

Programação Multimédia

Variáveis

Memória do Programa (Variáveis)

- Permite-nos armazenar valores (dados)
- Os dados são guardados na memória RAM do computador
- As variáveis representam posições de memória que podem guardar um valor
- As variáveis têm um nome que utilizamos no programa
- Não precisamos de decorar a posição de memória

Tipos de Dados

- Cada variável pode guardar valores de um determinado tipo:
 - `int`, `long`
 - `float`, `double`
 - `boolean`
 - `char`
- Estes são *tipos simples*
- Vectores e matrizes
 - *Tipos complexos*
 - Aula mais à frente (iteracção)

Declaração de Variáveis (processing)

```
[tipo] nomeDaVariavel;
```

- Isto declara a variável apenas
- O Programa fica “a saber” que vamos utilizar uma posição de memória para guardar dados de um determinado tipo
- Exemplo:

```
int idade;
```

Atribuição

- Para guardarmos valores numa variável temos de *atribuir* esses valores:
`idade = 31;`
- Em qualquer altura podemos atribuir um valor a uma variável

Inicialização

- Antes de usar uma variável é preciso colocar algo lá dentro – inicializar a variável
- Qual o resultado do programa seguinte?

```
int x;  
int y;  
point (x, y) ;
```

Variáveis: Examples->Data- >Variables

```
// Variables
// by REAS <http://reas.com>

// Variables are used for storing values.
// Changing the values of variables 'a' and 'b' significantly change the composition.

// Created 2 September 2002

size(200, 200);
background(0);
stroke(153);

int a = 20;
int b = 50;
int c = a*8;
int d = a*9;
int e = b-a;
int f = b*2;
int g = f+e;

line(a, f, b, g);
line(b, e, b, g);
line(b, e, d, c);
line(a, e, d-e, c);
```

Visibilidade (scope)

- Podemos declarar variáveis dentro e fora dos métodos
- Uma variável declarada dentro de um método só pode ser usada dentro desse método

Visibilidade (scope)

- Podemos declarar variáveis com o mesmo nome dentro e fora de um método ou em métodos diferentes:

```
void setup() {  
    int a;  
    a = 2;  
    mudaA();  
    print(a);  
}
```

```
void mudaA() {  
    int a;  
    a = 3;  
}
```

- Exemplo: Variable_Scope (Examples->Data->Variable_Scope)

Instruções

- A atribuição de valores a variáveis não tem de ser feita com valores simples (literais)
- Podemos atribuir a uma variável o resultado de uma expressão mais complicada:

```
int a;
```

```
int b;
```

```
a = 3*5;
```

```
b = a*35+22;
```

Operadores Aritméticos

- Adição: “+”
- Subtracção: “-”
- Divisão: “/”
- Multiplicação: “*”
- Resto da Divisão Inteira: “%”

Atribuição com métodos

- Podemos também usar alguns métodos nas atribuições:

```
float a;
```

```
a = 3 * random(100);
```

- `random()`
 - Procurar na referência
- Podemos usar métodos que devolvem valores
 - Aula sobre funções (31 Março)

Exemplo: Animar um ponto

```
int x = 0;
int y = 0;

void setup() {
  size(200, 200);
  framerate(5);
}

void draw() {
  background(0); // limpar o fundo (pintar a preto)

  stroke(255);   // usar a cor branca para pintar

  point(x, y);  // desenhar um ponto branco na posicao (x, y)

  // actualizar a posicao (x, y)
  x = x + 1;
  y = y + 1;
}
```

Exemplo: Animar um ponto

- Modificar a expressão de actualização da posição:

```
x = x + 1 ;
```

```
y = y + 1 ;
```

para

```
x = x + 2 ;
```

```
y = y + 5 ;
```

Projecto Semanal

1. Crie um programa em Processing que desenhe 5 linhas em posições diferentes alterando apenas os valores de variáveis. Ou seja a instrução para desenhar deve ser sempre `line(x0, y0, x1, y1)`; . Altere apenas os valores das variáveis.
2. Crie um programa que anime uma linha no ecrã. Irá ter de utilizar o método `draw()` (veja um dos exemplos e altere). Pense nas variáveis que terá de utilizar, onde as deverá inicializar e onde as deverá actualizar. Não faz mal se a sua linha desaparecer do ecrã a certa altura, mas pense no que teria de fazer para evitar isso.
3. Diga o que está mal no programa seguinte:

```
int largura;
```

```
void setup() {  
    size(200, 200);  
    largura = 100;  
    altura = 100;  
}
```

```
void draw() {  
    rect(10, 10, largura, altura);  
}
```

Corrija-o.