);***

***pincel.fill();***

***pincel.fillStyle = 'darkblue';***

***pincel.beginPath();***

***pincel.arc(300, 200, 100, 0, 2\*3.14);***

***pincel.fill();***

***</script>***

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

02 – Extraindo funções

- document.querySelector( );

- transporta uma tag HTML para o mundo JS

- aspas simples (apostrofo) só podem ser utilizados dentro do JS, no HTML é aspas (“)

- fillStyle = ‘nome da cor’

- recebe uma cor como atributo

- fillRect (0, 0, 100, 100)

- desenha uma forma de 4 lados (quadrado, retângulo)

- recebe o eixo x e y como posições iniciais e o tamanho da figura a ser desenhada

- os números 100 do exemplo representam o tamanho da figura em pixels

- strokeStyle = ‘nome da cor’

- desenha uma borda com a cor escolhida

- quando não definimos a cor da borda o padrão é preto

- strokerect (0, 0, 100 , 100);

- diz que a borda deve seguir determinadas coordenadas (podemos utilizar as coordenadas da figura geométrica por exemplo)

- podemos colocar toda a instrução dentro de uma função

<canvas width="600" height="400"></canvas>

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

***function desenhaQuadrado(x, y, cor) {***

***var tela = document.querySelector('canvas');***

***var pincel = tela.getContext('2d');***

***pincel.fillStyle = cor;***

***pincel.fillRect(x, y, 50, 50);***

***pincel.fillStroke = 'black';***

***pincel.strokeRect(x, y, 50, 50);***

***}***

***desenhaQuadrado(0, 0, 'green');***

***desenhaQuadrado(50, 0, 'yellow');***

***desenhaQuadrado(100, 0, 'blue');***

***desenhaQuadrado(0, 50, 'pink');***

***desenhaQuadrado(50, 50, 'black');***

***desenhaQuadrado (100,50, 'grey');***

***desenhaQuadrado(0, 100, 'purple');***

***desenhaQuadrado(50, 100, 'red');***

***desenhaQuadrado(100, 100, 'violet');***

***</script>***

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

- as funções podem nos ajudar pois:

- Automatizam tarefas com código mais genérico

-Podemos criar uma função genérica que receba os dados necessários por meio de parâmetros, e deixar que um loop chame a mesma função várias vezes, com um parâmetro diferente a cada

vez.

- repetição do codigo

- Em vez de escrever uma linha para cada forma que queremos criar, passando manualmente as informações de cada uma, podemos escrever uma função para, por exemplo, criar um quadrado colorido de qualquer tamanho e cor.

- Deixar o código organizado e separado.

- Utilizamos funções para deixar cada parte do código bem separada, assim sabemos exatamente o que está acontecendo em cada uma. Além do mais, é prática comum utilizarmos nomes bem específicos para cada função, assim quem ler o código não terá problemas em saber o que elas fazem.

- Os loops, ou laços de repetição, são uma estrutura básica em programação, que usamos para repetir tarefas - lembrando que repetir e automatizar tarefas são algumas das atividades principais que um programa deve fazer.

- Todo loop precisa de uma linha de código dentro dele, que serve para mudar a condição atual.

- Quando criamos um loop, passamos uma condição, por exemplo while (x < 600) ou “enquanto x for menor que 600”. Mas se não for passada nenhuma outra linha de código dentro do loop para que x em algum momento tenha esse valor, por exemplo x = x + 50, o código entra no chamado loop infinito.

- Loops executam enquanto a condição passada (normalmente entre parênteses()) for verdadeira.

- Quando o programa executa a linha de código onde está o loop, fica “preso” dentro do loop enquanto a condição for verdadeira. Por exemplo, while (x < 600).

- No caso abaixo:

***var x = 0;***

***while(x < 600) {***

***desenhaQuadrado(x, 0, 'green');***

***x = x + 50;***

***}***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***for(var x = 0; x < 600; x = x + 50) {***

***desenhaQuadrado(x, 0, 'green');***

***}***

- Os dois loops são escritos de formas diferentes, porém o resultado da execução é o mesmo.

- É normal existir várias formas de escrever códigos que façam a mesma tarefa, e este é um destes casos. Porém, embora o resultado final seja o mesmo, há uma diferença na sintaxe (ou seja, na forma de escrever) de cada um: enquanto o while precisa receber manualmente em alguma parte do código a linha que altera a condição verdadeira (no caso, x = x + 50;, pela sintaxe do for precisamos passar essa informação nos parênteses iniciais.

- Os loops podem ser utilizados em conjunto com funções para automatizar algumas tarefas de código

- É muito comum chamarmos funções para serem executadas dentro de loops, pois assim passamos para o programa a tarefa de controlar a quantidade de vezes que a função é executada, quando parar a execução e com quais informações ela vai trabalhar a cada vez - desde que não tenha erros no código que nós escrevemos

***var x = 0;***

***while(x < 600) {***

***desenhaQuadrado(x, 0, 'green');***

***x = x + 50;***

***}***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***for(var x = 0; x < 600; x = x + 50) {***

***desenhaQuadrado(x, 0, 'green');***

***}***

- O valor de x usado como condição para o loop (x = x + 50) é exatamente o mesmo passado como parâmetro para a função desenhaQuadrado(x, 0, ‘green’).

- Nos exemplos da aula, x é uma variável, ou seja, um espaço na memória onde podemos guardar valores e utilizá-los no código. Então, se a variável x guarda o valor 0 no início do código, em todos os lugares onde passarmos x no código, o programa vai automaticamente substituir pelo valor de 0. Essa lógica continua funcionando depois de modificarmos o valor de x, por exemplo com a instrução x = x + 50;.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Interface gráfica do usuário

Descrição gerada automaticamente com confiança média

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

- fillText ( )

- recebe um texto e as coordenadas que deve aparecer

- .font = ‘20px Georgia’

- tamanho e tipo da fonte

- .fillStyle = ‘black’;

- cor da fonte

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Gráfico

Descrição gerada automaticamente com confiança média

- através de gráficos podemos expressar visualmente dados ou valores numéricos

- auxiliam o entendimento entre valores

- no mundo dos gráficos os valores são chamados de série

- <canvas>

- desenha uma tela no HTML

- recebe width e height para delimitar altura e largura

- document.querySelector( );

- recebe uma tag do HTML dentro do JS

- getcontext

- define o elemento

;

- fillStyle = ‘cor’

- recebe a cor do elemento

- fillRect (x, y, largura, altura);

- recebe as coordenadas iniciais do objeto (x, y)

- bem como as dimensões em pixels (aqui representado por largura e altura)

- strokeStyle = ‘ cor’ ;

- recebe a cor da linha

- strokeRect (x, y, largura, altura);

- recebe as cordenadas iniciais do objeto (x, y)

- nesse caso da linha que sera exibida na tela

- bem com as dimensões em pixel dessa linha

- fillText ( )

- recebe um texto e as coordenadas que deve aparecer

- .font = ‘20px Georgia’

- tamanho e tipo da fonte

- .fillStyle = ‘black’;

- cor da fonte

***<canvas width="600" height="400"></canvas>***

***<script>***

***function desenhaRetangulo(x, y, largura, altura, cor) {***

***var tela = document.querySelector('canvas');***

***var pincel = tela.getContext('2d');***

***pincel.fillStyle=cor;***

***pincel.fillRect(x,y, largura, altura);***

***pincel.strokeStyle='black';***

***pincel.strokeRect(x,y, largura, altura);***

***}***

***function desenhaTexto(x , y, texto) {***

***var tela = document.querySelector('canvas');***

***var pincel = tela.getContext('2d');***

***pincel.font='15px Georgia';***

***pincel.fillStyle='black';***

***pincel.fillText(texto, x, y);***

***}***

***function desenhaBarra(x, y, serie, cores, texto) {***

***desenhaTexto(x, y - 10, texto);***

***var somaAltura = 0;***

***for (var i = 0; i < serie.length; i++) {***

***var altura = serie[i];***

***desenhaRetangulo(x, y + somaAltura, 50, altura, cores[i]);***

***somaAltura = somaAltura + altura;***

***}***

***}***

***var cores = ['blue','green','yellow', 'red'];***

***var serie2015 = [50,25,20,5];***

***var serie2016 = [65,20,13,2];***

***desenhaBarra(50, 50, serie2015, cores, '2015');***

***desenhaBarra(150, 50, serie2016, cores, '2016');***

***</script>***

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

- esquadro 2

- no exercício do esquadro

- o X do ponto A é também do ponto B

- o Y do ponto C é também do ponto B

- então B é a junção de X de A com Y de C

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo, Word

Descrição gerada automaticamente

03 – Interagindo com o usuário

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

- onclick

- recebe a função exibeAlerta quando clicamos

- a função exibeAlerta retorna um alert com a mensagem ‘Cliquei’

- console.log retorna as coordenadas no navegador de onde clicamos (X e Y) entre outros parâmetros

Texto

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo, Word

Descrição gerada automaticamente

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo, Word

Descrição gerada automaticamente

***<canvas width="800" height="600"></canvas>***

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

***var tela = document.querySelector ('canvas');***

***var pincel = tela.getContext ('2d');***

***pincel.fillStyle = 'grey';***

***pincel.fillRect (0, 0, 800, 600);***

***var cores = ['blue', 'red', 'green']***

***var corAtual = 0;***

***function desenhaCirculo(evento) {***

***var x = evento.pageX - tela.offsetLeft;***

***var y = evento.pageY - tela.offsetTop;***

***pincel.fillStyle = cores[corAtual];***

***pincel.beginPath();***

***pincel.arc(x, y, 10, 0, 2 \* 3.14);***

***pincel.fill();***

***console.log(X + ', ' + y);***

***}***

***tela.onclick = desenhaCirculo;***

***function mudaCor () {***

***corAtual++;***

***if(corAtual >= cores.length) {***

***corAtual = 0;***

***}***

***return false;***

***}***

***tela.oncontextmenu = mudaCor;***

***</script>***

***<canvas width="600" height="400"></canvas>***

***<script>***

***var tela = document.querySelector('canvas');***

***var pincel = tela.getContext('2d');***

***pincel.fillStyle = 'grey';***

***pincel.fillRect(0, 0, 600, 400);***

***var cores = ['blue', 'red', 'green']***

***var corAtual = 0;***

***function desenhaCirculo(evento) {***

***var x = evento.pageX - tela.offsetLeft;***

***var y = evento.pageY - tela.offsetTop;***

***var raio = 10;***

***console.log(x + ',' + y);***

***if (evento.shiftKey) {***

***raio = raio + 20;***

***}***

***pincel.fillStyle = cores[corAtual];***

***pincel.beginPath();***

***pincel.arc(x, y, raio, 0, 2 \* 3.14);***

***pincel.fill();***

***}***

***tela.onclick = desenhaCirculo;***

***function mudaCor () {***

***corAtual++;***

***if(corAtual >= cores.length) {***

***corAtual = 0;***

***}***

***return false;***

***}***

***tela.oncontextmenu = mudaCor;***

***</script>***

***<canvas width="800" height="600"></canvas>***

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

***var tela = document.querySelector('canvas');***

***var pincel = tela.getContext('2d');***

***pincel.fillStyle = 'grey';***

***pincel.fillRect (0, 0, 800, 600);***

***var cores = ['blue', 'red', 'green']***

***var corAtual = 0;***

***var raio = 10;***

***function desenhaCirculo(evento) {***

***var x = evento.pageX - tela.offsetLeft;***

***var y = evento.pageY - tela.offsetTop;***

***console.log(x + ',' + y);***

***if (evento.shiftKey) {***

***raio = raio + 10;***

***}***

***pincel.fillStyle = cores[corAtual];***

***pincel.beginPath();***

***pincel.arc(x, y, raio, 0, 2 \* 3.14);***

***pincel.fill();***

***}***

***tela.onclick = desenhaCirculo;***

***function mudaCor () {***

***corAtual++;***

***if(corAtual >= cores.length) {***

***corAtual = 0;***

***}***

***return false;***

***}***

***tela.oncontextmenu = mudaCor;***

***</script>***

***<canvas width="800" height="600"></canvas>***

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

***var tela = document.querySelector('canvas');***

***var pincel = tela.getContext('2d');***

***pincel.fillStyle = 'grey';***

***pincel.fillRect(0, 0, 800, 600);***

***var cores = ['blue', 'red', 'green', 'yellow', 'purple', 'darkgreen', 'darkblue', 'pink', 'lightgreen', 'lightblue', 'orange'];***

***var corAtual = 0;***

***var raio = 10;***

***function desenhaCirculo(evento) {***

***var x = evento.pageX - tela.offsetLeft;***

***var y = evento.pageY - tela.offsetTop;***

***console.log(x + ',' +y);***

***if (evento.shiftKey && evento.altKey) {***

***alert('Por favor, aperte apenas uma tecla por vez.');***

***} else if (evento.shiftKey && raio + 10 <= 80) {***

***raio = raio + 10;***

***} else if (evento.altKey && raio - 5 >= 10) {***

***raio = raio - 5;***

***}***

***pincel.fillStyle = cores[corAtual];***

***pincel.beginPath();***

***pincel.arc(x, y, raio, 0, 2 \* 3.14);***

***pincel.fill();***

***}***

***tela.onclick = desenhaCirculo;***

***function mudaCor () {***

***corAtual++;***

***if(corAtual >= cores.length) {***

***corAtual = 0;***

***}***

***return false;***

***}***

***tela.oncontextmenu = mudaCor;***

***</script>***

Gráfico, Gráfico de bolhas

Descrição gerada automaticamente

- Desenhando com o mouse

- onmousemove

- permite capturar o movimento do mouse na tela e consequentemente suas coordenadas

- onmousedown

- evento que é disparado toda vez que o botão esquerdo do mouse é clicado (ou seja se mativermos o botão pressionado este evento ficará ativo pelo mesmo tempo)

- onmouseup

- evento que é disparado quando o botão esquerdo do mouse é solto

***<canvas width="800" height="600"></canvas>***

***<br>***

***Escolha uma cor <input type="color">)***

***<script>***

***var tela = document.querySelector('canvas');***

***var pincel = tela.getContext('2d');***

***var paleta = document.querySelector('input');***

***pincel.fillStyle = 'grey';***

***pincel.fillRect (0, 0, 800, 600);***

***var desenha = false;***

***function desenhaCirculo(evento) {***

***if (desenha){***

***var x = evento.pageX - tela.offsetLeft;***

***var y = evento.pageY - tela.offsetTop;***

***pincel.fillStyle = paleta.value;***

***pincel.beginPath ();***

***pincel.arc(x, y, 10, 0, 2 \* 3.14);***

***pincel.fill();***

***}***

***console.log (x + ',' + y);***

***}***

***tela.onmousemove = desenhaCirculo;***

***function habilitaDesenhar() {***

***desenha = true;***

***}***

***function desabilitaDesenhar() {***

***desenha = false;***

***}***

***tela.onmousedown = habilitaDesenhar;***

***tela.onmouseup = desabilitaDesenhar;***

***</script>***

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

***<canvas width="800" height="600"></canvas>***

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

***var tela = document.querySelector('canvas');***

***var pincel = tela.getContext('2d');***

***pincel.fillStyle = 'lightgray';***

***pincel.fillRect(0, 0, 800, 600);***

***function desenhaCirculo(x, y, raio) {***

***pincel.fillStyle = 'blue';***

***pincel.beginPath();***

***pincel.arc(x, y, raio, 0, 2 \* Math.PI);***

***pincel.fill();***

***}***

***desenhaCirculo(100, 100, 20);***

***</script>***

Interface gráfica do usuário, Aplicativo, Word

Descrição gerada automaticamente

- pincel.clearRect();

- indica ao JS que ele limpar o retângulo

- setInterval

- aceita como parâmetro a função que queremos chamar e depois a quantidade de tempo que queremos em milissegundos

***<canvas width="800" height="600"></canvas>***

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

***var tela = document.querySelector('canvas');***

***var pincel = tela.getContext('2d');***

***pincel.fillStyle = 'lightgray';***

***pincel.fillRect(0, 0, 800, 600);***

***function desenhaCirculo(x, y, raio) {***

***pincel.fillStyle = 'blue';***

***pincel.beginPath();***

***pincel.arc(x, y, raio, 0, 2 \* Math.PI);***

***pincel.fill();***

***}***

***function limpaTela() {***

***pincel.clearRect(0, 0, 800, 600);***

***}***

***var x = 20;***

***function atualizaTela() {***

***limpaTela();***

***desenhaCirculo(x, 20, 15);***

***x++;***

***}***

***setInterval(atualizaTela, 15);***

***</script>***

- uma variável no JS pode ter um escopo de função ou seja só estará disponível dentro da função ou quando declarada fora da função é um escopo global podendo ser acessada por qualquer outra função do nosso programa

- quando queremos passar uma função como parâmetro para outra função precisamos omitir os parênteses ( )

***<canvas width="800" height="600"></canvas>***

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

***var tela = document.querySelector('canvas');***

***var pincel = tela.getContext('2d');***

***pincel.fillStyle = 'lightgray';***

***pincel.fillRect(0, 0, 800, 600);***

***function desenhaCirculo(x, y, raio) {***

***pincel.fillStyle = 'blue';***

***pincel.beginPath();***

***pincel.arc(x, y, raio, 0, 2 \* Math.PI);***

***pincel.fill();***

***}***

***function limpaTela() {***

***pincel.clearRect(0, 0, 800, 600);***

***}***

***var x = 20;***

***var sentido = 1;***

***function atualizaTela() {***

***limpaTela();***

***if ( x > 600) {***

***sentido = -1;***

***} else if (x< 0) {***

***sentido = 1;***

***}***

***desenhaCirculo(x, 20, 20);***

***x = x + sentido;***

***}***

***setInterval(atualizaTela, 10);***

***</script>***

- o codigo acima permite que o circulo va e volte na tela, para isso definimos a variável sentido com valor = 1, e através do if quando o circulo chega ao limite do canvas a variável sentido recebe o valor -1, sendo assim ela vai regredir até a posição 0 do canvas e novamente através de um else if invertemos o sentido atribuindo novamente o valor = 1, dando assim a continuidade de ida e volta do circulo

***<canvas width="600" height="400"></canvas>***

***<script>***

***var tela = document.querySelector('canvas');***

***var pincel = tela.getContext('2d');***

***pincel.fillStyle = 'lightgray';***

***pincel.fillRect(0, 0, 600, 400);***

***function desenhaCirculo(x, y, raio, cor) {***

***pincel.fillStyle = cor;***

***pincel.beginPath();***

***pincel.arc(x, y, raio, 0, 2 \* Math.PI);***

***pincel.fill();***

***}***

***function limpaTela() {***

***pincel.clearRect(0, 0, 600, 400);***

***}***

***var raio = 9;***

***var fatorCrescimento = 0;***

***function atualizaTela() {***

***limpaTela();***

***if (raio > 100) {***

***fatorCrescimento = -1;***

***} else if (raio < 10) {***

***fatorCrescimento = 1;***

***}***

***raio = raio + fatorCrescimento;***

***desenhaCirculo(300, 200, raio, 'blue');***

***}***

***setInterval (atualizaTela, 20);***

***</script>***

- a função acima faz com que tenhamos um circulo que cresce e diminui ate determinados parâmetros, utiliza a mesma logica do exercício de movimentação do circulo

- no JS existe um evento chamado ***onkeydown*** que permite identificar qual tecla esta pressionada

- ***keycode*** é um evento do JS que traz o codigo da tecla que foi pressionada, cada tecla possui um keycode único que já esta definido em uma tabela que é utilizada pelos navegadores

***<canvas width="600" height="400"></canvas>***

***<script>***

***var tela = document.querySelector('canvas');***

***var pincel = tela.getContext('2d');***

***pincel.fillStyle = 'lightgray';***

***pincel.fillRect(0, 0, 600, 400);***

***var x = 20;***

***var y = 20;***

***var esquerda = 37***

***var cima = 38***

***var direita = 39***

***var baixo = 40***

***var taxa = 10;***

***function desenhaCirculo(x, y, raio) {***

***pincel.fillStyle = 'blue';***

***pincel.beginPath();***

***pincel.arc(x, y, raio, 0, 2 \* Math.PI);***

***pincel.fill();***

***}***

***function limpaTela() {***

***pincel.clearRect(0, 0, 600, 400);***

***}***

***function atualizaTela() {***

***limpaTela();***

***desenhaCirculo(x, y, 10);***

***}***

***setInterval(atualizaTela, 20);***

***function leDoTeclado(evento) {***

***if(evento.keyCode == cima) {***

***y = y - taxa;***

***} else if (evento.keyCode == baixo) {***

***y = y + taxa;***

***} else if (evento.keyCode == esquerda) {***

***x = x - taxa;***

***} else if (evento.keyCode == direita) {***

***x = x + taxa;***

***}***

***}***

***document.onkeydown = leDoTeclado;***

***</script>***

- o codigo acima permite que movimentemos o circulo utilizando as teclas do mouse

- nas funções sempre colocamos entre parênteses os parâmetros quando necessários

- nem toda função do JS possui parâmetros, porem quando possuir é necessário que sejam colocados entre parênteses;

- a função ***setInterval*** requer dois paramentros (o codigo ou função a ser processado e o tempo em milissegundos entre os eventos do temporizador.

***<canvas width="800" height="600"></canvas>***

***<script>***

***var tela = document.querySelector('canvas');***

***var pincel = tela.getContext('2d');***

***pincel.fillStyle = 'lightgray';***

***pincel.fillRect(0, 0, 800, 600);***

***var raio = 10;***

***var xAleatorio;***

***var yAleatorio;***

***function desenhaCirculo(x, y, raio, cor) {***

***pincel.fillStyle = cor;***

***pincel.beginPath();***

***pincel.arc(x, y, raio, 0, 2 \* Math.PI);***

***pincel.fill();***

***}***

***function limpaTela() {***

***pincel.clearRect(0, 0, 800, 600);***

***}***

***function desenhaAlvo(x,y) {***

***desenhaCirculo(x, y, raio+20, 'red');***

***desenhaCirculo(x, y, raio+10, 'white');***

***desenhaCirculo(x, y, raio, 'red');***

***}***

***function sorteiaPosicao(maximo) {***

***return Math.floor(Math.random() \* maximo);***

***}***

***function atualizaTela() {***

***limpaTela();***

***xAleatorio = sorteiaPosicao(800);***

***yAleatorio = sorteiaPosicao(600);***

***desenhaAlvo(xAleatorio, yAleatorio);***

***}***

***setInterval(atualizaTela, 2000);***

***function dispara(evento) {***

***var x = evento.pageX - tela.offsetLeft;***

***var y = evento.pageY - tela.offsetTop;***

***if((x > xAleatorio - raio)***

***&& (x < xAleatorio + raio)***

***&& (y > yAleatorio - raio)***

***&& (y < yAleatorio + raio)) {***

***alert('Acertou!');***

***}***

***}***

***tela.onclick = dispara;***

***</script>***

***<meta charset="UTF-8">***

***<canvas width="600" height="400"></canvas>***

***<script>***

***function desenhaQuadrado(x, y, tamanho, cor) {***

***pincel.fillStyle = cor;***

***pincel.fillRect(x, y, tamanho, tamanho)***

***pincel.fill();***

***}***

***function desenhaCirculo(x, y, raio, cor) {***

***pincel.fillStyle = cor; // o padrão é blue***

***pincel.beginPath();***

***pincel.arc(x, y, raio, 0, 2 \* 3.14);***

***pincel.fill();***

***}***

***function desenhaPaletaDeCores() {***

***desenhaQuadrado(xVermelho, yQuadrados, tamanhoQuadrados, 'red');***

***desenhaQuadrado(xVerde, yQuadrados, tamanhoQuadrados, 'green');***

***desenhaQuadrado(xAzul, yQuadrados, tamanhoQuadrados, 'blue');***

***desenhaQuadrado(xRoxo, yQuadrados, tamanhoQuadrados, 'purple');***

***}***

***function lidaComMovimentoDoMouse(evento) {***

***var x = evento.pageX - tela.offsetLeft;***

***var y = evento.pageY - tela.offsetTop;***

***// usando a nova função***

***if(desenha && podeDesenharNaArea(x,y)) {***

***desenhaCirculo(x, y, 5, corAtual);***

***}***

***}***

***function habilitaDesenhar() {***

***desenha = true;***

***}***

***function desabilitaDesenhar() {***

***desenha = false;***

***}***

***function podeDesenharNaArea(x, y) {***

***if(x >= 0 && x < 4\*tamanhoQuadrados && y >= 0 && y < tamanhoQuadrados) {***

***return false;***

***} else {***

***return true;***

***}***

***}***

***function selecionaCor(evento) {***

***var x = evento.pageX - tela.offsetLeft;***

***var y = evento.pageY - tela.offsetTop;***

***// cada condição muda a variável `corAtual`***

***// começamos pela condição do y que é igual em todos os casos***

***if (y > yQuadrados && y < yQuadrados + tamanhoQuadrados) {***

***if (x > xVermelho && x < xVermelho + tamanhoQuadrados) {***

***corAtual = 'red';***

***} else if (x > xVerde && x < xVerde + tamanhoQuadrados) {***

***corAtual = 'green';***

***} else if (x > xAzul && x < xAzul + tamanhoQuadrados) {***

***corAtual = 'blue';***

***} else if (x > xRoxo && x < xRoxo + tamanhoQuadrados) {***

***corAtual = 'purple';***

***}***

***}***

***}***

***var tela = document.querySelector('canvas');***

***var pincel = tela.getContext('2d');***

***pincel.fillStyle = 'lightgray';***

***pincel.fillRect(0, 0, 600, 400);***

***var desenha = false;***

***var corAtual = 'blue';***

***var xVermelho = 0;***

***var xVerde = 50;***

***var xAzul = 100;***

***var xRoxo = 150;***

***var yQuadrados = 0;***

***var tamanhoQuadrados = 50;***

***desenhaPaletaDeCores();***

***tela.onmousemove = lidaComMovimentoDoMouse;***

***tela.onmousedown = habilitaDesenhar;***

***tela.onmouseup = desabilitaDesenhar;***

***tela.onclick = selecionaCor;***

***</script>***

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente