CC50

O CURSO DE HARVARD. NO BRASIL.

Uma história de amor de Cookies

```
when [green flag] clicked
say [O hai, world!]
```
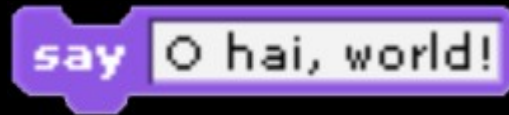
```c
#include <stdio.h>

int
main()
{
    printf("O hai, world!\n");
}
```

instruções

instruções


```
printf("O hai, world!\n");
```

loops

loops


```
while (true)
{
    printf("O hai!\n");
}
```

loops

loops


```
for (int i = 0; i < 10; i++)
{
    printf("O hai!\n");
}
```

variáveis

variáveis

```
int counter = 0;
while (true)
{
    printf("%d\n", counter);
    counter++;
}
```
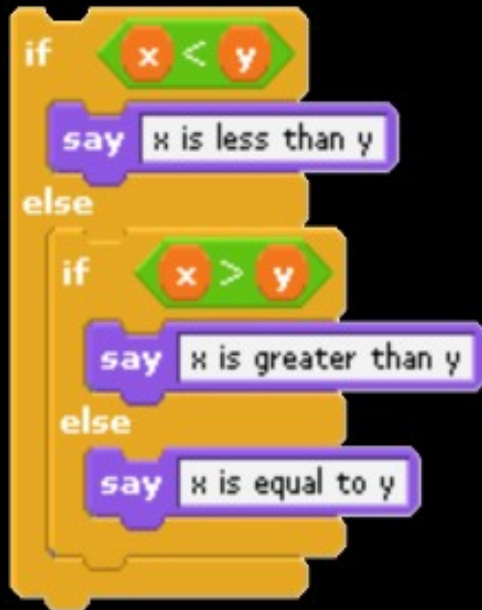
Expressões Booleanas

Expressões Booleanas


```
(x < y)
((x < y) && (y < z))
```
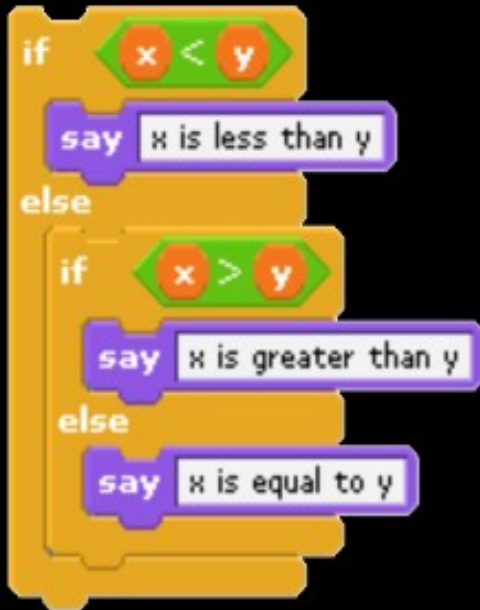
Condições

Condições

```c
if (x < y)
{
    printf("x é menor que y\n");
}
else if (x > y)
{
    printf("x é maior que y\n");
}
else
{
    printf("x é igual a y\n");
}
```

arrays


```
string inventory[1];
Inventory[0] = "Orange";
```

```c
#include <stdio.h>

int
main()
{
    printf("O hai, world!\n");
}
```

```
10000011 00000001 00010001 00000000 00111101 11111100 01110100 00111101
00000000 01000000 00000000 00000000 00000000 00000000 00000000 00000000
10010000 00000000 00000000 00000000 01010000 00000000 00000111 00110000
00001011 00000001 00001011 00000011 00001010 00000000 00000000 00000000
00000000 00100000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00100000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
01110000 00010000 00000000 00100000 00000001 00000000 00000000 00000000
00000000 00000000 00000000 00100000 00000001 00000000 00000000 00000000
00000000 00000000 00000000 01000000 00000001 00000000 00000000 00000000
00000000 00100000 00000000 01000000 00000001 00000000 00000000 00000000
11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
10010000 10000000 00000000 01000000 00000001 00000000 00000000 00000000
00101110 01100100 01111001 01101110 01100001 01101101 01101001 01100011
10110000 00000100 00000000 00100000 00000001 00000000 00000000 00000000
10110000 00000100 00000000 00100000 00000001 00000000 00000000 00000000
10100000 00000001 00000000 00000000 00000000 00000000 00000000 00000000
10110000 00000100 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00100000 00000000 00000000
```

. . .

Como fazer um programa

1. `nano hello.c`
2. gcc hello.c
3. `./a.out`

Como fazer um programa

1. `nano hello.c`
2. gcc -o hello hello.c
3. ./hello

Como fazer um programa

```
1. nano hello.c
2. make hello
3. ./hello
```

funções

```
main      ...
```

Standard Library

```
printf

...
```

Biblioteca do CC50

```
GetChar

GetDouble

GetFloat

GetInt

GetLongLong

GetString
```

printf

`%c   %d   %f   %lld %s   ...`

escape sequences

\n \r \t \' \" \\ \0 ...

matemática

+ - * / %

estruturas primitivas

char double float int long long ...

estruturas CC50

bool string

Operator	Description	Associativity
( ) [ ] . -> ++ --	Parentheses (grouping) Brackets (array subscript) Member selection via object name Member selection via pointer Postfix increment/decrement (see Note 1)	left-to-right
++ -- + - ! ~ (type) * & sizeof	Prefix increment/decrement Unary plus/minus Logical negation/bitwise complement Cast (change *type*) Dereference Address Determine size in bytes	right-to-left
* / %	Multiplication/division/modulus	left-to-right
+ -	Addition/subtraction	left-to-right
<< >>	Bitwise shift left, Bitwise shift right	left-to-right
< <= > >=	Relational less than/less than or equal to Relational greater than/greater than or equal to	left-to-right
== !=	Relational is equal to/is not equal to	left-to-right
&	Bitwise AND	left-to-right
^	Bitwise exclusive OR	left-to-right
|	Bitwise inclusive OR	left-to-right
&&	Logical AND	left-to-right
||	Logical OR	left-to-right
?:	Ternary conditional	right-to-left
= += -= *= /= %= &= ^= |= <<= >>=	Assignment Addition/subtraction assignment Multiplication/division assignment Modulus/bitwise AND assignment Bitwise exclusive/inclusive OR assignment Bitwise shift left/right assignment	right-to-left
,	Comma (separate expressions)	left-to-right

Como fazer um programa

1. `nano hello.c`
2. `gcc -o hello hello.c -lcc50`
3. `./hello`

X

to be continued...

CC50

O CURSO DE HARVARD. NO BRASIL.