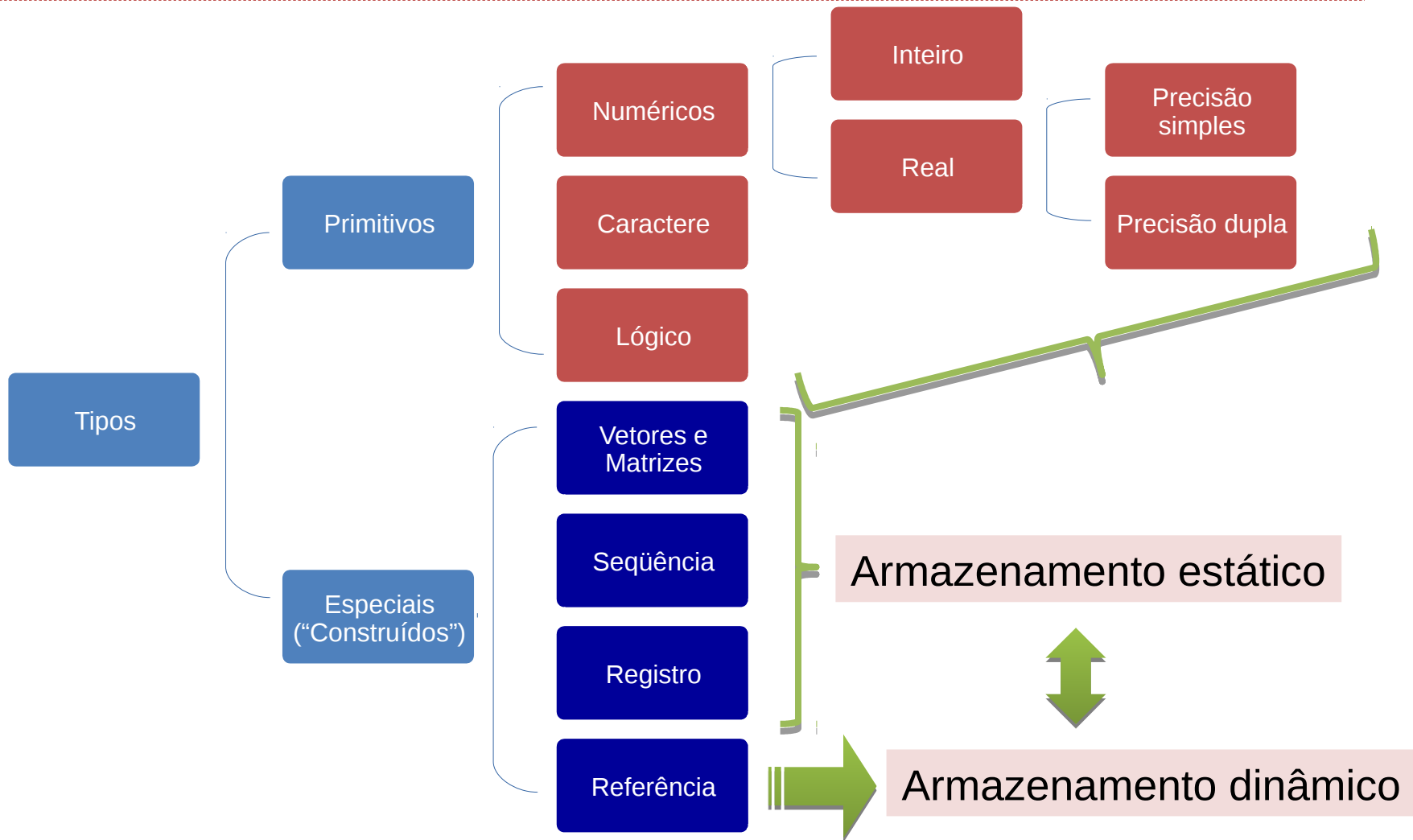

Tipos de dados, constantes e variáveis

Prof. Karl Apaza Agüero

Tipos de Dados



Tipos de Dados

- ▶ Tipos primitivos

- ▶ `char`:

- ▶ Caractere

- ▶ `int`:

- ▶ Inteiro

- ▶ `float`:

- ▶ Real

- ▶ `double`:

- ▶ Real

Tipos de Dados

- ▶ Tipos e valores para **números**
 - ▶ O tipo **int**
 - permite a representação de números negativos e positivos inteiros em *4 bytes*
 - ▶ O tipo **float**
 - utiliza 4 bytes
 - armazena valores reais com 8 dígitos de precisão
 - ▶ O tipo **double**
 - utiliza 8 bytes
 - armazena valores reais com 16 dígitos de precisão

```
int n1=10;  
float n2 = 18.52;  
double n3= 0.123456789;
```

Tipos de Dados

► Tipos e valores para **caracteres**

► O tipo **char**

- utiliza 1 *byte*
- armazena caracteres
- Caracteres geralmente são armazenados em códigos, usualmente o código ASCII - *American Standard Code for Information Interchange*

```
char caractere;    // declaração da variável caractere de tipo char
caractere = 'A';   // ambas as atribuições armazenam
caractere = 65;    // na variável caractere o código ASCII 65
```

```
// declaração das variáveis numero e caractere de tipo char
char numero, caractere;
```

```
numero = 3;        //armazena na variável o valor 3
caractere = '3';    //armazena na variável o código ASCII 51
```

Tabela ASCII

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 Space		64	40	100	@ @		96	60	140	` `	
1	1	001	SOH (start of heading)	33	21	041	! !		65	41	101	A A		97	61	141	a a	
2	2	002	STX (start of text)	34	22	042	" "		66	42	102	B B		98	62	142	b b	
3	3	003	ETX (end of text)	35	23	043	# #		67	43	103	C C		99	63	143	c c	
4	4	004	EOT (end of transmission)	36	24	044	$ \$		68	44	104	D D		100	64	144	d d	
5	5	005	ENQ (enquiry)	37	25	045	% %		69	45	105	E E		101	65	145	e e	
6	6	006	ACK (acknowledge)	38	26	046	& &		70	46	106	F F		102	66	146	f f	
7	7	007	BEL (bell)	39	27	047	' '		71	47	107	G G		103	67	147	g g	
8	8	010	BS (backspace)	40	28	050	((72	48	110	H H		104	68	150	h h	
9	9	011	TAB (horizontal tab)	41	29	051))		73	49	111	I I		105	69	151	i i	
10	A	012	LF (NL line feed, new line)	42	2A	052	* *		74	4A	112	J J		106	6A	152	j j	
11	B	013	VT (vertical tab)	43	2B	053	+ +		75	4B	113	K K		107	6B	153	k k	
12	C	014	FF (NP form feed, new page)	44	2C	054	, ,		76	4C	114	L L		108	6C	154	l l	
13	D	015	CR (carriage return)	45	2D	055	- -		77	4D	115	M M		109	6D	155	m m	
14	E	016	SO (shift out)	46	2E	056	. .		78	4E	116	N N		110	6E	156	n n	
15	F	017	SI (shift in)	47	2F	057	/ /		79	4F	117	O O		111	6F	157	o o	
16	10	020	DLE (data link escape)	48	30	060	0 0		80	50	120	P P		112	70	160	p p	
17	11	021	DC1 (device control 1)	49	31	061	1 1		81	51	121	Q Q		113	71	161	q q	
18	12	022	DC2 (device control 2)	50	32	062	2 2		82	52	122	R R		114	72	162	r r	
19	13	023	DC3 (device control 3)	51	33	063	3 3		83	53	123	S S		115	73	163	s s	
20	14	024	DC4 (device control 4)	52	34	064	4 4		84	54	124	T T		116	74	164	t t	
21	15	025	NAK (negative acknowledge)	53	35	065	5 5		85	55	125	U U		117	75	165	u u	
22	16	026	SYN (synchronous idle)	54	36	066	6 6		86	56	126	V V		118	76	166	v v	
23	17	027	ETB (end of trans. block)	55	37	067	7 7		87	57	127	W W		119	77	167	w w	
24	18	030	CAN (cancel)	56	38	070	8 8		88	58	130	X X		120	78	170	x x	
25	19	031	EM (end of medium)	57	39	071	9 9		89	59	131	Y Y		121	79	171	y y	
26	1A	032	SUB (substitute)	58	3A	072	: :		90	5A	132	Z Z		122	7A	172	z z	
27	1B	033	ESC (escape)	59	3B	073	; ;		91	5B	133	[[123	7B	173	{ {	
28	1C	034	FS (file separator)	60	3C	074	< <		92	5C	134	\ \		124	7C	174	| 	
29	1D	035	GS (group separator)	61	3D	075	= =		93	5D	135]]		125	7D	175	} }	
30	1E	036	RS (record separator)	62	3E	076	> >		94	5E	136	^ ^		126	7E	176	~ ~	
31	1F	037	US (unit separator)	63	3F	077	? ?		95	5F	137	_ _		127	7F	177	 DEL	

Source: www.LookupTables.com

Tipos de Dados

- ▶ Modificadores dos tipos básicos
 - ▶ `short`
 - ▶ Exemplo: `short int n=10;`
 - ▶ `long`
 - ▶ Exemplo: `long int n=10000000;`
 - ▶ `signed`
 - ▶ Exemplo: `signed int n=-10;`
 - ▶ `unsigned`
 - ▶ Exemplo: `unsigned int n=10;`
-

Tipos de Dados

Tipo	Bytes	Faixa Mínima
char	1	-127 a 127
unsigned char	1	0 a 255
signed char	1	-127 a 127
int	4	-2.147.483.648 a 2.147.483.647
unsigned int	4	0 a 4.294.967.295
signed int	4	-2.147.483.648 a 2.147.483.647
short int, short	2	-32.768 a 32.767
unsigned short int	2	0 a 65.535
signed short int	2	-32.768 a 32.767
long int, long	4	-2.147.483.648 a 2.147.483.647
signed long int	4	-2.147.483.648 a 2.147.483.647
unsigned long int	4	0 a 4.294.967.295
long long int, long long	8	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807
signed long long int, signed long long	8	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807
unsigned long long int, unsigned long long	8	0 a 18.446.744.073.709.551.615
float	4	8 dígitos de precisão
double	8	16 dígitos de precisão
long double	12	16 dígitos de precisão

Tipos de Dados

► Problema:

Fazer um programa que gere a saída a seguir:

Entre com um caractere qualquer.

d

Codigo ASCII do caractere d vale 100.

Agora dois inteiros separados por espaco.

2 5

A soma destes numeros vale 7.

Solução

```
#include <stdio.h>
int main(){
    printf("Entre com um caractere qualquer.\n");
    char c;
    scanf("%c",&c);

    printf("Codigo ASCII do caractere %c vale %d.\n", c,c);

    printf("Agora dois inteiros separados por espaco.\n");
    int a,b;
    scanf("%d%d",&a, &b);

    printf("A soma destes numeros vale %d.\n", a+b);

    return 0;
}
```

Constantes

- ▶ Valor não modificável durante a execução de um programa
 - ▶ Exemplos:
 - ▶ Numérica: 1997, -3, 123.45, +23.45e-10
 - ▶ Caractere: 'a', 'A', '\n', '9'
 - ▶ Sequência de caracteres: "alo mundo!!!"
-

Constantes

- Uso da diretiva `#define` para definir uma constante

```
#include <bibliotecas>
#define <constantes>
<variaveis globais>

int main ([parâmetros]) {
    <variaveis locais>
    <instrucoes>
    return 0;
}
```

```
#define PI      3.14
#define TRUE    1
#define FALSE   0
```

Constantes

► Problema:

Fazer um programa que solicite um valor de raio ao usuário, e calcule e imprima a área de uma circunferência

Observação:

Considere $\pi = 3.14$

$\text{area} = \pi \times \text{raio}^2$

Solução

```
#include <iostream>
using namespace std;
#define PI 3.14
int main(){
    float r;
    cin>>r;
    float area=PI*r*r;
    cout<<area<<endl;
    return 0;
}
```

Variáveis

- ▶ Frequentemente precisamos armazenar dados referentes ao problema, como um nome, um número ou mesmo o resultado de uma operação
- ▶ Para armazenar esses dados, precisamos solicitar ao computador que ele reserve uma área da memória para nosso uso
- ▶ A forma de solicitar ao computador que reserve memória é chamada de **declaração de variáveis**

```
int i;  
float f;  
char c;
```

Variáveis

```
#include <bibliotecas>
#define <constantes>
<variaveis globais>

int main ([parâmetros]) {
    <variaveis locais>
    <instrucoes>
    return 0;
}
```


Variáveis

- ▶ Declaração de uma variável
 - ▶ Esquema genérico (sintaxe):

<tipo de dado> <nome da variável>;



Quantidade de
armazenamento que deve ser
reservada para os objetos
declarados com esse tipo



Endereço de memória

```
int x, y;
```

```
float raio;
```

```
char letra;
```

Variáveis

► Declaração e/ou atribuição

//Exemplo 1:

```
int x;    //declaração de variável x
int y;    //declaração de variável y
x = 2;    //atribuição de valor a x
y = 100;  //atribuição de valor a y
```

//Exemplo 2:

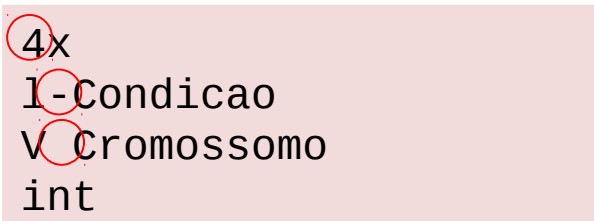
```
int x,y;  //declaração de duas variáveis x e y
x = 2;    //atribuição de valor a x
y = 100;  //atribuição de valor a y
```

//Exemplo 3:

```
int x=2,y=100; //declaração e atribuição a variáveis x e y
```

Variáveis

- ▶ Nomes das variáveis
 - ▶ Só podem conter letras, dígitos e o caractere ‘_’
 - ▶ Todo primeiro caractere deve ser sempre uma letra ou o caractere ‘_’
 - ▶ Letras maiúsculas e minúsculas são consideradas caracteres diferentes
 - ▶ Palavras reservadas não podem ser usadas como nomes de variáveis
- ▶ Nomes inválidos:



- 4x
- l-Condicao
- V Cromossomo
- int

Variáveis

- ▶ Nomes das variáveis
 - ▶ Refletir sobre o significado do valor que a variável ou constante possuirá
 - ▶ É boa política escolher nomes que indiquem a função da variável:

```
Raio, raio, RAI0  
mediaNotas  
taxa_imposto
```

- ▶ Raio, raio e RAI0 referem-se a diferentes variáveis.
Sugestão:
 - evite diferenciar variáveis por letras maiúsculas e minúsculas
 - usar letras maiúsculas para representar constantes
-

Variáveis

► Problema:

Fazer um programa que solicite 2 números (quilômetros) e imprima a conversão destes números de quilômetros a milhas.

Observação:

Considere o valor de conversão

1 Milha = 1.609 Quilômetro

Solução

```
#include <iostream>
using namespace std;
#define FATOR 1.609
int main(){
    float a,b;
    cin>>a>>b;
    cout<<a/FATOR<<endl;
    cout<<b/FATOR<<endl;
    return 0;
}
```