

# Processing

## Aula 2

Prof. Wallace Lages

# Aula 2

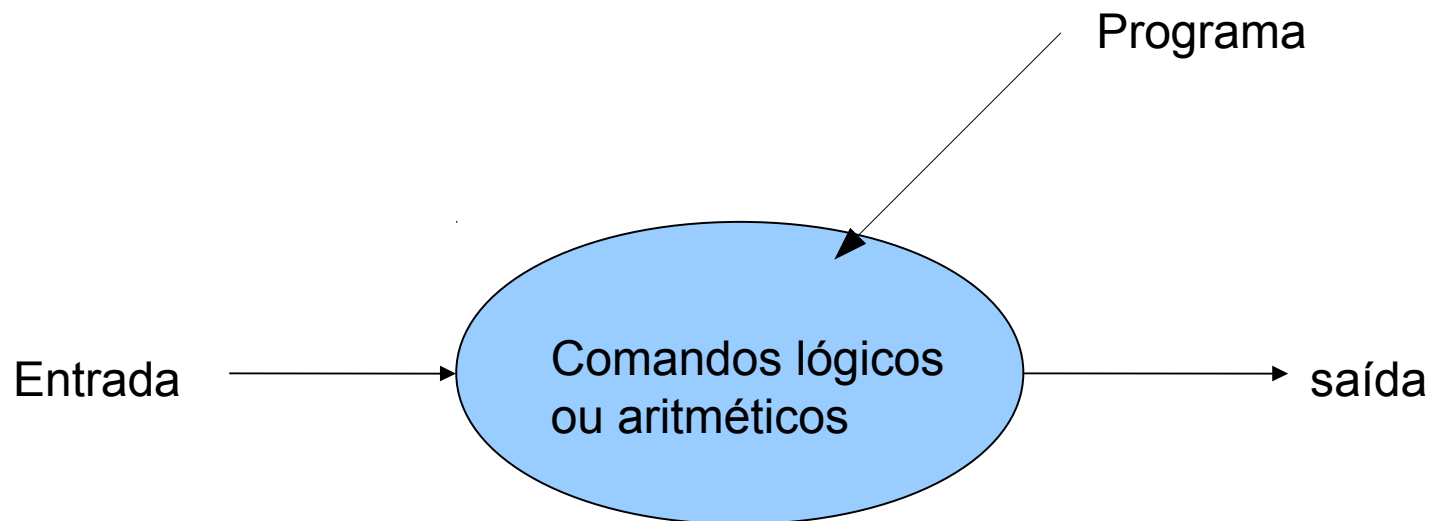
- Computadores e lógica digital
- Variáveis
- Condicionais

# Computador ou calculadora?



# Computador

- Máquina programável capaz de realizar automaticamente uma sequência de operações lógicas e aritméticas



# Z3



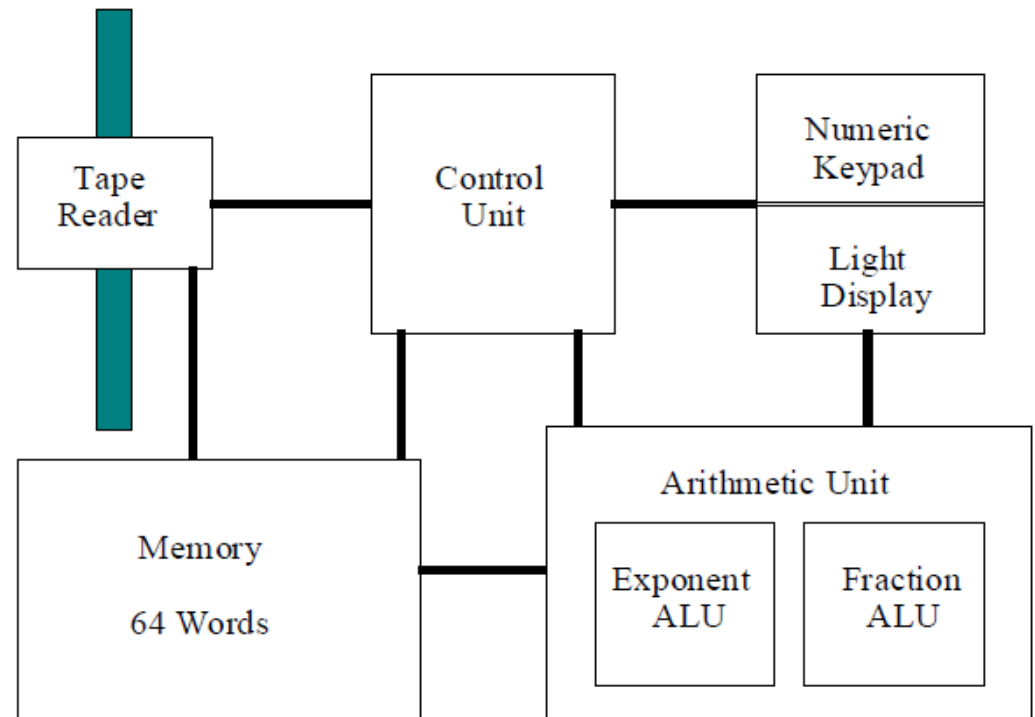
Réplica do Zuse Z3 no Deutsches Museum em Munique - Wikipedia

- Primeiro computador programável (Berlim, 1941).
- Desenvolvido por Konrad Zuse
- Eletro-mecânico (relés)
- Programa armazenado em película de filme 35mm perfurada
- Sem desvios

# Z3

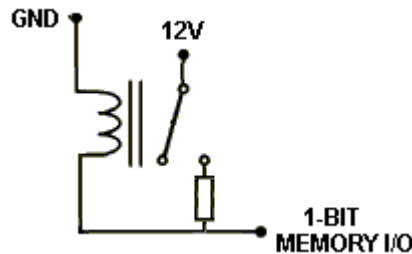
- Operações:

- Ler
- Mostrar
- Multiplicar
- Dividir
- Raiz Quadrada
- Adicionar/Subtrair
- Sinal



# Lógica booleana

- Desenvolvida por George Boole em 1840
- Cálculo lógico realizado sobre um conjunto de dois valores  $\{0, 1\}$
- Conveniente porque era fácil armazenar valores binários e fazer contas com eles.



# Lógica Booleana



A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1



A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

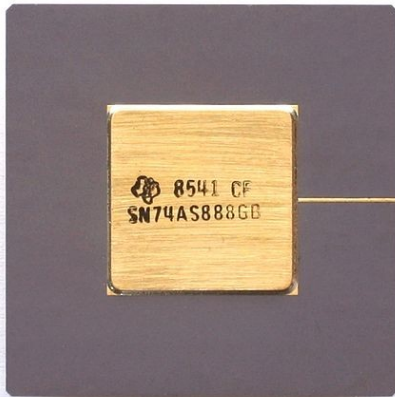


A	NOT A
0	1
1	0

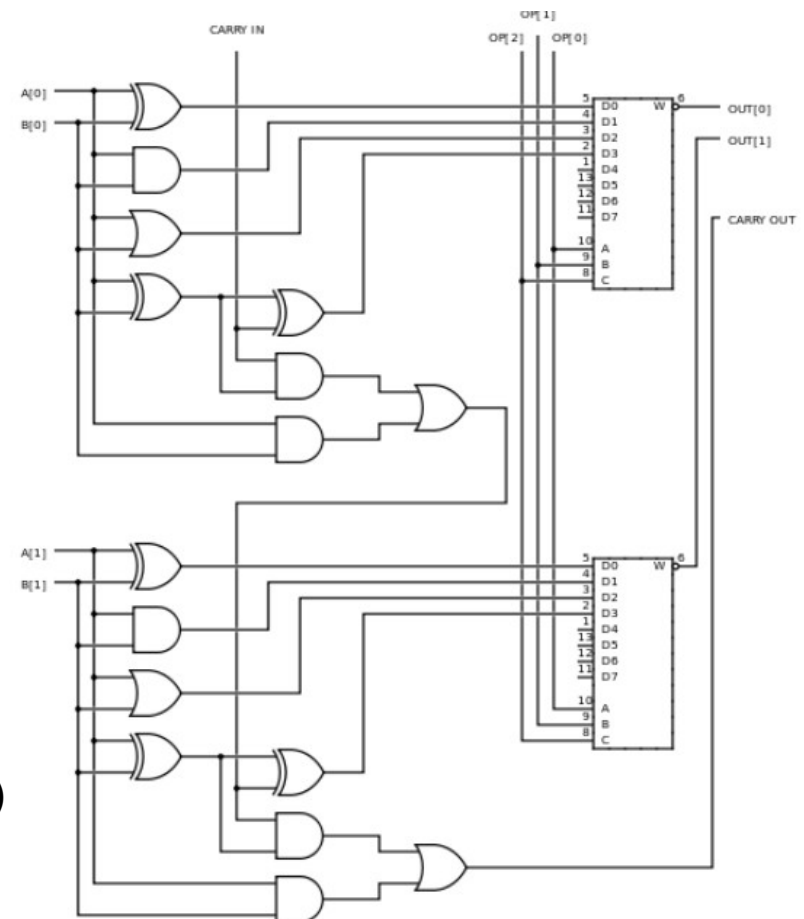


# Unidade lógica Aritmética

- Circuito responsável por realizar operações lógicas e aritméticas



Operações lógicas bit a bit AND,OR,XOR,etc)  
Operações inteiras (soma, multiplicação, etc)



# Codificação em base 2

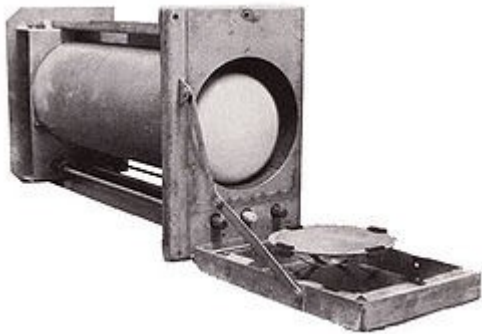
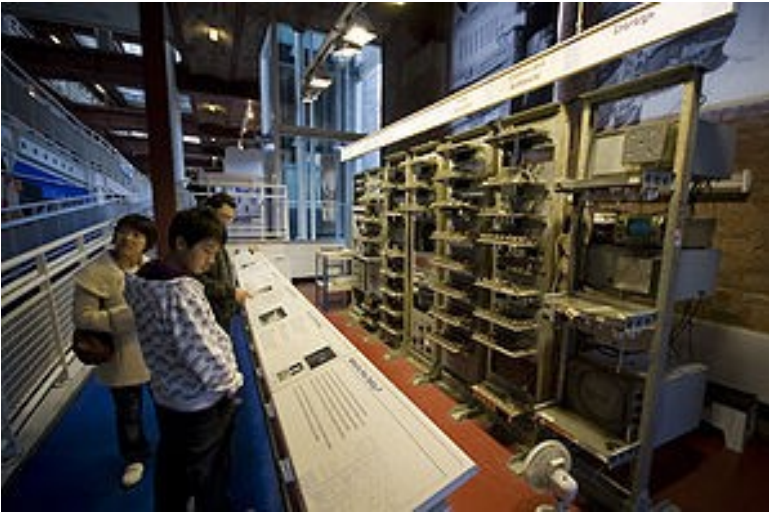
- Binário 1 1 0 1
- Decimal  $1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13$
- 
- 8 bits =  $2^8 = 256$  possibilidades
- 16 bits =  $2^{16} = 65.536$  possibilidades
-

# ENIAC



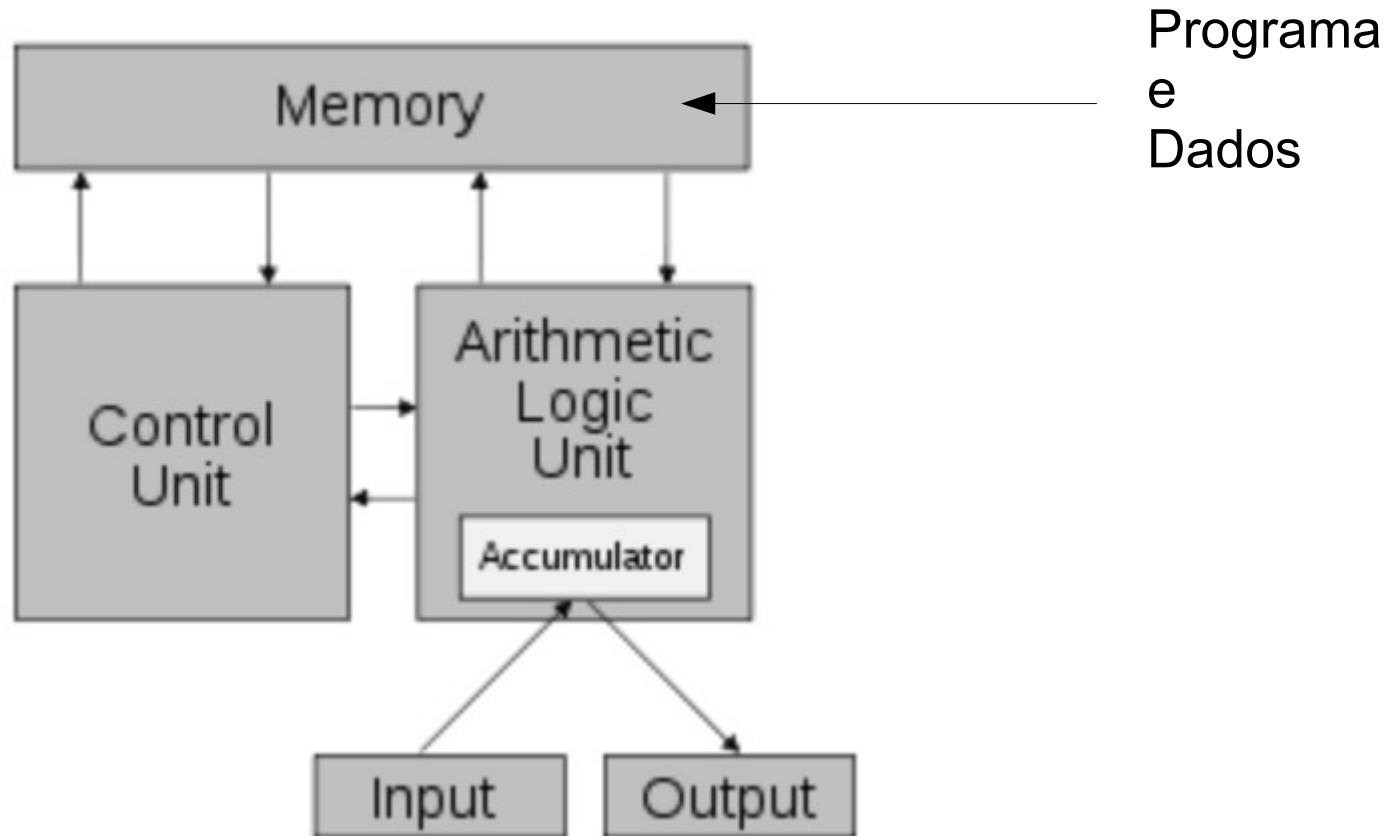
- Primeiro Computador Eletrônico programável (1946, EUA)
- Projetado e desenvolvido pela Universidade da Pensilvânia
- Programação por meio de religação dos fios

# Manchester Small-Scale Experimental Machine



- Primeiro computador com programa armazenado
- Construído na Universidade Vitória de Manchester (1948)
- Utilizava CRT como memória de acesso randômico

# Arquitetura Von Neumann



# Um computador atual



# Um computador atual

- Processador, disco, placa de vídeo, memória



## Core i7 2600 (Sandy Bridge)

Muito mais performance em games exigentes. Conte com o consagrado Core i7 Sandy Bridge.



## HD 1TB (SATA3)

Muito espaço para se instalar jogos e armazenar filmes HD.



## GeForce GTX 560 TI

Uma super placa de vídeo preparada para rodar os jogos atuais em resolução máxima.



## 4GB DDR3 Corsair

Memórias Corsair são certeza de excelente desempenho, em qualquer condição.



# Compiladores

Facilitar a programação dos computadores!

Linguagem de máquina:

110 001111

111 010001



# Compiladores

Linguagem assembly:

Sto 15

Add 17

# Compiladores

Linguagem de alto nível:

$17 + 15$

# Compiladores

Processo de compilação:

17 + 15            =>        110 001111  
   111 010001

# Variáveis

Na matemática: Valor indeterminado

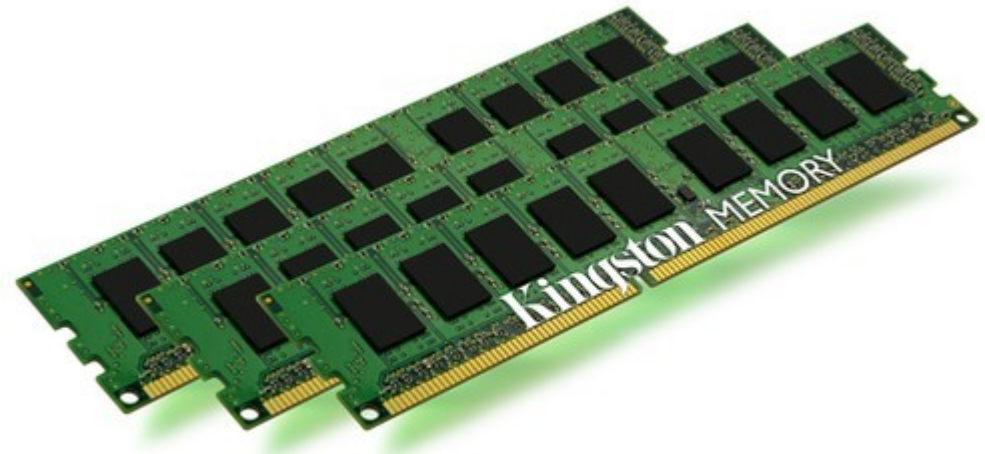
$$20x + 10 = 50$$

Qual o valor de  $x$  ?

# Variáveis

Na computação: Espaço na memória associado a um nome

$$x = 17 + 15$$



# Variáveis

Útil para guardar valores intermediários

$$x = 17 + 15$$

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	50	41	47	45	44	55	4D	50	0F	00	00	00	93	08	00	00	PAGEDUMP.....
0010h:	00	00	03	00	00	C0	97	82	C0	14	48	80	48	2F	48	80	.....H./H.
0020h:	4C	01	00	00	01	00	00	00	E2	00	00	00	00	00	00	00	L.....
0030h:	00	00	00	00	00	00	00	00	00	00	00	00	00	41	47	45	.....AGE
0040h:	50	41	47	45	50	41	47	45	50	41	47	45	50	41	47	45	PAGEPAGEPAGEPAGE
0050h:	50	41	47	45	50	41	47	45	50	41	47	45	00	41	47	45	PAGEPAGEPAGE.AGE
0060h:	B0	D3	46	80	04	00	00	00	CC	F6	02	00	02	00	00	00	..F.....
0070h:	1E	00	00	00	30	00	00	00	6F	00	00	00	00	01	00	00	....0...0.....
0080h:	FF	0E	00	00	00	10	00	00	40	E7	02	00	50	41	47	45	.....@...PAGE
0090h:	50	41	47	45	50	41	47	45	50	41	47	45	50	41	47	45	PAGEPAGEPAGEPAGE
00A0h:	50	41	47	45	50	41	47	45	50	41	47	45	50	41	47	45	PAGEPAGEPAGEPAGE
00B0h:	50	41	47	45	50	41	47	45	50	41	47	45	50	41	47	45	PAGEPAGEPAGEPAGE
00C0h:	50	41	47	45	50	41	47	45	50	41	47	45	50	41	47	45	PAGEPAGEPAGEPAGE
00D0h:	50	41	47	45	50	41	47	45	50	41	47	45	50	41	47	45	PAGEPAGEPAGEPAGE
00E0h:	50	41	47	45	50	41	47	45	50	41	47	45	50	41	47	45	PAGEPAGEPAGEPAGE
00F0h:	50	41	47	45	50	41	47	45	50	41	47	45	50	41	47	45	PAGEPAGEPAGEPAGE
0100h:	50	41	47	45	50	41	47	45	50	41	47	45	50	41	47	45	PAGEPAGEPAGEPAGE
0110h:	50	41	47	45	50	41	47	45	50	41	47	45	50	41	47	45	PAGEPAGEPAGEPAGE
0120h:	50	41	47	45	50	41	47	45	50	41	47	45	50	41	47	45	PAGEPAGEPAGEPAGE
0130h:	50	41	47	45	50	41	47	45	50	41	47	45	50	41	47	45	PAGEPAGEPAGEPAGE
0140h:	50	41	47	45	50	41	47	45	50	41	47	45	50	41	47	45	PAGEPAGEPAGEPAGE

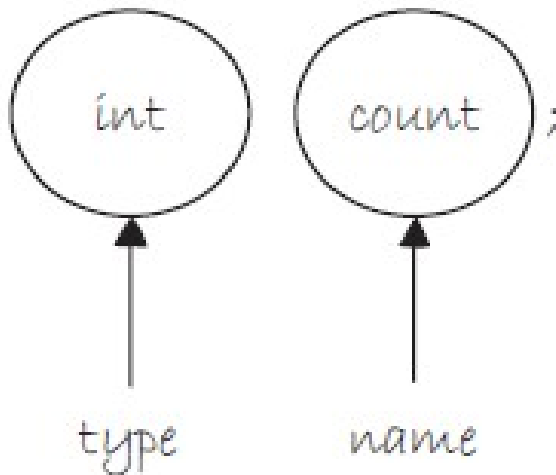
# Variáveis

- Analogia:



# Variáveis

## Declaração



- boolean : true or false
- char : caractere ('a', 'b')
- byte : -128 to 127
- short : -32768 to 32767
- int : -2147483648 to 2147483647
- long : números maiores que int
- float : decimais
- double : decimais de precisão dupla



# Variáveis

## Inicialização

```
int count;  
count = 50;
```

Ou

```
int count = 50;
```

# Variáveis

## Nomes

**width** — largura em pixels da janela de desenho

**height** — altura em pixels da janela de desenho

**frameCount** — número de quadros processados

**frameRate** — taxa de processamento de quadros

**screen.width** — largura em pixels da tela

**screen.height** — altura em pixels da tela

**key** — última tecla pressionada

**keyCode** — código numérico para a tecla pressionada

**keyPressed** — verdadeiro se uma tecla foi pressionada

**mousePressed** — verdadeiro se o mouse foi pressionado

**mouseButton** — botão do mouse pressionado

# Variáveis

## Utilizando variáveis

```
int circleX = 100;
```

```
int circleY = 100;
```

```
void setup() {
```

```
    size(200,200);
```

```
}
```

```
void draw() {
```

```
    background(100);
```

```
    stroke(255);
```

```
    fill(0);
```

```
    ellipse(circleX,circleY,50,50);
```

```
}
```

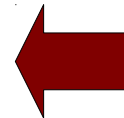
# Variáveis

## Variando as variáveis

```
int circleX = 0;
int circleY = 100;

void setup() {
    size(200,200);
}

void draw() {
    background(255);
    stroke(0);
    fill(175);
    ellipse(circleX,circleY,50,50);
    circleX = circleX + 1;
}
```



# Variáveis

## Exercício

Façam o círculo crescer a medida que desloca

Façam o círculo seguir o mouse e crescer

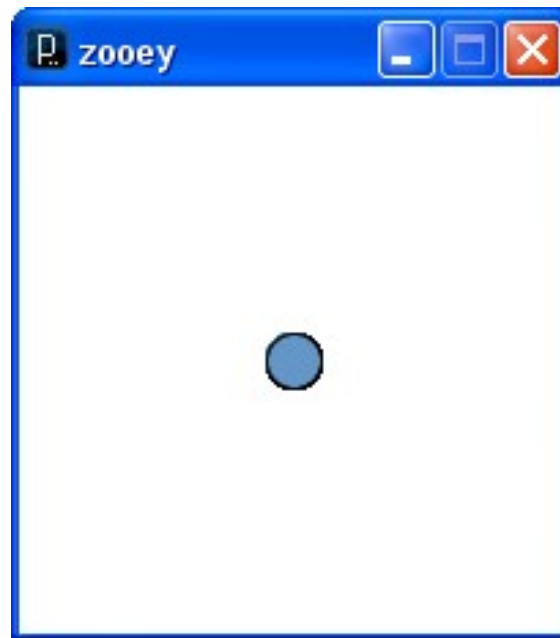
Como variar a velocidade com que o círculo cresce ?

# Variáveis

## Exercício 2

Escreva um programa que desenhe um círculo colorido.

Nota: Utilizem variáveis!



# Variáveis

## Exercício 2

```
float r = 100;
```

```
float g = 150;
```

```
float b = 200;
```

```
float a = 200;
```

```
float diam = 20;
```

```
float x = 100;
```

```
float y = 100;
```

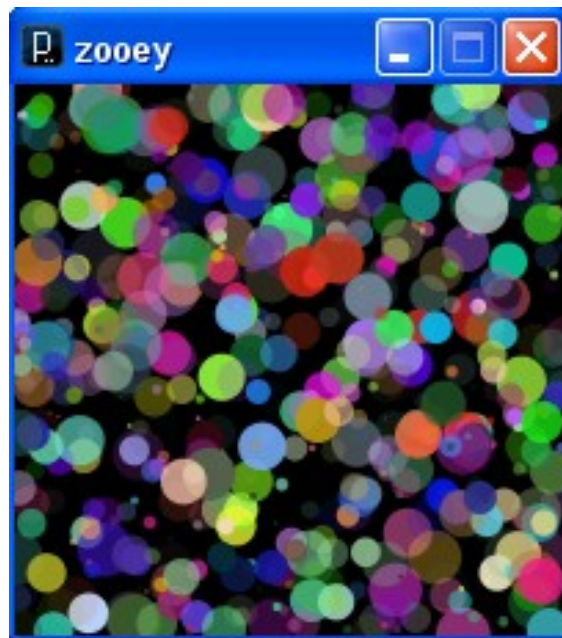
```
void setup() {  
    size(200,200);  
    background(255);  
    Smooth();  
}
```

```
void draw() {  
    stroke(0);  
    fill(r,g,b,a);  
    ellipse(x,y,diam,diam);  
}
```

# Variáveis

## Random

Mudar o código para criar círculos aleatórios de tamanho diferentes





# Variáveis

## Random

Função random:

```
float w = random(1,100);
```

```
int w = (int) random(1,100);
```

```
fill(r, g, b, a) ;
```

```
ellipse(x, y, diam, diam) ;
```

# Condicionais

Expressões booleanas

Testes de Verdadeiro ou Falso.

15 é maior que 20  $\rightarrow$  Falso

22 é menor que 32  $\rightarrow$  Verdadeiro

# Condicionais

Expressões booleanas – Operadores relacionais

> maior que

< menor que

>= maior ou igual

<= menor ou igual

== igual

!= diferente

# Condicionalis

Expressões booleanas – Operadores relacionais

15 > 20

22 < 32

X != Y

Z <= 10

# Condicionais

Expressões booleanas – if, else, else if

Se o mouse estiver no lado esquerdo da tela

Desenhe um retângulo do lado esquerdo da tela

# Condicionais

Expressões booleanas – if, else, else if

Se o mouse estiver no lado esquerdo da tela

Desenhe um retângulo do lado esquerdo da tela

```
If ( mouseX < width/2) {  
    Fill (255)  
    rect (0, 0, width/2, width/2)  
}
```

# Condicionais

Expressões booleanas – if, else, else if

```
if (expressão booleana) {  
    // código a ser executado se verdadeiro  
}
```

# Condicionais

Expressões booleanas – if, else, else if

```
if (expressão booleana A) {  
    // código a ser executado se verdadeiro  
}  
else if (expressão booleana B){  
    //código a ser executado se A for falso e B verdadeiro  
}  
else{  
    //código a ser executado se A e B forem falsos  
}
```



# Condicionais

Expressões booleanas – if, else, else if

```
if (expressão booleana) {  
    // código a ser executado se verdadeiro  
}  
else {  
    //código a ser executado se falso  
}
```

# Condicionais

Expressões booleanas – if, else, else if

```
if (expressão booleana) {  
    // código a ser executado se verdadeiro  
}  
else {  
    //código a ser executado se falso  
}
```

# Condicionais

Expressões booleanas –exercício

Escreva um programa que:

1-Possua 3 variáveis (R,G e B)

2-Desenhe o fundo continuamente dessas 2 cores

3-Incremente R se o mouse estiver do lado direito da tela

4-Decremente R se o mouse estiver do lado esquerdo da tela

5-Não deixa o valor de R passar do intervalo (0,255)

# Condicionais

## Expressões booleanas – exercício

```
float r = 150;    void draw() {
float g = 0;      background(r,g,b);
float b = 0;      stroke(255);
void setup() {   line(width/2,0,width/2,height);
size(200,200);  if(mouseX > width/2) {
}               r = r + 1;
                } else {
                r = r - 1;
                }
                if (r > 255) {
                    r = 255;
                } else if (r < 0) {
                    r = 0;
                }
            }
}
```

# Condicionais

## Expressões booleanas – constrain

```
if (r > 255) {  
    r = 255;  
} else if (r < 0) {  
    r = 0;  
}  
  
r = constrain(r, 0, 255);
```

# Condicionais

## Operadores lógicos

Se o mouse estiver no lado direito da tela

E

o mouse estiver na parte de baixo da tela

Desenhe um retângulo no canto direito inferior

```
if (mouseX > width/2) {  
    if (mouseY > height/2) {  
        fill(255);  
        rect(width/2,height/2,width/2,height/2);  
    }  
}
```