Variáveis e Tipos de Numéricos

Sérgio Lopes

I. Conhecer as estruturas que permitem guardar dados
II. Reconhecer os processos e restrições na criação de variáveis
III. Saber declarar e utilizar variáveis
IV. Conhecer a diferença entre os vários tipos de dados
V. Saber escolher o tipo adequado à situação
VI. Prever a conversão entre tipos feita pelo compilador

Objectivos

O que são?

Elementos onde podemos guardar os dados.

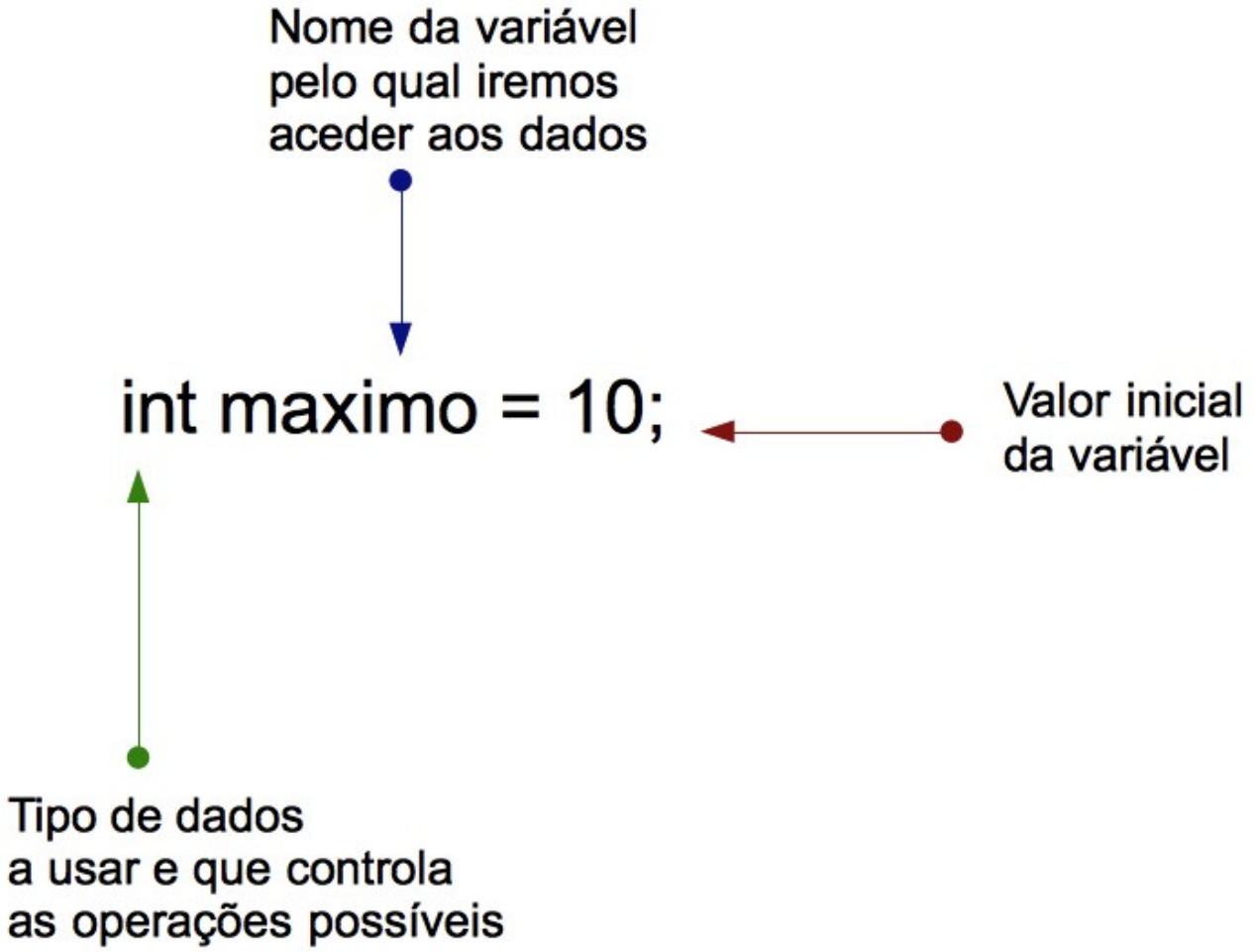
Representam zonas de memória onde os nossos dados são guardados e permitem o acesso a essas zonas através de um elemento ao qual damos um nome.

Regras

Genericamente: podem conter letras entre a e z, A e Z, números e o caracter “_”

(a) Não aceitam caracteres com acentuação
(b) Não podem começar com números, não podem existir variáveis cujo nome seja apenas um número, ex: 78 é inválido
(c) São acessíveis apenas no domínio em que forma declaradas

Dissecar


Nome da variável
pelo qual iremos
aceder aos dados
int maximo = 10;
Valor inicial
da variável
Tipo de dados
a usar e que controla
as operações possíveis

Constantes

Permitem definir variáveis cujo valor não pode ser alterado.

Uso da palavra reservada “const” antes da declaração limita a variável a uma única atribuição a partir da qual o valor dessa variável não pode ser alterado.

Ex:

```
const PI = 3.14;
```

#define

Por vezes, também chamadas de constantes, no entanto não o são!

A directiva de pré-processamento “define” permite declarar elementos do tipo chave-valor, em que a chave é substituída automaticamente pelo compilador aquando da compilação.

Ao contrários das constantes não é um valor que exista em memória nem tem um tipo de dados associado, não é portanto uma variável.

Domínio/Âmbito/Escopo

Uma variável é sempre limitada a um domínio, correspondente ao local onde foi declarada, e não existe for a desse domínio.

```
int main(int argc, char* argv[])
{
    int x = 0;
    {
        int z = 1;
    }

    x = z + x; //<-- erro de compilação
    return x;
}
```

Variáveis Globais

O domínio das variáveis globais é o ficheiro onde estão definidas.

Tipos de Dados Numéricos

Lista de Tipos

Characteres	signed char unsigned char char	Permite guardar caracteres gráficos, ex: a, b, c, 1, 0, #, etc. O sinal não afecta o tamanho de um char.
Inteiros	short unsigned short int unsigned int long unsigned long long long unsigned long long	Dados inteiros, sem componentes decimais, em que cada variante permite números com tamanho diferente.
Vírgula Flutuante	double float	Dados fraccionários, também chamados de vírgula flutuante, permitem valores com componentes decimais. A diferença prende-se com a precisão permitida

Com Sinal/Sem Sinal

1. Com sinal por omissão
2. A remoção de sinal duplica a quantidade de valores permitidos
3. Sem sinal, o limite inferior é zero

Caracteres

Não confundir caracter gráfico numérico com o seu valor numérico ou com o valor numérico interno!

Permitem guardar caracteres, um apenas, que podem ser usados Como os restantes valores numéricos.

Internamente são valores inteiros entre 0 e 255.

Inteiros

Valores inteiros não aceitam componente decimal.

Qualquer atribuição de valores decimais resulta na perda de precisão!

Vírgula Flutuante?

Para quando são necessários valores decimais.

Vírgula flutuante porque a vírgula flutua/desloca.

“float” e “double” diferem no número de bits disponíveis.

Operações

As operações entre tipos diferentes forçam à conversão de tipos.

Exercícios

Questões

1. Permitem guardar dados em zonas de memória
2. Possuem algumas restrições na criação dos nomes
3. Estão limitadas ao domínio onde foram declaradas (salvo a excepções indicas)
4. Existem precisões diferentes para tipos diferentes
5. Por omissão todos os tipos têm sinal
6. O tipo deve ser escolhido de acordo com o objectivo

Resumindo