Laboratório de Programação I

Tipos simples de dados

Conceito de tipo de dados

Um "tipo" é definido por três propriedades. As duas primeiras compõem a sua definição conceitual (ou abstrata). A terceira refere-se à sua realização física.

⇨ **Plano abstrato**

- **Conjunto de valores**
- **Conjunto de operações**

⇨ **Plano concreto** — **Organização física**

Conceito de tipo de dados

Propriedades que definem um tipo de dados

⇨ **CONJUNTO DE <u>VALORES</u>**

Especifica os valores que objetos deste tipo podem assumir.
Depende da linguagem e do compilador.

⇨ **CONJUNTO DE <u>OPERAÇÕES</u>**

Especifica as operações que podem ser aplicadas a valores deste tipo.
Depende da linguagem.

⇨ **ORGANIZAÇÃO <u>FÍSICA</u>**

Especifica as convenções utilizadas na interpretação dos bits e bytes da memória para representar valores do tipo.
Depende do compilador.

Conceito de tipo de dados

Exemplo: tipo "integer"

⇨ **CONJUNTO DE <u>VALORES</u>**

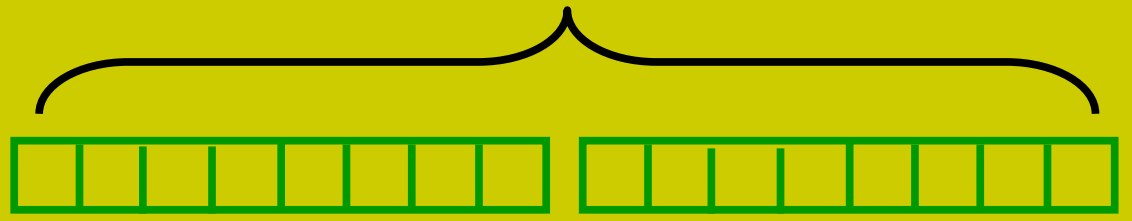
{ min, ... , -3, -2, -1, 0, 1, 2, 3, ... , máx }
"min" e "máx" dependem da representação física:
Se 1 byte, min = -128 e máx = 127;
Se 2 bytes, min = -32.768 e máx = 32.767 etc

⇨ **CONJUNTO DE <u>OPERAÇÕES</u>**

Aritméticas: { +, -, *, /, % }
Relacionais: { ==, !=, >, >=, <, <= }
Atribuição: { = } e várias outras

⇨ **ORGANIZAÇÃO <u>FÍSICA</u>**

Exemplo com 2 bytes:



Conceito de tipo de dados

Exemplo: tipo "char"

⇨ **CONJUNTO DE <u>VALORES</u>**

{ 'A', 'B', ... 'Z', 'a', ... 'z', '0' ... '9', '!', '@', '#', ... }

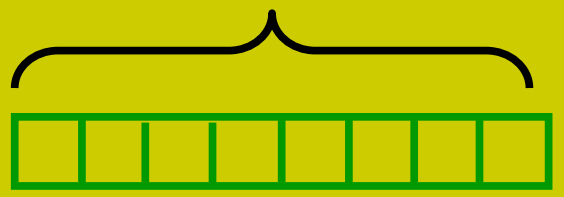
⇨ **CONJUNTO DE <u>OPERAÇÕES</u>**

Relacionais: { ==, !=, >, >=, <, <= }
Atribuição: { = } e outras

⇨ **ORGANIZAÇÃO <u>FÍSICA</u>**

Exemplo com 1 byte:

8 bits, convenção ASCII



Conceito de tipo de dados

Conjunto de valores x organização física

- **Um certo conjunto de valores pode sugerir ou demandar uma certa organização física mínima, suficiente para a sua representação.**

- **Mas um conjunto de valores não determina a organização física dos dados na memória. Outros critérios são levados em conta (por exemplo, economia de espaço e eficiência no acesso aos dados);**

- **Uma determinada organização física pode sugerir um conjunto de valores, mas necessita ainda de regras para interpretação dos mesmos (ex: inteiro com ou sem sinal).**

Conceito de tipo de dados

Comparando dois tipos de dados.
Exemplo: "integer" e "float"

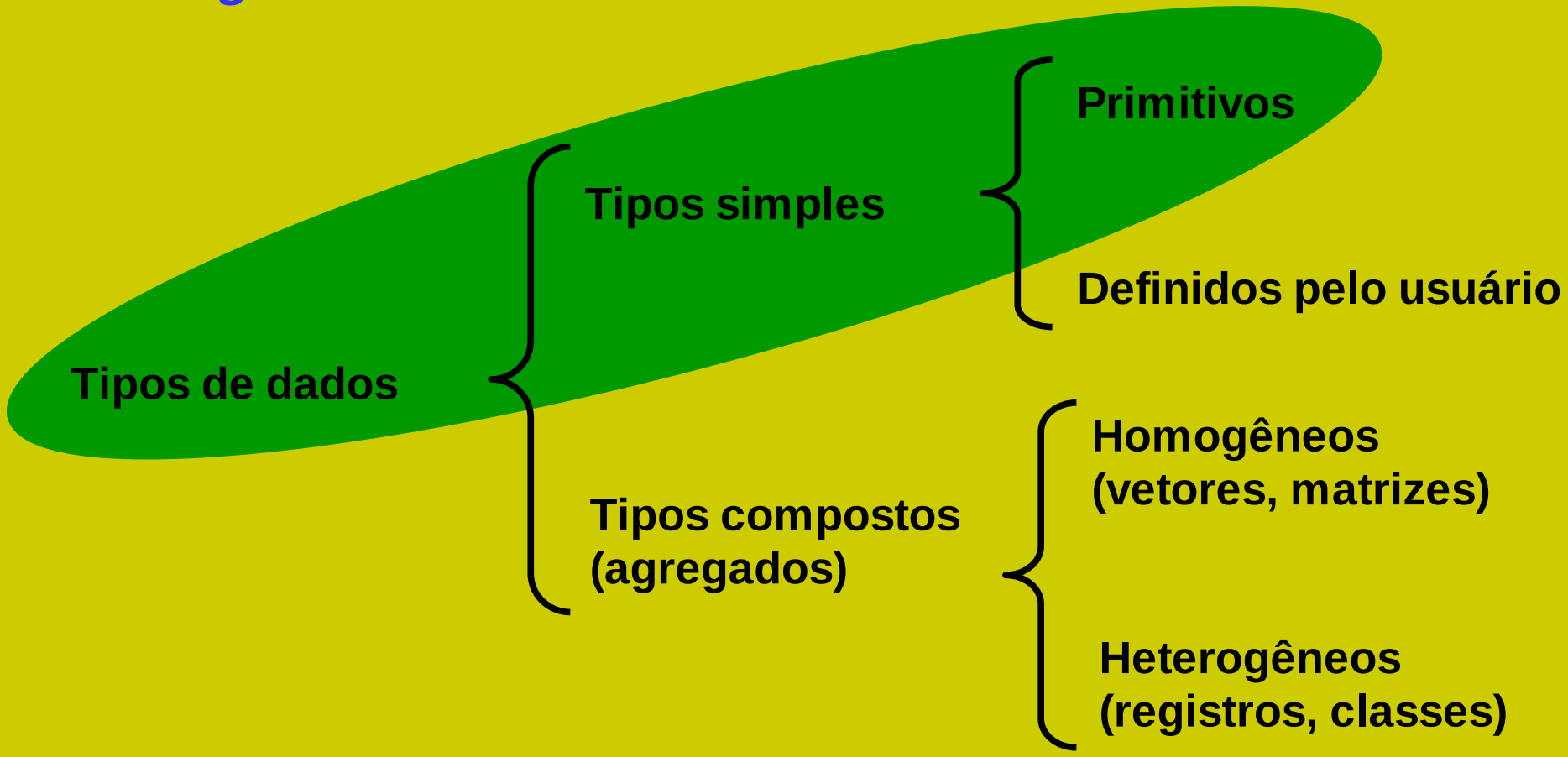
- **Conjunto de valores para "integer": todos os números inteiros entre min e máx;**
  Conjunto de valores para "float": números racionais entre min e máx, considerada uma certa precisão.

- **Conjunto de operações para "integer": conforme visto anteriormente;**
  Conjunto de operações para "float": idem.

- **Organização física para "integer": geralmente 1, 2 ou 4 bytes, com sinal (complemento de 1 ou de 2) ou sem sinal;**
  Organização física para "float": geralmente 4, 6 ou 8 bytes, com precisões e limites variados.

Taxonomia dos tipo de dados

Visão geral



Tipos primitivos de dados

Representam um único valor em cada instante de tempo.

- **Os tipos simples primitivos da linguagem C são: “int”, “float” e “char”;**
- **Os tipos simples primitivos da linguagem C: “int”, “float”, “char”, “short int”, “long int”, “unsigned int”, “double” etc (vide manual de referência);**
- **Outras linguagens possuem os tipos “boolean” e “string”, entre outros (Pascal, Java etc).**

Tipos primitivos de dados

Também chamados de ordinais.

- **Tipos primitivos da linguagem são aqueles que foram pré-definidos pelo projetista da mesma e estão à disposição dos programadores para serem usados diretamente.**

- **Normalmente representam abstrações ou modelos largamente utilizados de grandezas encontradas na vida real.**

- **Possíveis variações incluem mudanças nos valores min e máx, mudanças de precisão, exclusão do sinal etc.**

- **Toda linguagem especifica um conjunto de tipos (simples) primitivos de dados.**

Tipos Primitivos do C

Tipo	Tamanho	Intervalo		
unsigned char	8 bits	0	to	255
char	8 bits	-128	to	127
enum	16 bits	-32,768	to	32,767
unsigned int	16 bits	0	to	65,535
short int	16 bits	-32,768	to	32,767
int	16 bits	-32,768	to	32,767
unsigned long	32 bits	0	to	4,294,967,295
long	32 bits	-2,147,483,648	to	2,147,483,647
float	32 bits	3.4 * (10**-38)	to	3.4 * (10**+38)
double	64 bits	1.7 * (10**-308)	to	1.7 * (10**+308)
long double	80 bits	3.4 * (10**-4932)	to	1.1 * (10**+4932)