

Interfaces de usuario e Interacción Visualización e Interfaces

Eventos de Usuarios & Formas

Eventos

- Mecanismos que permiten crear una interacción entre los elementos del documento y a las acciones del usuario.
- Existe un número de eventos definidos para todos los elementos:
 - *ONLOAD*
 - *ONMOUSEDOWN*
 - *ONMOUSEENTER*
 - *ONMOUSEDOWN*
 -

Eventos

```
var canvas = document.getElementById("canvas");  
var ctx = canvas.getContext("2d");  
.....  
canvas.onclick = pintarFondo();  
  
function pintarFondo()  
{  
.....  
}
```

Clases y objetos

- **Clase:** definiciones de las propiedades y comportamiento de una estructura.
- **Objeto:** una instancia de una clase.
- **Método:** Algoritmo asociado a un objeto.
- **Propiedad o atributo:** datos asociados a un objeto.

JavaScript & POO

- Definición de una nueva clase

```
function Circle() {  
  this.posX = 4;  
  this.posY = 4;  
  this.radio = 10;  
  this.color = '#141444';  
}
```

```
function Circle(paramPosX, paramPosY, paramRadio, paramColor) {  
  this.posX = paramPosX;  
  this.posY = paramPosY;  
  this.radio = paramRadio;  
  this.color = paramColor;  
}
```

```
Circle.prototype.message = function() {  
  alert("I'm a Circle with Radio: " + this.radio);  
}
```

Métodos de una clase

- Uso

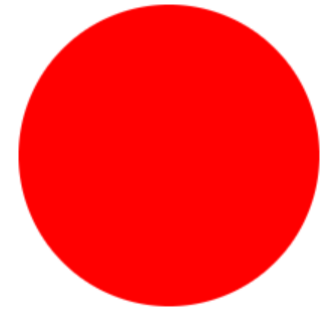
```
var circle1 = new Circle(4,4,10,'#141444');  
var circle2 = new Circle(5,5,10,'#ffffff');  
var circle3 = new Circle(6,6,10,'#000000');  
var circle4 = new Circle(1,9,44,'#14ff44');  
  
circle1.message();  
.....
```

Como Dibujar en Context 2D:

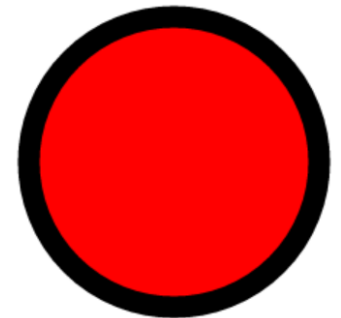
- ***Líneas***
- ***Rectángulo***
- ***Caminos***
- ***Imágenes***
- ***Texto***
- ***Setear colores***
- ***Setear grosor y estilo de lápiz***
- ***...***

Dibujar un Círculo en el context

```
var ctx = document.getElementById("canvas").getContext("2d");  
ctx.fillStyle = "#FF0000";  
ctx.beginPath();  
ctx.arc(250, 250, 100, 0, Math.PI * 2);  
ctx.fill();  
ctx.closePath();
```



```
ctx.beginPath();  
ctx.arc(250, 250, 100, 0, Math.PI * 2);  
ctx.lineWidth = 15;  
ctx.lineCap = 'round';  
ctx.strokeStyle = 'black';  
ctx.stroke();  
ctx.closePath();
```



Fill styles

- **Gradiente**

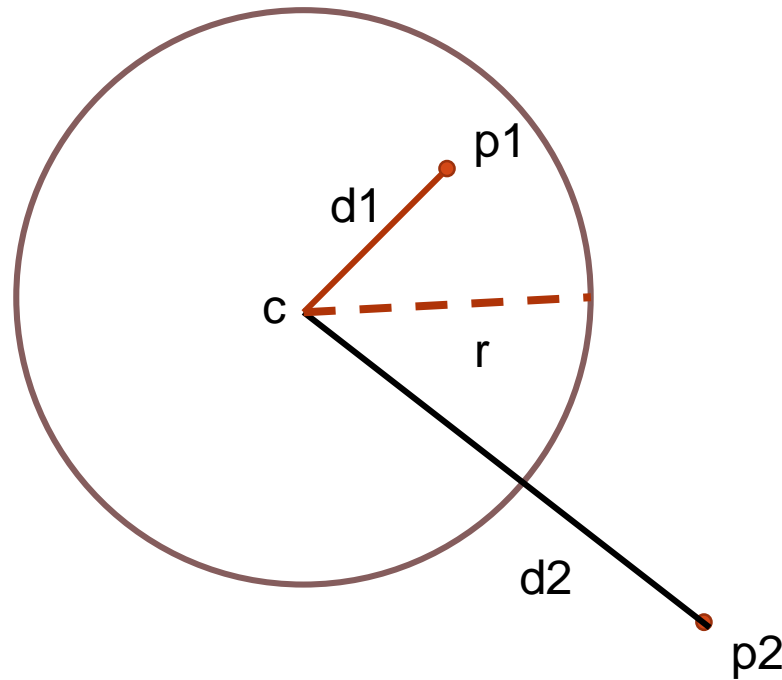
```
var gradient = ctx.createLinearGradient(10, 90, 200, 90);  
gradient.addColorStop(0, 'black');  
gradient.addColorStop(1, 'white');  
ctx.fillStyle = gradient;  
ctx.fillRect(10, 10, 200, 250);
```

- **Imagen**

```
var image = ctx.createPattern(img, "repeat");  
ctx.rect(0, 0, 150, 100);  
ctx.fillStyle = image;  
ctx.fill();
```

Detectar Punto dentro de Figura

Círculo:



$$d1 = \sqrt{[(p1.x - c.x)^2 + (p1.y - c.y)^2]}$$

Parámetros de Mouse Event

```
function mouseDown(e) {  
    //dentro del Element del document  
    var cX = e.clientX;  
    var cY = e.clientY;  
  
    //dentro de la pantalla  
    var sX = e.screenX;  
    var sY = e.screenY;  
}
```

Detectar Punto dentro de Figura

Rectángulo:



pizarrón