

Algoritmos

Introdução a Computação

Prof. Hitoshi Nagano, Ph.D.

Aula 2

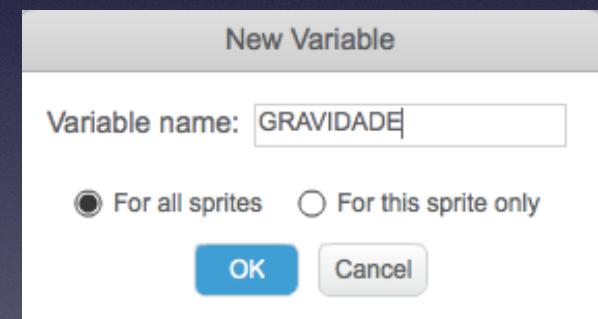
Variáveis & Tipos

Variáveis

- **nome** para uma posição de memoria



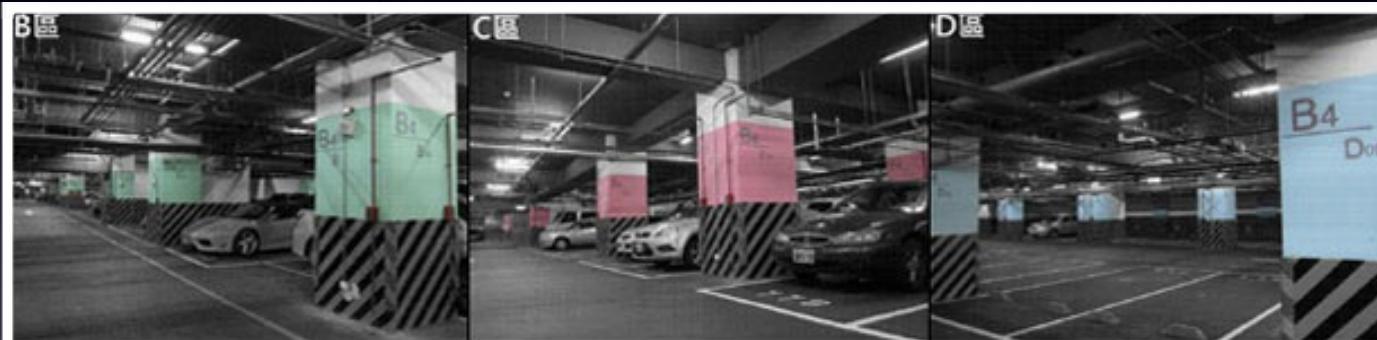
- Requer uma declaração



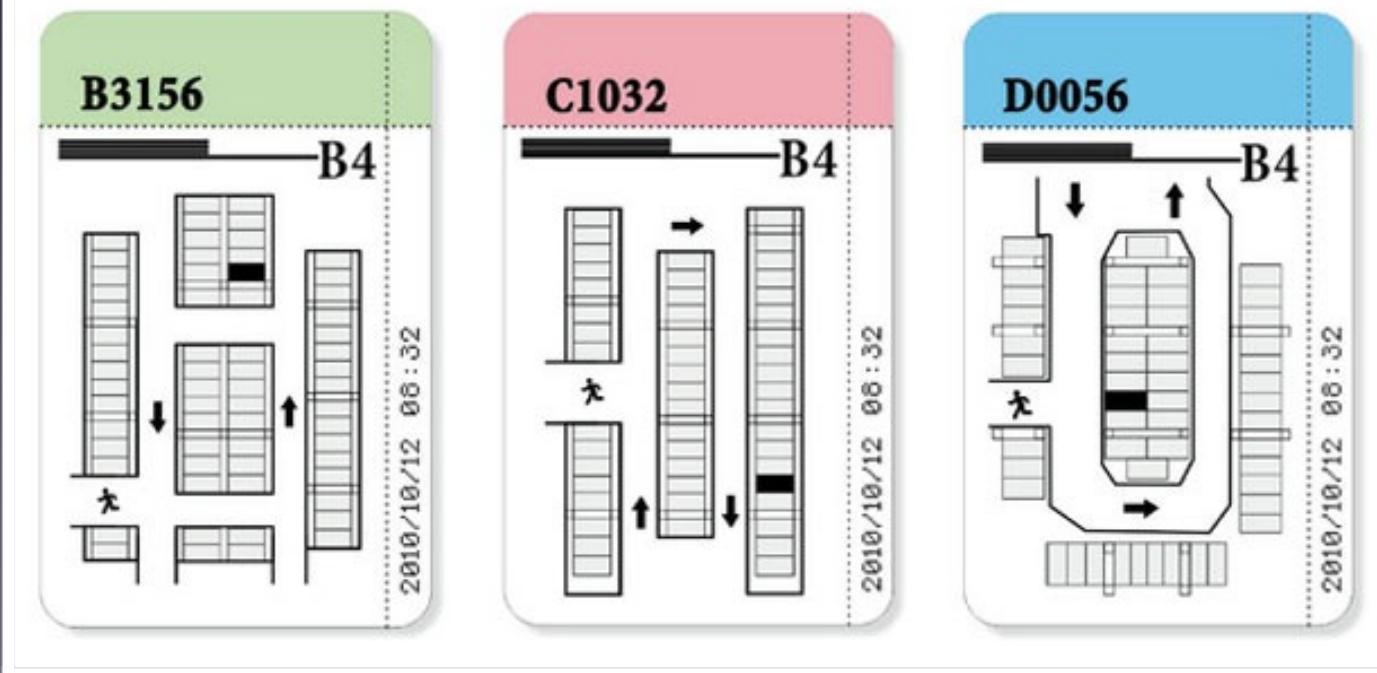
- atribuição e acesso pelo **nome**



Variáveis



posição de
memória: **vaga**



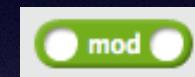
nome da
variável: **B3156**

conteúdo: **carro**

Operadores

Operadores

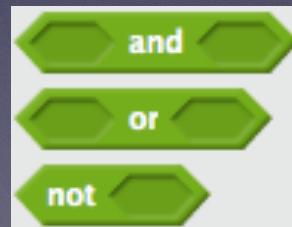
- operadores aritméticos



- operadores de comparação (relação)



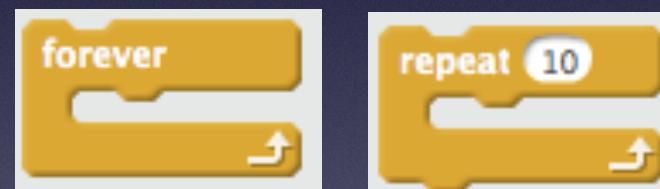
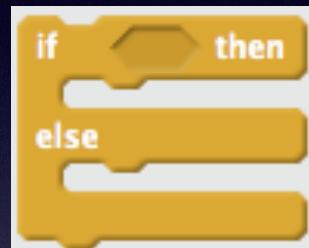
- operadores lógicos



Estrutura

Estrutura

- condicional
- loops
- funções



loop

```
#include <stdio.h>

int main()
{
    while (1)
    {
        printf("Olá mundo!\n");
    }
}
```



loop

```
#include <stdio.h>

int main()
{
    for (int i = 0; i < 10; i++)
    {
        printf("Ola mundo!\n");
    }
}
```



variáveis

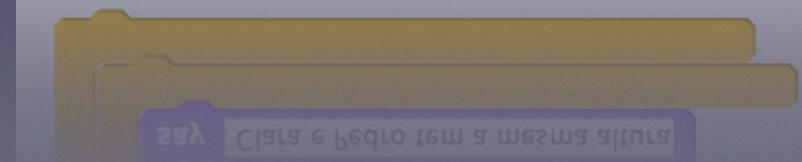
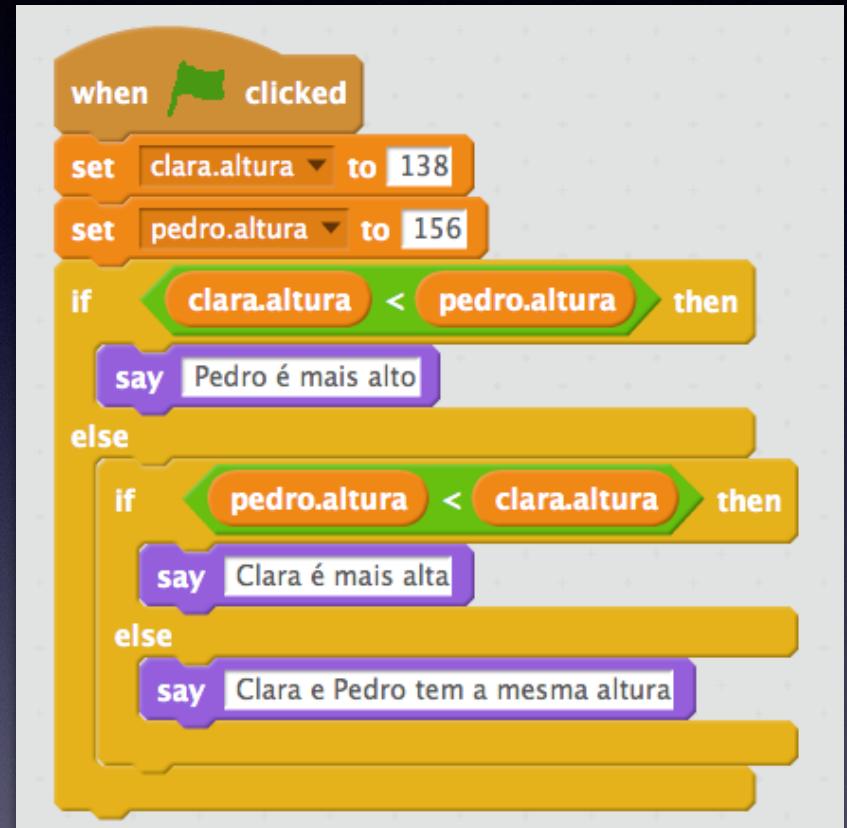
```
#include <stdio.h>

int main()
{
    int contador = 0;
    while (1)
    {
        printf("%d\n", contador);
        contador++;
    }
}
```

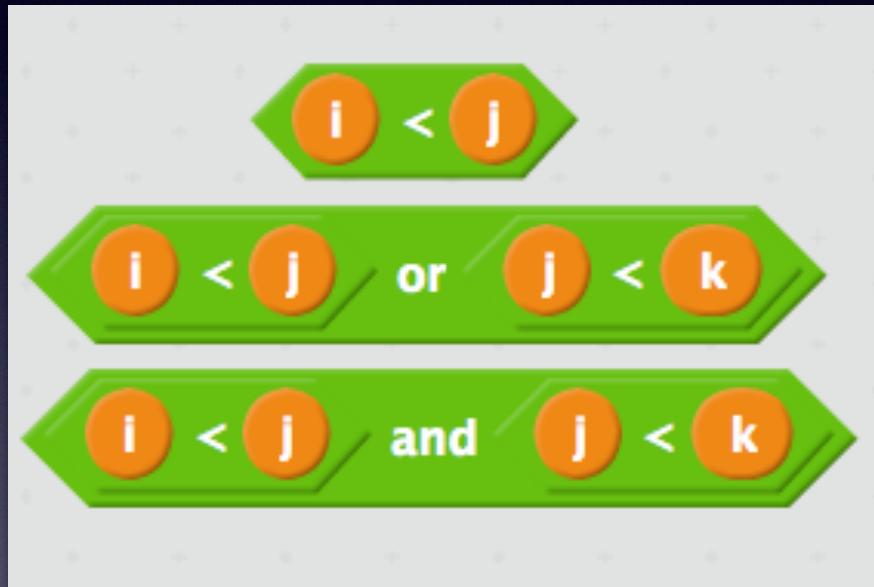


condições

```
int claraAltura = 138;  
int pedroAltura = 156;  
  
if (claraAltura < pedroAltura)  
{  
    printf("Pedro eh mais alto\n");  
}  
  
else if (pedroAltura < claraAltura)  
{  
    printf("Clara eh mais alta\n");  
}  
  
else  
{  
    printf("Clara e Pedro tem a mesma altura\n");  
}
```



expressões booleanas

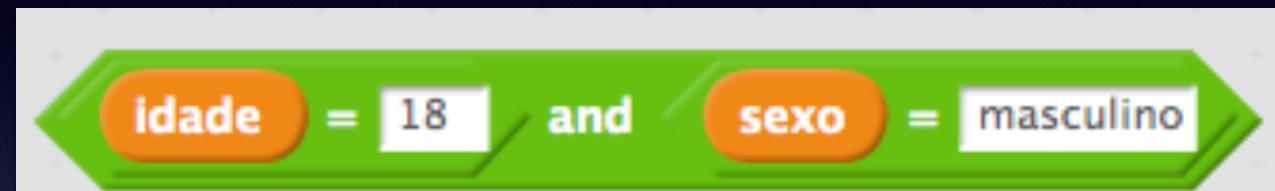


`(i < j)`

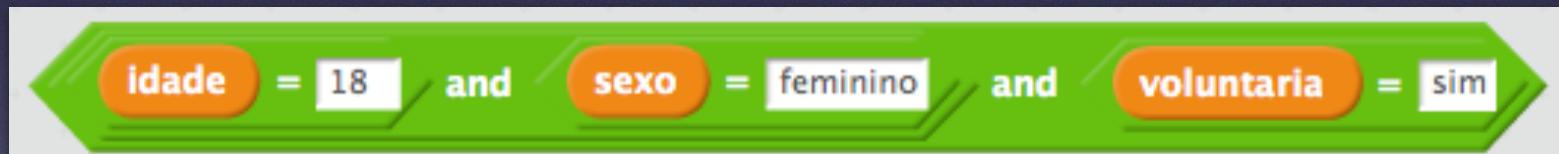
`((i < j) || (j < k))`

`((i < j) && (j < k))`

explique...



```
idade = 18 and sexo = masculino
```



```
idade = 18 and sexo = feminino and voluntaria = sim
```



```
idade = 18 and sexo = masculino or idade = 18 and sexo = feminino and voluntaria = sim
```

Algebra Booleana

TBD

Linguagem C

Implementação



Interpretada

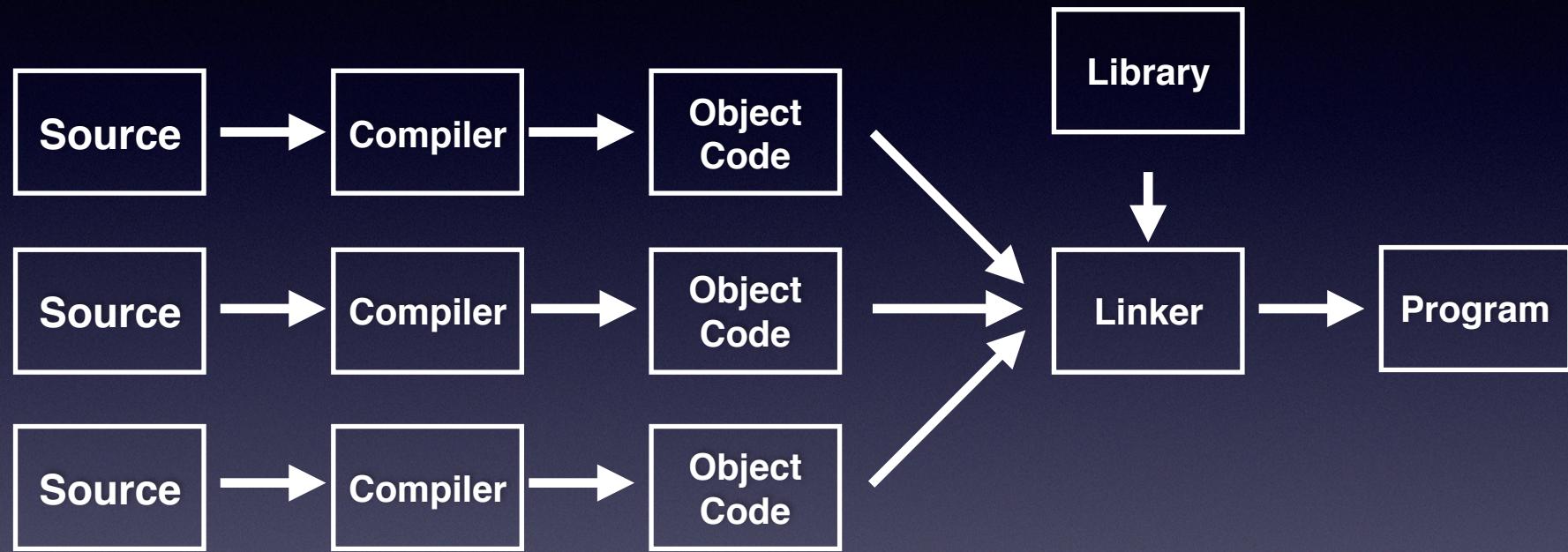
C

Compilada

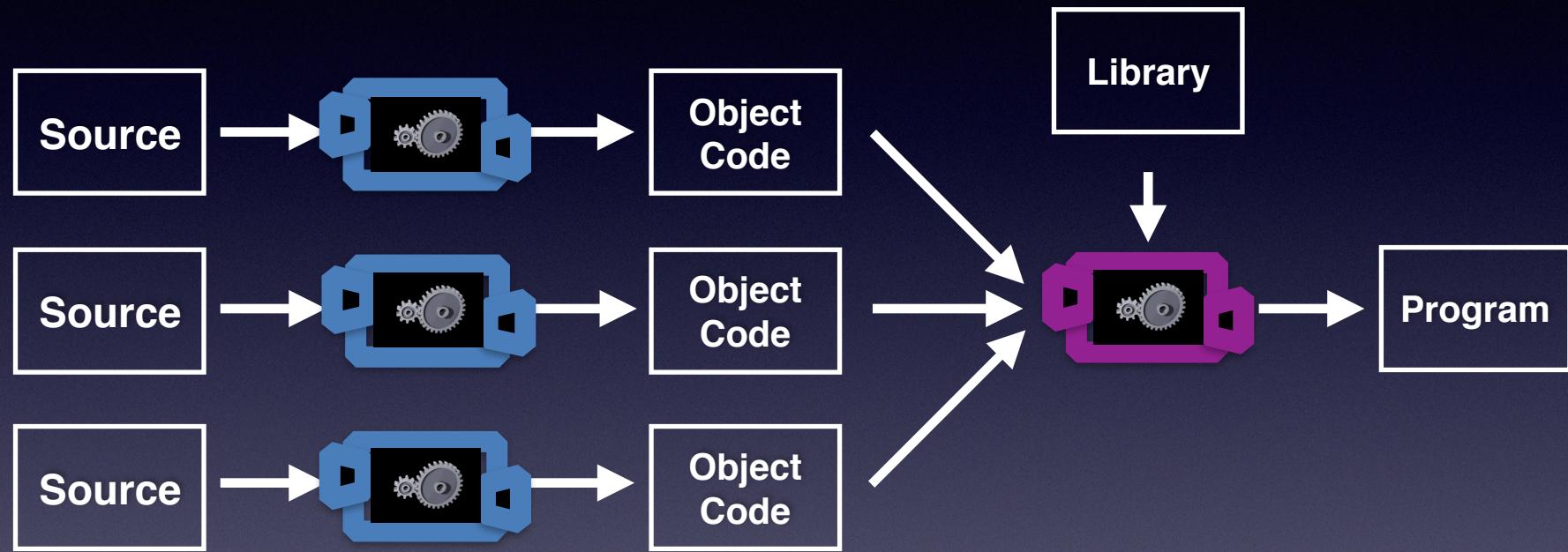
C



C



C



Source

(Código Fonte)

```
#include <stdio.h>

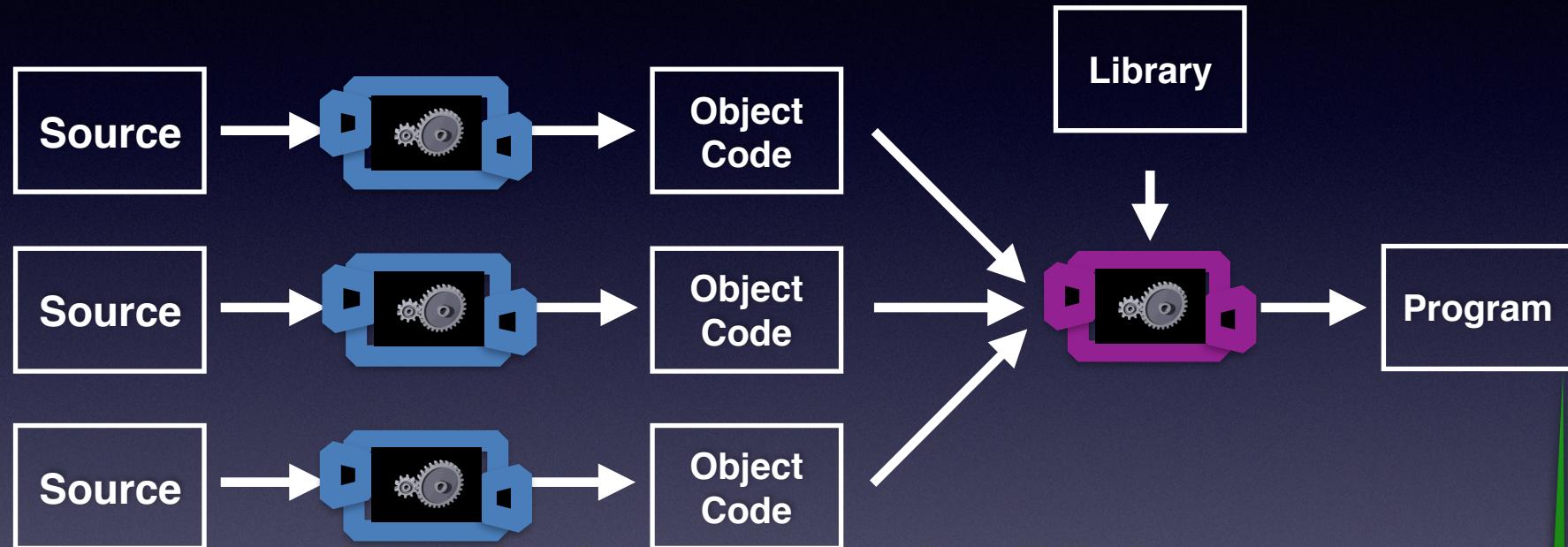
int main()
{
    printf("ola mundo!\n");
    return 0;
}
```

Program

(Executável)

```
0000000: 01111111 01000101 01001100 01000110 00000010 00000001 .ELF..  
0000006: 00000001 00000000 00000000 00000000 00000000 00000000 .....  
000000c: 00000000 00000000 00000000 00000000 00000010 00000000 .....  
0000012: 00111110 00000000 00000001 00000000 00000000 00000000 >....  
0000018: 01000000 00000100 01000000 00000000 00000000 00000000 @. @...  
000001e: 00000000 00000000 01000000 00000000 00000000 00000000 .. @...  
0000024: 00000000 00000000 00000000 00000000 11001000 00010001 .....  
000002a: 00000000 00000000 00000000 00000000 00000000 00000000 .....  
0000030: 00000000 00000000 00000000 00000000 01000000 00000000 .... @.  
0000036: 00111000 00000000 00001001 00000000 01000000 00000000 8... @.
```

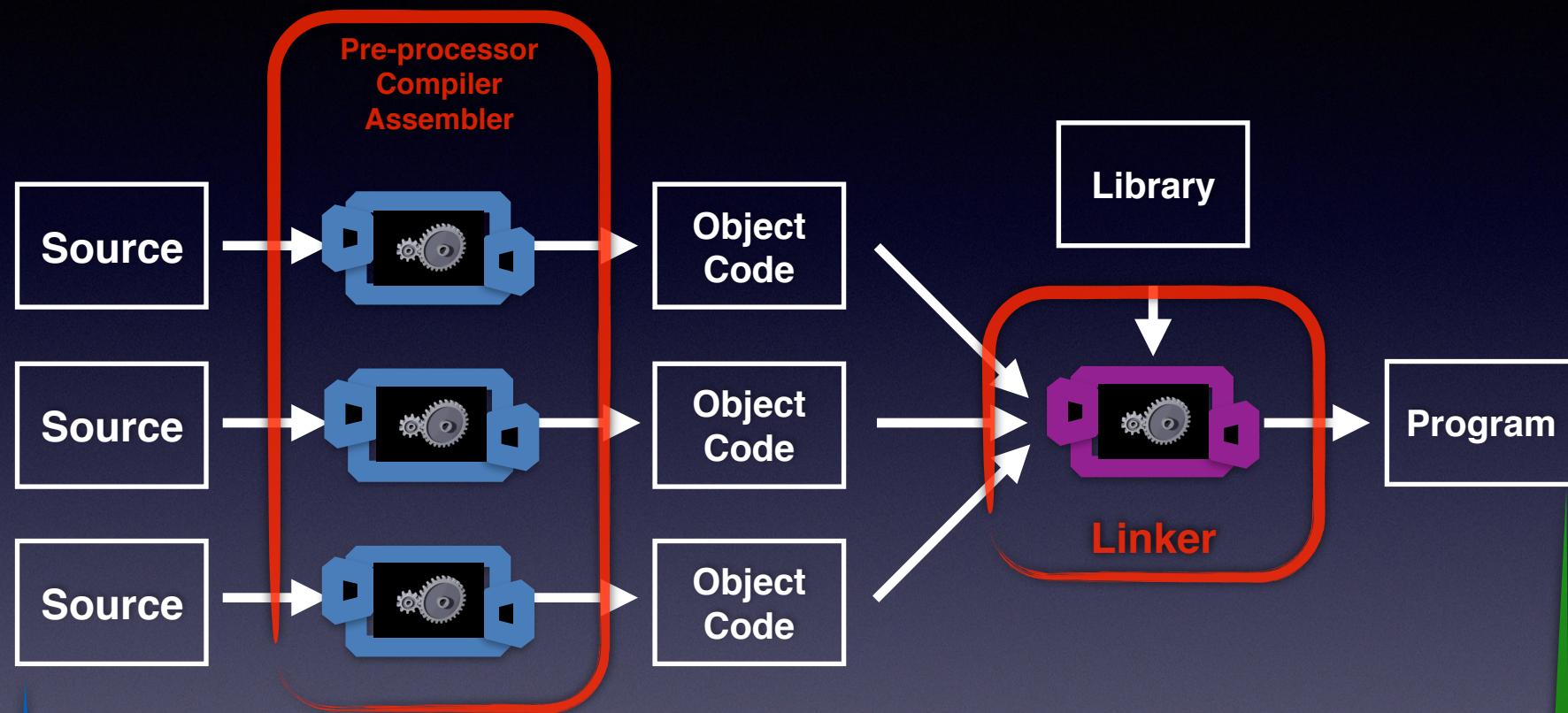
C



```
#include <stdio.h>
int main()
{
    printf("ola mundo!\n");
}
```

000000: 01111111 01000101 01001100 01000110 00000010 00000001 .ELF..
000006: 00000001 00000000 00000000 00000000 00000000 00000000
00000c: 00000000 00000000 00000000 00000000 00000010 00000000
000012: 00111110 00000000 00000001 00000000 00000000 00000000 >.....
000018: 01000000 00000100 01000000 00000000 00000000 00000000 @.@@...
00001e: 00000000 00000000 01000000 00000000 00000000 00000000 ..@...
000024: 00000000 00000000 00000000 00000000 11001000 00010001
00002a: 00000000 00000000 00000000 00000000 00000000 00000000
000030: 00000000 00000000 00000000 00000000 01000000 00000000@.
000036: 00111000 00000000 00001001 00000000 01000000 00000000 8...@.

C



```
#include <stdio.h>
int main()
{
    printf("ola mundo!\n");
}
```

0000000: 01111111 01000101 01001100 01000110 00000010 00000001 .ELF..
0000006: 00000001 00000000 00000000 00000000 00000000 00000000
000000c: 00000000 00000000 00000000 00000000 00000010 00000000
0000012: 00111110 00000000 00000001 00000000 00000000 00000000 >.....
0000018: 01000000 00000100 01000000 00000000 00000000 00000000 @.@@...
000001e: 00000000 00000000 01000000 00000000 00000000 00000000 ..@...
0000024: 00000000 00000000 00000000 00000000 11001000 00010001
000002a: 00000000 00000000 00000000 00000000 00000000 00000000
0000030: 00000000 00000000 00000000 00000000 01000000 00000000@.
0000036: 00111000 00000000 00001001 00000000 01000000 00000000 8...@.

C

- Executa em qualquer coisa, controle total sobre a máquina
- Exige poucos recursos
- Impõe poucas restrições
- Sintaxe similar a outras linguagens, php, Java
- + Rigor e Disciplina
- Por outro lado: pouca flexibilidade, orientação a objetos (OO), ...

Livros sobre C

- http://publications.gbdirect.co.uk/c_book/
- <http://infolab.stanford.edu/~ullman/focs.html>

Variáveis

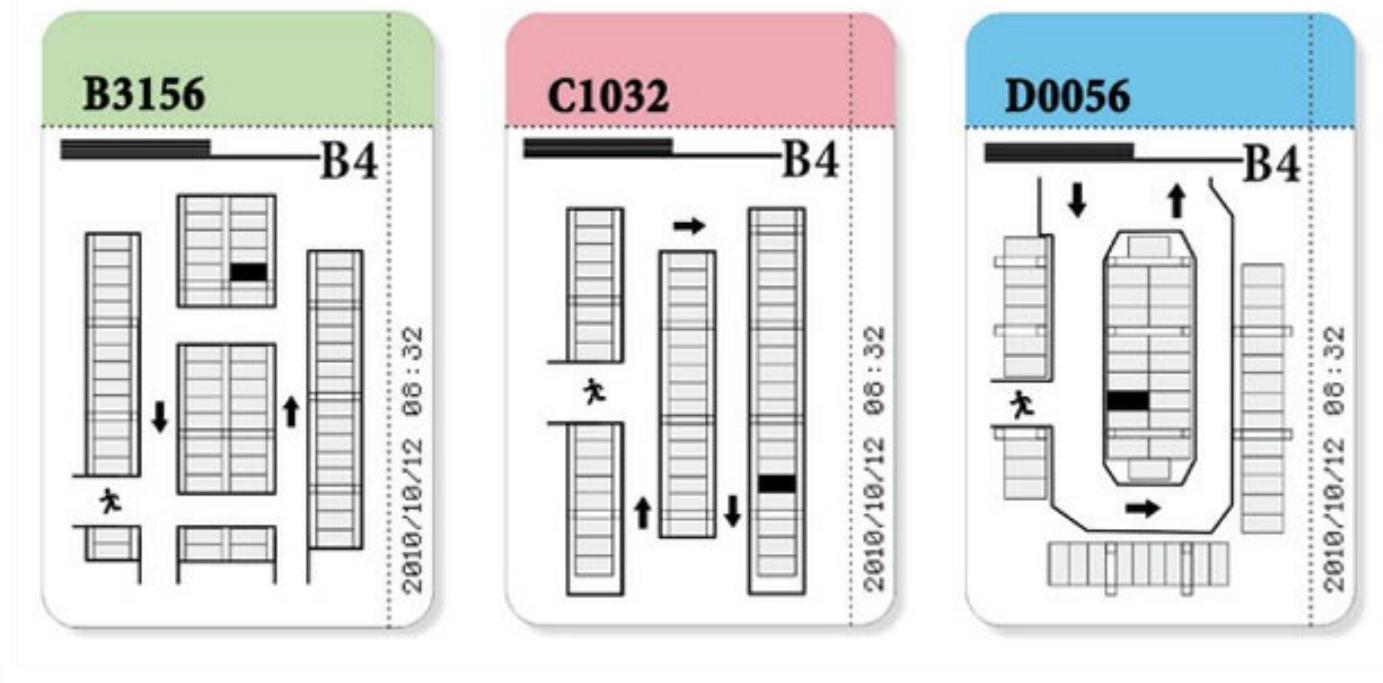
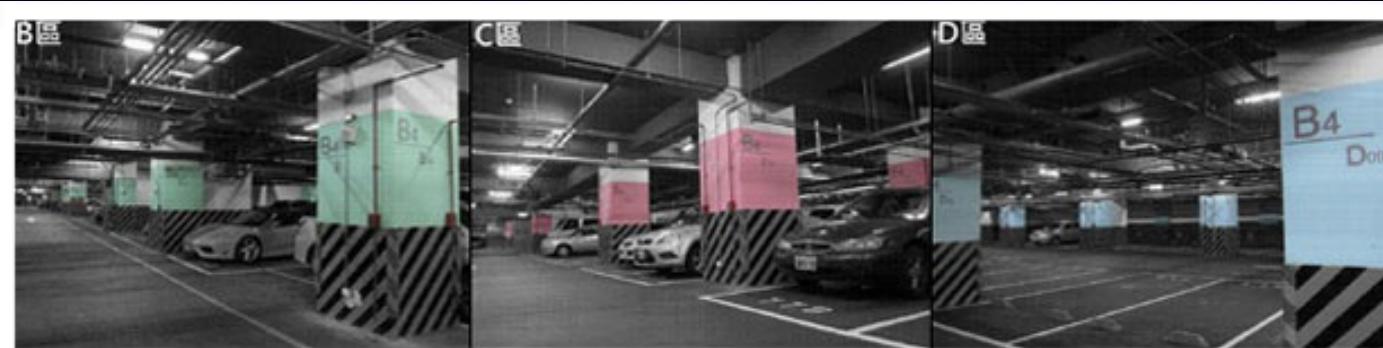
- **nome** para uma posição de memoria
- Requer uma declaração
- Atribuição e acesso pelo **nome**

Tipos em C

- char
- integer
- float
- bool

Tipos de Variáveis

(Variable Types)



posição de
memória: **vaga**

nome da
variável: **B3156**

conteúdo: **carro**

Tipo: **veículo
motorizado de
4 rodas**

Tipos de Variáveis

(Variable Types)

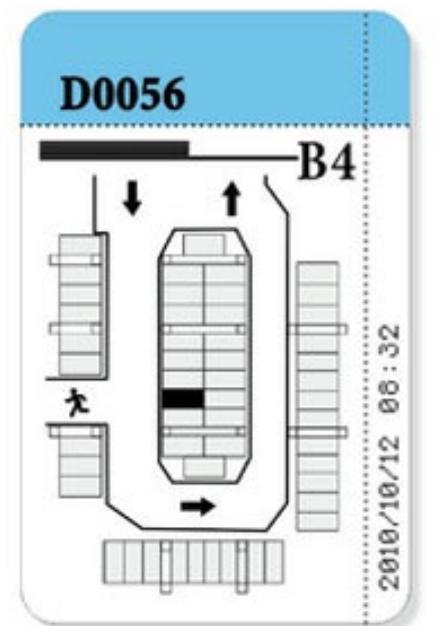
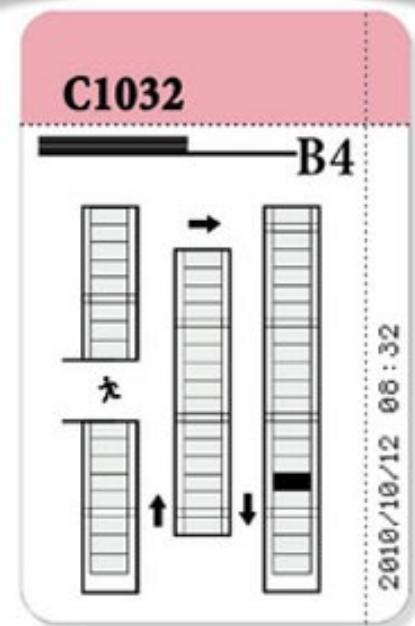
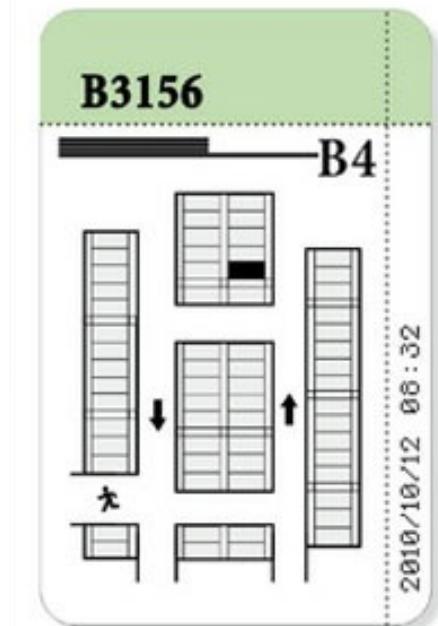


posição de
memória: **vaga**

nome da
variável: **C1032**

conteúdo: **moto**

Tipo: **veículo
motorizado de
2 rodas**



Tipos de Variáveis

(Variable Types)

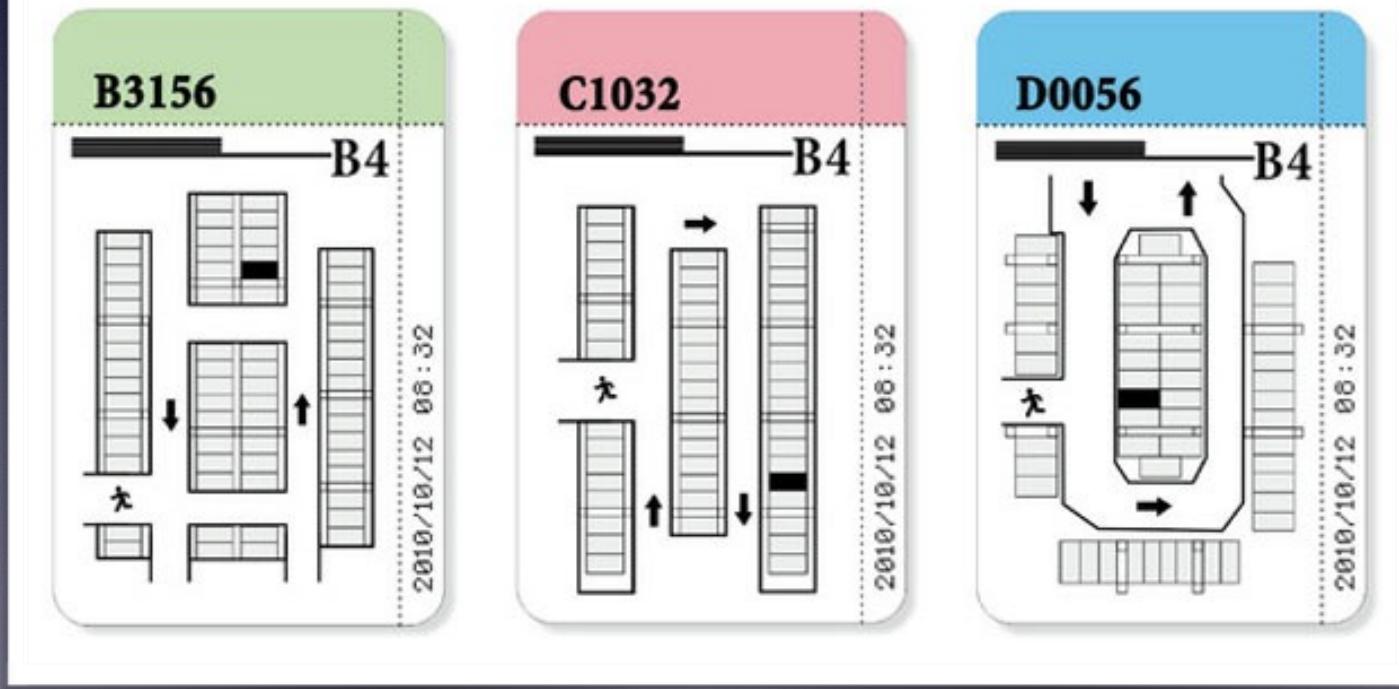


posição de
memória: **vaga**

nome da variável:
D0056

conteúdo:
bicicleta

Tipo: **veículo**
propulsão humana
2 rodas



Qual é o Tipo da Variável?



Tipos em C

- char 1 byte
- integer 4 bytes
- float 4 bytes
- bool 1 byte
- long 8 bytes
- double 8 bytes

ASCII

| | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|
| A | B | C | D | E | F | G | H | I | J | K | L | M |
| 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 |

| | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|
| N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| 78 | 79 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 |

| 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 | @ | Ø | 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.comSource: www.LookupTables.com

| | | | | | | | | | | | | | | | | | | |
|----|----|-----|-------------------------------|----|----|-----|-------|-------------|----|----|-----|-------|----------|-----|----|-----|-------|-------------------------|
| 3T | TE | 033 | DS2 (data separator) | 63 | 3E | 033 | ? | 3 | 62 | 2L | T33 | > | - | TS1 | 3L | T33 | > | TS1 : DEF |
| 30 | TE | 030 | RS2 (record separator) | 65 | 3E | 030 | ? | > | 64 | 2E | T30 | @ | - | TSE | 3E | T30 | @ | TSE : |
| 5A | TD | 032 | C2 (group separator) | 67 | 3D | 032 | C | = | 63 | 2D | T32 | ? |] | TS2 | 3D | T32 | ? | } TS2 : |
| 5B | TC | 034 | FS2 (file separator) | 68 | 3C | 034 | D | < | 64 | 2C | T34 | @ | / | TS3 | 3C | T34 | @ | TS3 : |

Inteiros

In the early days of computing, designers made computers express numbers using **unsigned binary**.

And they were content...

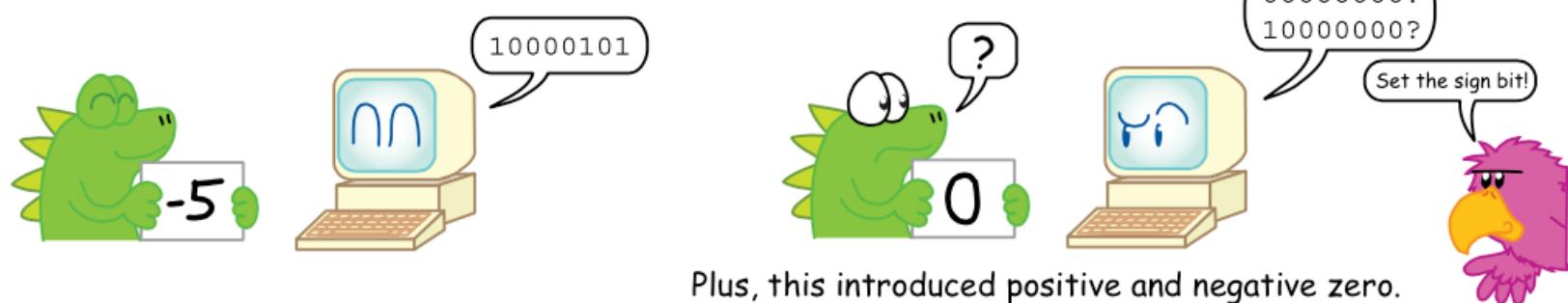
Until there were negative numbers.



To include negative numbers, designers came up with **sign magnitude**.

That took care of the negative numbers...

But the computer had to count backwards
for the negative numbers.



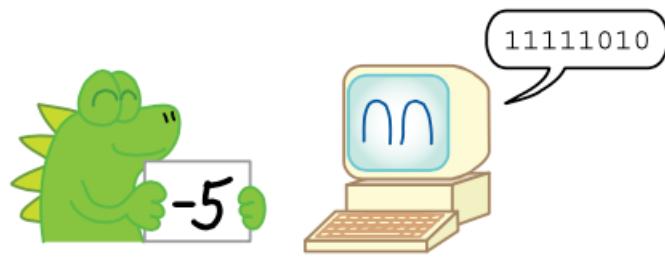
Plus, this introduced positive and negative zero.

Arte: Katrina Yim

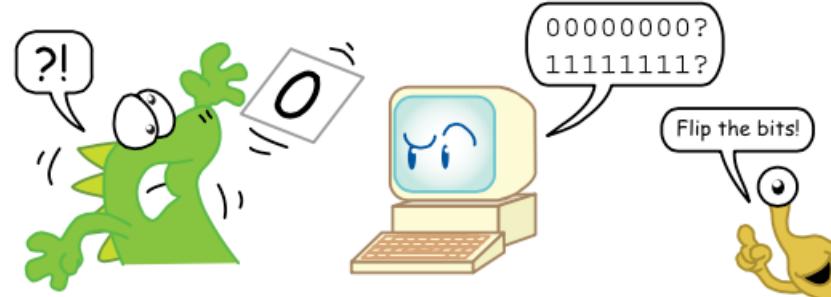
Inteiros

Then designers created **one's complement**.

Now computers only had to count
in one direction...



But there were still two zeroes!



Finally, designers developed **two's complement**.

Now, there was only one zero...



And they were content.



Katrina Yim

KatrinaYim

Linux: 3 comandos básicos

para criar, compilar e rodar programas

- gedit
- clang ou gcc
- ./<nome do executável>

Linux: 3 comandos básicos

para criar, compilar e rodar programas

EXEMPLO:

```
$ gedit hello.c
```

```
$ gcc hello.c -o hello
```

```
$ ./hello
```

História Linguagens