

TRILHA DE KOTLIN



venturus

DEVELOPING THE FUTURE

CAMPINAS TECH

←TALENTS!→

DADOS ALGORITMOS

REALIZAÇÃO:



CAMPINAS
TECH

novofuturo
tech

share^{rh}
Valor compartilhado
em recursos humanos

PATROCÍNIO:



venturus
DEVELOPING THE FUTURE

DADOS ALGORITMOS

Dados nada mais que é algo do mundo real que pode ser representado computacionalmente. Por exemplo, os números que pertencem ao conjunto dos números inteiros, os números que pertencem ao conjunto dos números reais, letras, caracteres especiais, acentuação, pontuação, palavras, etc.

TIPOS DE DADOS INTERNOS

Para que seja possível armazenar e manipular dados no computador é necessário representá-los internamente de alguma forma. Nós seres humanos, representamos nossos números usando um sistema que chamamos de sistema decimal (ou sistema na base 10).

Esse sistema, que se originou do fato de utilizarmos os 10 dedos das mãos para realizarmos nossas contas, possui 10 diferentes dígitos para representar as infinitas quantidades e valores que desejamos (0 1 2 3 4 5 6 7 8 e 9).

Nos caso dos computadores digitais, a notação que é utilizada possui apenas 2 algarismos ou dígitos para representar uma quantidade desejada, o **0** e o **1**. Esse sistema de representação é chamado de sistema binário (ou sistema na base 2) e utiliza a noção de ligado/desligado, ou verdadeiro/falso, ou finalmente 0/1 .

Pelo fato de um número precisar de muitos algarismos para ser expresso no sistema binário, outras formas de representação auxiliares também são utilizadas nos computadores, como por exemplo:

a **representação pelo sistema hexadecimal** (ou sistema na base 16)

que utiliza 16 dígitos (0 1 2 3 4 5 6 7 8 9 A B C D E F),

e a **representação no sistema octal** (ou sistema na base 8) que utiliza 8 dígitos (0 1 2 3 4 6 7 8).

Referência para entendimento

| Decimal Base-10 | Binary Base-2 | Octal Base-8 | Hexa Decimal Base-16 |
|--------------------|------------------|-----------------|-------------------------|
| 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| 2 | 10 | 2 | 2 |
| 3 | 11 | 3 | 3 |
| 4 | 100 | 4 | 4 |
| 5 | 101 | 5 | 5 |
| 6 | 110 | 6 | 6 |
| 7 | 111 | 7 | 7 |
| 8 | 1000 | 10 | 8 |
| 9 | 1001 | 11 | 9 |
| 10 | 1010 | 12 | A |
| 11 | 1011 | 13 | B |
| 12 | 1100 | 14 | C |
| 13 | 1101 | 15 | D |
| 14 | 1110 | 16 | E |
| 15 | 1111 | 17 | F |
| 16 | 10000 | 20 | 10 |

REALIZAÇÃO:

CAMPINAS
TECHnovofuturo
techshare^{rh}
Valor compartilhado
em recursos humanos

PATROCÍNIO:

venturus
DEVELOPING THE FUTURE

TIPOS DE DADOS EXTERNOS

O **ASCII** é um código que foi proposto por Robert W. Bemer como uma solução para unificar a representação de caracteres alfanuméricos em computadores. Antes de 1960 cada computador utilizava uma regra diferente para representar estes caracteres e o código ASCII nasceu para se tornar comum entre todas as máquinas.

TIPOS DE DADOS EXTERNOS

O nome **ASCII** vem do inglês ***American Standard Code for Information Interchange*** ou “**Código Padrão Americano para o Intercâmbio de Informação**”. Ele é baseado no alfabeto romano e sua função é padronizar a forma como os computadores representam letras, números, acentos, sinais diversos e alguns códigos de controle.

REALIZAÇÃO:

**CAMPINAS
TECH****novofuturo**
tech**share**^{rh}
Valor compartilhado
em recursos humanos

PATROCÍNIO:

**venturus**
DEVELOPING THE FUTURE

TIPOS DE DADOS EXTERNOS

No **ASCII** existem apenas **95 caracteres** que podem ser impressos, eles são numerados de **32 a 126** sendo os caracteres de **0 a 31 reservados para funções de controle**. Ou seja, funções de computador. Alguns caracteres acabaram caindo em desuso pois eram funções específicas para computadores da época como o Teletype (máquinas de escrever eletro-mecânicas), fitas de papel perfurado e impressoras de cilindro.

TIPOS DE DADOS EXTERNOS

Tabela ASCII

| ASCII control characters | | | ASCII printable characters | | | | | |
|--------------------------|------|-----------------------|----------------------------|-------|----|---|-----|---|
| 00 | NULL | (Null character) | 32 | space | 64 | @ | 96 | ` |
| 01 | SOH | (Start of Header) | 33 | ! | 65 | A | 97 | a |
| 02 | STX | (Start of Text) | 34 | " | 66 | B | 98 | b |
| 03 | ETX | (End of Text) | 35 | # | 67 | C | 99 | c |
| 04 | EOT | (End of Trans.) | 36 | \$ | 68 | D | 100 | d |
| 05 | ENQ | (Enquiry) | 37 | % | 69 | E | 101 | e |
| 06 | ACK | (Acknowledgement) | 38 | & | 70 | F | 102 | f |
| 07 | BEL | (Bell) | 39 | ' | 71 | G | 103 | g |
| 08 | BS | (Backspace) | 40 | (| 72 | H | 104 | h |
| 09 | HT | (Horizontal Tab) | 41 |) | 73 | I | 105 | i |
| 10 | LF | (Line feed) | 42 | * | 74 | J | 106 | j |
| 11 | VT | (Vertical Tab) | 43 | + | 75 | K | 107 | k |
| 12 | FF | (Form feed) | 44 | , | 76 | L | 108 | l |
| 13 | CR | (Carriage return) | 45 | - | 77 | M | 109 | m |
| 14 | SO | (Shift Out) | 46 | . | 78 | N | 110 | n |
| 15 | SI | (Shift In) | 47 | / | 79 | O | 111 | o |
| 16 | DLE | (Data link escape) | 48 | 0 | 80 | P | 112 | p |
| 17 | DC1 | (Device control 1) | 49 | 1 | 81 | Q | 113 | q |
| 18 | DC2 | (Device control 2) | 50 | 2 | 82 | R | 114 | r |
| 19 | DC3 | (Device control 3) | 51 | 3 | 83 | S | 115 | s |
| 20 | DC4 | (Device control 4) | 52 | 4 | 84 | T | 116 | t |
| 21 | NAK | (Negative acknowl.) | 53 | 5 | 85 | U | 117 | u |
| 22 | SYN | (Synchronous idle) | 54 | 6 | 86 | V | 118 | v |
| 23 | ETB | (End of trans. block) | 55 | 7 | 87 | W | 119 | w |
| 24 | CAN | (Cancel) | 56 | 8 | 88 | X | 120 | x |
| 25 | EM | (End of medium) | 57 | 9 | 89 | Y | 121 | y |
| 26 | SUB | (Substitute) | 58 | : | 90 | Z | 122 | z |
| 27 | ESC | (Escape) | 59 | ; | 91 | [| 123 | { |
| 28 | FS | (File separator) | 60 | < | 92 | \ | 124 | |
| 29 | GS | (Group separator) | 61 | = | 93 |] | 125 | } |
| 30 | RS | (Record separator) | 62 | > | 94 | ^ | 126 | ~ |
| 31 | US | (Unit separator) | 63 | ? | 95 | _ | | |
| 127 | DEL | (Delete) | | | | | | |

TIPOS DE DADOS EXTERNOS

O “a” minúsculo que você digita no teclado é representado na tabela pelo decimal **97**, mas internamente o computador trabalha isso em binário, ou seja, com 8 bits **01100001** (97 em decimal).

Faça esse teste: crie um arquivo de texto e coloque nele apenas a letra **a**. Depois de salvá-lo, o seu sistema operacional vai reconhecer que ele ocupa apenas **1 byte (8 bits)** :

TIPOS DE DADOS EXTERNOS

Na década de 60 notaram a necessidade de normalizar a tabela para usar os 8 bits disponíveis (**1 byte**) , que era o sistema mais usado na época. Criaram uma versão estendida da **ASCII** e então ela passou representar até **256 caracteres**. Além dos caracteres da versão original, a estendida dispõe de alguns novos símbolos e caracteres gráficos:

TIPOS DE DADOS EXTERNOS

Tabela Extendida
ASCII

| | | | | | | | |
|-----|---|-----|---|-----|-----|-----|------|
| 128 | Ç | 160 | à | 192 | Ł | 224 | ó |
| 129 | ù | 161 | í | 193 | ł | 225 | ô |
| 130 | é | 162 | ó | 194 | Ł | 226 | õ |
| 131 | â | 163 | ú | 195 | ł | 227 | ö |
| 132 | ä | 164 | ñ | 196 | — | 228 | ø |
| 133 | å | 165 | Ñ | 197 | + ä | 229 | Ō |
| 134 | â | 166 | ª | 198 | + ä | 230 | μ |
| 135 | ç | 167 | º | 199 | Ä | 231 | þ |
| 136 | ê | 168 | ¿ | 200 | ℓ | 232 | þ |
| 137 | ë | 169 | ® | 201 | ℓ | 233 | Ů |
| 138 | è | 170 | ¬ | 202 | ℓ | 234 | Ů |
| 139 | ï | 171 | ½ | 203 | ℓ | 235 | Ů |
| 140 | ï | 172 | ¼ | 204 | ℓ | 236 | ý |
| 141 | ì | 173 | í | 205 | = | 237 | Ÿ |
| 142 | Ä | 174 | « | 206 | ℓ | 238 | — |
| 143 | Å | 175 | » | 207 | □ | 239 | · |
| 144 | É | 176 | — | 208 | ø | 240 | ≡ |
| 145 | æ | 177 | — | 209 | ø | 241 | ± |
| 146 | Æ | 178 | ■ | 210 | É | 242 | ≡ |
| 147 | ó | 179 | — | 211 | É | 243 | ¼ |
| 148 | ö | 180 | — | 212 | É | 244 | ¶ |
| 149 | ö | 181 | Ä | 213 | í | 245 | § |
| 150 | ü | 182 | Ä | 214 | í | 246 | ÷ |
| 151 | ü | 183 | Ä | 215 | í | 247 | · |
| 152 | ÿ | 184 | © | 216 | í | 248 | · |
| 153 | Ö | 185 | — | 217 | — | 249 | — |
| 154 | Ü | 186 | — | 218 | — | 250 | · |
| 155 | ø | 187 | — | 219 | — | 251 | · |
| 156 | £ | 188 | — | 220 | — | 252 | · |
| 157 | Ø | 189 | — | 221 | — | 253 | · |
| 158 | x | 190 | — | 222 | — | 254 | · |
| 159 | f | 191 | — | 223 | — | 255 | nbsp |

REALIZAÇÃO:

CAMPINAS
TECHnovofuturo
techshare^{rh}
Valor compartilhado
em recursos humanos

PATROCÍNIO:

venturus
DEVELOPING THE FUTURE

TIPOS DE DADOS EXTERNOS

O problema, na tabela **ASCII** é que ela **não consegue representar os símbolos de outras línguas** como a hebraica, árábica, grega, japonesa

REALIZAÇÃO:

CAMPINAS
TECHnovofuturo
techshare^{rh}
Valor compartilhado
em recursos humanos

PATROCÍNIO:

venturus
DEVELOPING THE FUTURE

TIPOS DE DADOS EXTERNOS

Foi então que no início dos anos 90 surgiu o **Unicode**, bem no início da internet, momento crucial em que a informática passou a necessitar de um padrão universal de representação de texto, para que o tráfego das informações ficasse inteligível em todos os computadores, não importando o idioma de escrita.

REALIZAÇÃO:

CAMPINAS
TECHnovofuturo
techshare^{rh}
Valor compartilhado
em recursos humanos

PATROCÍNIO:

venturus
DEVELOPING THE FUTURE

TIPOS DE DADOS EXTERNOS

O **Unicode** faz parte do ISO 10646 e é o padrão de representação de texto mais usado. *Uni* em latim significa **um**, Unicode remete à ideia de um **padrão universal de codificação de caracteres**.

TIPOS DE DADOS EXTERNOS

Ele foi projetado com base no código ASCII, com a diferença que ele é capaz de codificar quase todos os caracteres e símbolos presentes nas mais variadas línguas do mundo ele consegue representar. Existem diversos sistemas de codificação Unicode e versão(junho 2020) eles conseguem representar mais de 143 mil caracteres. Unicode nos permite usar 8, 16 e até 32 bits (**mais de 2 bilhões de caracteres possíveis**).

REALIZAÇÃO:

CAMPINAS
TECHnovofuturo
techshare^{rh}
Valor compartilhado
em recursos humanos

PATROCÍNIO:

venturus
DEVELOPING THE FUTURE

TIPOS DE DADOS EXTERNOS

Quando salvar um caracter com formato unicode:

Ele vai ocupar 4 bytes, ou seja, estará usando 32 bits.

Unicode mapeia cada caractere para um código específico chamado de **code point** que tem o formato

U+<código hexadecimal>

numa faixa de **U+0000** a **U+10FFFF**.

TIPOS DE DADOS EXTERNOS

| Unicode | ASCII (decimal) | Caractere |
|---------|-----------------|-----------|
| U+0021 | 33 | ! |
| U+0024 | 36 | \$ |
| U+0026 | 38 | & |
| U+002A | 42 | * |

TIPOS DE DADOS EXTERNOS

UTF-8

O *encoding* da família Unicode mais amplamente usado e difundido em sistemas operacionais, web e linguagens de programação é o UTF-8, que foi co-criado no início dos anos 90 Rob Pike (Desenvolvedor de linguagem).

Além disso, UTF-8 é o encoding padrão adotado pela W3C para a Web e ele é 100% compatível com a tabela ASCII.

TIPOS DE DADOS EXTERNOS

UTF-8

Ele é um padrão de encoding variável e pode usar de 1 até 4 bytes, a depender do caractere que se está representando. Ele resolve o problema de não se fixar 32 bits para todos os caracteres, o que naturalmente faria um texto escrito em inglês/Português ocupar um espaço muito maior do que ele poderia ocupar.

TIPOS DE DADOS EXTERNOS

UTF-8

No seu HTML quando você indica:

<meta charset="utf-8">

Você está dizendo pro navegador que sua página utiliza esse encoding.

TIPOS DE DADOS EXTERNOS

UTF-8

Nas IDEs (**integrated development environment**) já vem por padrão configurado com UTF-8

Verificar no INTELLIJ

TIPOS DE DADOS PRIMITIVOS

Tratando-se de tipos de dados primitivos, são constatados quatro formatos:

- Texto
- Inteiro
- Real
- lógico

TEXTO

Os dados do tipo texto representam uma sequência de um ou mais caracteres. (0...9), alfabéticos (a...z,A...Z) e especiais (! @ # \$ % & *)

Normalmente são colocados entre aspas duplas ("").

Exemplos de dado do tipo texto:

Nome (Marcos)

Cidade (Santo André)

Obs: espaços também contam como caracteres.

INTEIRO

São representados por valores numéricos, tanto negativos, quando positivos (sem casas decimais).

Exemplos de dado do tipo inteiro:

Idade (18)

Materia (5)

Pote (3)

...

REAL

Os dados do tipo real são valores numéricos, tanto positivos, quanto negativos que utilizam casas decimais,

Exemplos, podemos citar:

Salário (R\$1.100,00)

Valor Produto(R\$ 350,00)

Valor Total Circulo(8,3565)

LÓGICO

Por último, mas não menos importante, o tipo lógico. Este tipo de dado normalmente é representado por alternativas:

SIM ou **NÃO**, **VERDADEIRO** ou **FALSO**.

Dados do tipo lógico também podem ser chamados de Booleano.

Exemplo de dado lógico que é frequentemente utilizado em programação é:
Genero (Masculino ou Feminino)

Produto Vendido (verdadeiro/ falso ou sim/nao)

VARIÁVEIS E CONSTANTES

Dentro de um algoritmo podemos encontrar basicamente duas classes diferentes de dados, os dados constantes e os variáveis.

Variável quando seu valor se altera conforme a necessidade da lógica do algoritmo.

VARIÁVEIS E CONSTANTES

Constante quando seu valor não se altera ao longo do tempo em que o algoritmo é executado, ou seja, permanece o mesmo desde o início até o final da execução. Já um dado que pode ter seu valor alterado durante a execução do programa é tido como uma variável. Por exemplo, no cálculo da área de uma circunferência ($A = \pi r^2$), o valor de π é constante, pois é sempre igual a 3.1416..., e o raio r é variável, pois pode assumir diferentes valores a cada cálculo da área. Seguindo a mesma lógica, a área A calculada para diferentes r também é variável.

var marca, modelo: **caractere**;
ano: **inteiro**;
preco: **real**;
vendido: **lógico**;

PI: **real**;
MAXIMO: **inteiro**;
TECLA: **caractere**

| REGRA | EXEMPLO |
|---|---|
| Inicie sempre por um caractere alfabético, nunca por um número. | Nome (correto) - 1nome (errado) |
| Não utilize caracteres especiais como “, () / *; +. | Nome (M); N*B |
| Não coloque espaços em branco ou hífen entre nomes. | salario-bruto |
| Utilize, se necessário, <i>underline</i> | salario_bruto |
| Crie suas variáveis com nomes sugestivos. | Se vai guardar salário de funcionários, dê à variável o nome <i>salario</i> . |

ATRIBUIÇÃO

Após realizada a declaração de um identificador, é possível iniciar a manipulação dos dados que esse identificador irá representar a partir da atribuição de valores ao mesmo. Esse processo de atribuir ou alterar o valor dos dados de um identificador é chamado de atribuição e é representado pelo símbolo \leftarrow quando estivermos trabalhando com identificadores variáveis, e pelo símbolo $=$ quando estivermos trabalhando com identificadores constantes. A atribuição de valores a uma variável é feita da seguinte forma:

ATRIBUIÇÃO

<identificador da variável> ← <valor do mesmo tipo da variável>

<identificador da variável> ← <operações cujo resultado é o mesmo da variável>

REALIZAÇÃO:

CAMPINAS
TECHnovofuturo
techshare^{rh}
Valor compartilhado
em recursos humanos

PATROCÍNIO:

venturus
DEVELOPING THE FUTURE

Como pode ser visto acima, no lado esquerdo do operador \leftarrow será colocado o nome da variável que irá receber o valor, e do lado direito o valor que será armazenado na mesma. A seguir são apresentados alguns exemplos de atribuições de valores a variáveis:

TipoVeiculo \leftarrow motocicleta;

Aceleracao \leftarrow 15.52;

Massa \leftarrow 12.3;

Forca \leftarrow Massa * Aceleracao;

Usado \leftarrow F;

Uma variável pode armazenar apenas um único valor por vez, sendo que sempre que um novo valor é atribuído a variável o valor anterior que estava armazenado na mesma é perdido. Por exemplo, consideremos o algoritmo a seguir:

Algoritmo Atribuições de valores a uma variável.

var numero: **inteiro**

numero \leftarrow 222

numero \leftarrow 1000

numero \leftarrow 23

CAMPINAS TECH

←TALENTS→

REALIZAÇÃO:



CAMPINAS
TECH

novofuturo
tech

share^{rh}
Valor compartilhado
em recursos humanos

PATROCÍNIO:



venturus
DEVELOPING THE FUTURE