Programação em Linguagem C

Compilador Online GDB

Estrutura Básica de Programação

Plataforma de Programação
Online GDB

https://www.onlinegdb.com/

Plataforma de Programação
Online GDB

https://www.onlinegdb.com/

Botões de comandos

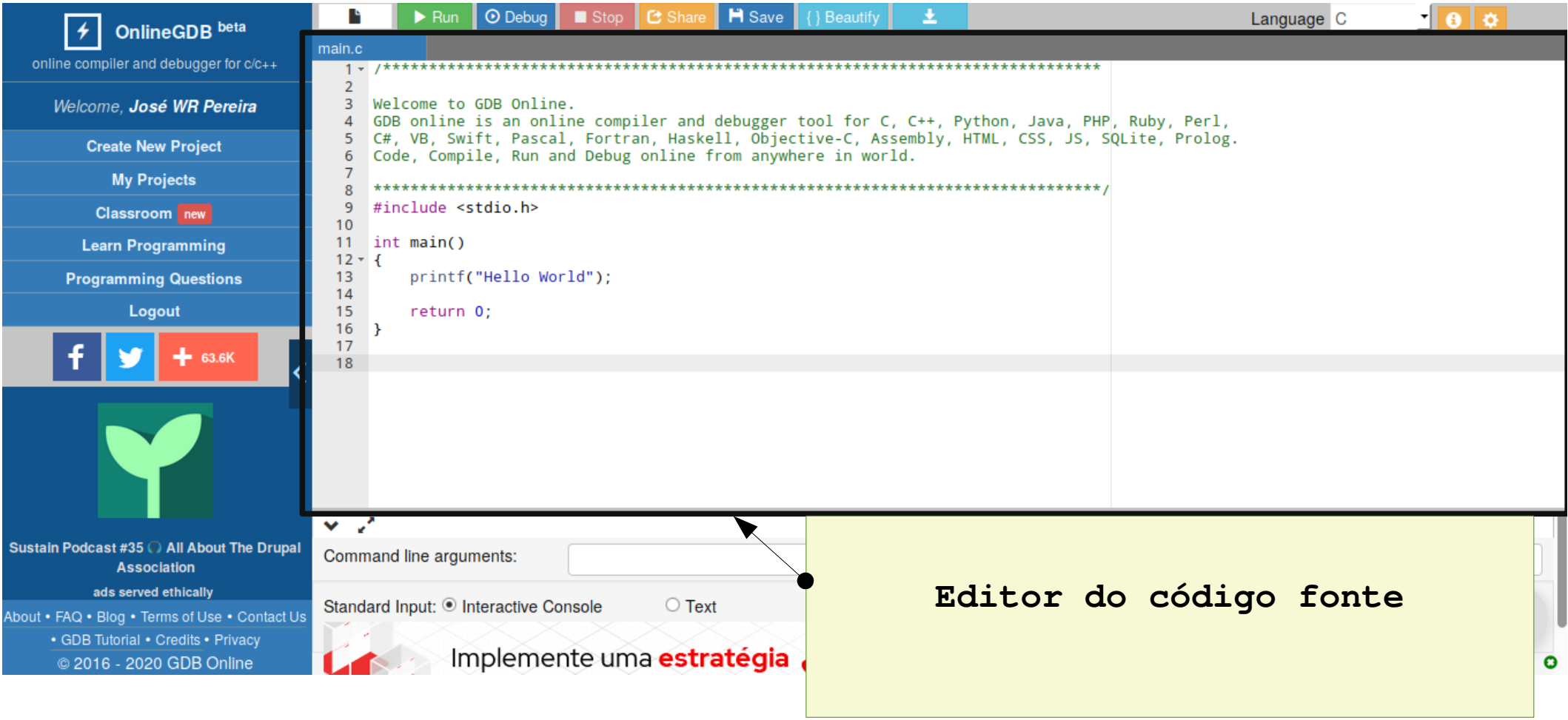
New File: Novo arquivo.
Run: Executar o programa.
Debug: Depurar o programa.
Stop: Parar a execução do prog.
Share: Compartilhar o código.
Save: Salvar o código.
Beautify: 'Embelezar'– Ajustar a indentação do código.
Download: Baixar o código.

Plataforma de Programação
Online GDB

https://www.onlinegdb.com/

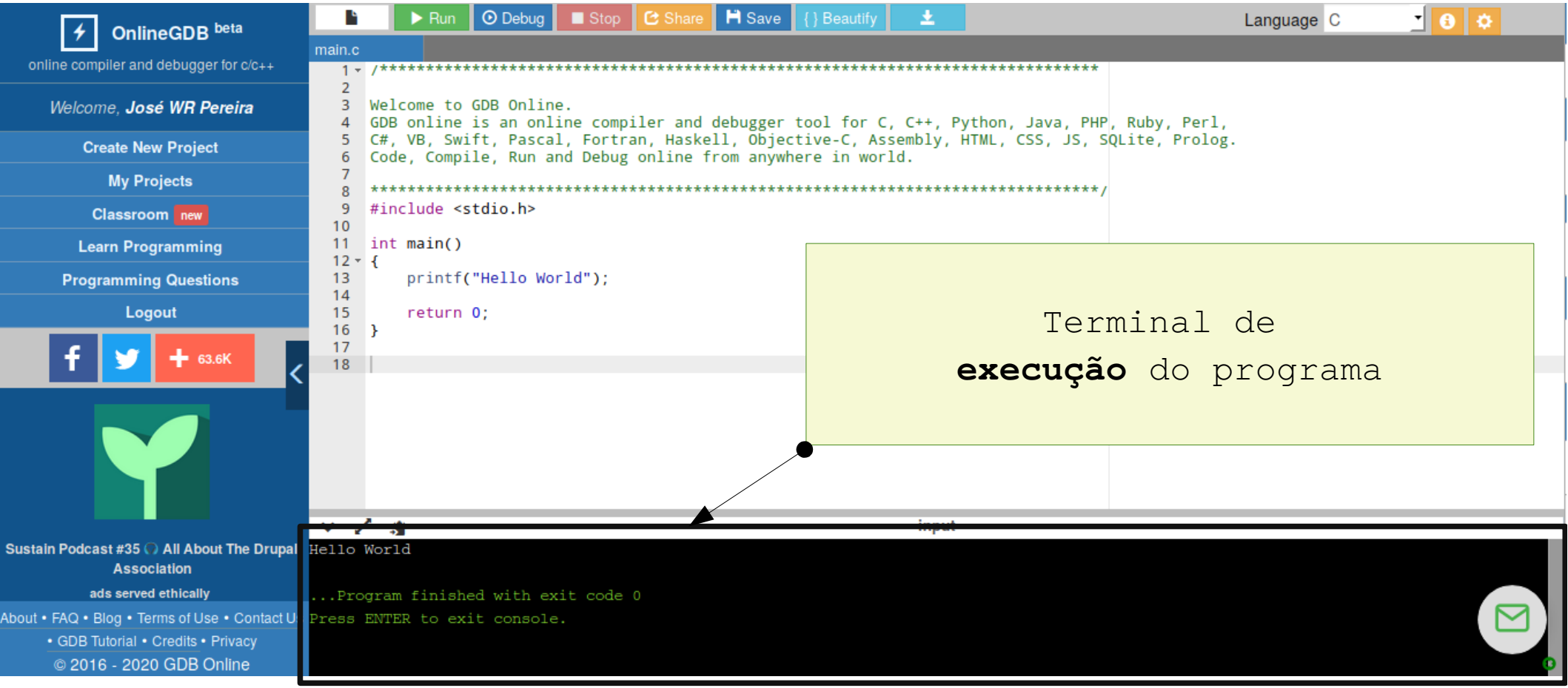
Plataforma de Programação
Online GDB

https://www.onlinegdb.com/



Plataforma de Programação
Online GDB

https://www.onlinegdb.com/



Estrutura básica de um programa em linguagem C

main.c

```c
/******************************************************

Welcome to GDB Online.
GDB online is an online compiler and debugger tool for
C, C++, Python, Java, PHP, Ruby, Perl, C#, VB, Swift,
Pascal, Fortran, Haskell, Objective-C, Assembly, HTML,
CSS, JS, SQLite, Prolog.

******************************************************/

// Code, Compile, Run and Debug online from anywhere in world.


#include <stdio.h>


int main()
{
    printf("Hello World");

    return 0;
}

```

Comentário em bloco

Comentário de linha

Inclusão de bibliotecas

Programa principal

Estilos de código

main.c

```c
#include<stdio.h>
int
main
(
)
{
printf
(
"Hello World"
)
;
return
0;
}
```

main.c

```c
#include<stdio.h>

int main(){
    printf("Hello World");
    return 0;
}
```

main.c

```c
#include<stdio.h>
int main(){ printf("Hello World");return 0;}
```

main.c

```c
#include <stdio.h>

int
main ()
{
  printf ("Hello World");

  return 0;
}
```

main.c

```c
#include <stdio.h>

int main()
{
    printf("Hello World");

    return 0;
}
```

Estilos de código

main.c

```
1  #include<stdio.h>
2  int
3  ain
4
5
6
7
8
9  "Hell World"
10 )
11 ;
12 return
13 0;
14 }
15
```

main.c

```
1  #include<stdio.h>
2
3  int main(){
4      printf("Hello World");
5      return 0;
6  }
7
```

main.c

```
1  #include <stdio.h>
2
3  int
4  main ()
5  {
6    printf ("Hello World");
7
8    return 0;
9  }
10
```

main.c

```
1  #include <stdio.h>
2
3  int main()
4  {
5      printf("Hello World");
6
7      return 0;
8  }
9
```

main.c

```
1  #include
2  int main(
3
4
```

```
ntf("Hello World");return 0;}
```

Biblioteca padrão
Entradas e Saídas

main.c

```c
#include <stdio.h>

int main()
{
    printf("Hello World");

    return 0;
}

```

Inclusão de biblioteca

stdio.h

(Standard Input Output)

Possui funções de manipulação da entrada e da saída padrão.

Exemplo:

printf()

scanf()

Estrutura básica
função main()

main.c

```c
#include <stdio.h>

int main()
{
    printf("Hello World");

    return 0;
}

```

Programa principal (main)

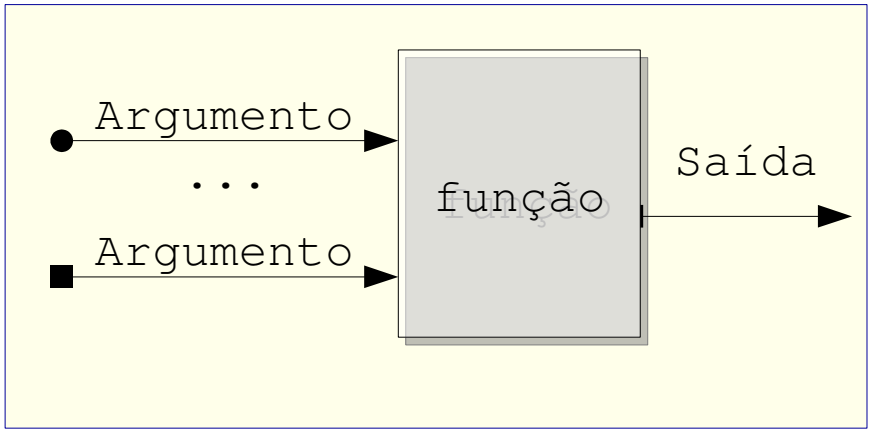
Todo programa em linguagem C é estruturado em uma ou mais **Funções.**

O nome das funções é de escolha do programador, exceto a única função obrigatória, **main.**

Estrutura básica de uma função

main.c

```c
#include <stdio.h>

int main()
{
    printf("Hello World");

    return 0;
}
```



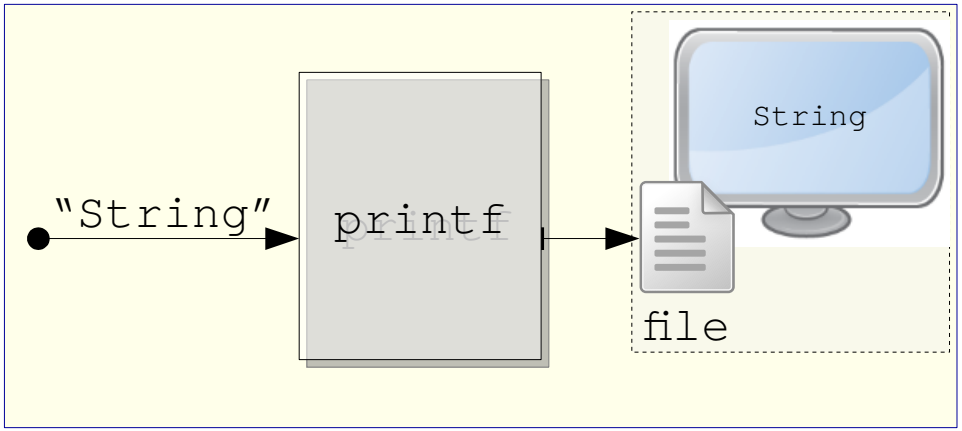
Estrutura de uma função

```
<tipo_saída> nome_da_funcao(<tipo_entrada> argumento,…)
{  início do corpo da função
   corpo da função com retorno de dado
}  fim do corpo da função
```

Função printf()

source code

```c
#include <stdio.h>

int main()
{
    printf("String");
    return 0;
}

```



printf

Imprime uma cadeia de caracteres no arquivo de saída padrão.
A saída padrão, geralmente, é o monitor.

Memória

...	'S'	't'	'r'	'i'	'n'	'g'	\0	...

Função printf()

source code

```
1  #include <stdio.h>
2
3  int main()
4  {
5      printf("%s", "String");
6      return 0;
7  }
8
```

Saída

```
String
```

printf

Imprime uma cadeia de caracteres no arquivo de saída padrão.
A saída padrão, geralmente, é o monitor.

Memória

...	'S'	't'	'r'	'i'	'n'	'g'	\0	...

Função printf()

source code

```
1  #include <stdio.h>
2
3  int main()
4  {
5      printf("%s%s", "Ola ","Mundo");
6      return 0;
7  }
8
```

Saída

```
Ola Mundo
```

Formato de impressão

%s: string – cadeia de caracteres

8 bits
1 byte

...	'O'	'l'	'a'	' '	\0	...	'M'	'u'	'n'	'd'	'o'	\0	...

Função printf()

main.c

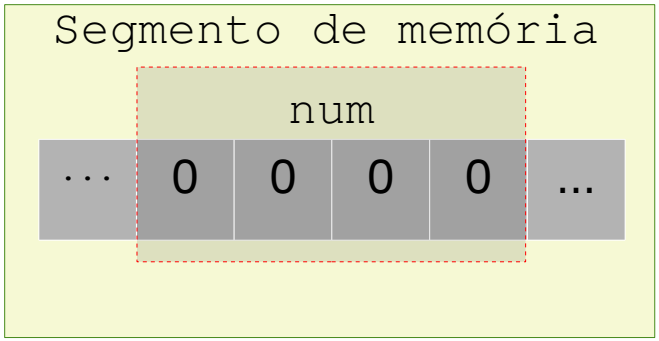
```c
#include <stdio.h>

int main()
{
    int num = 0;
    printf("Numero: %d\n", num );
    return 0;
}
```

Saída

```
Numero: 0
```

Formato de impressão

%d: número inteiro – 4 bytes



Função printf()

main.c

```c
#include <stdio.h>

int main()
{
    int num = 0;
    printf("Numero: %d\n", num );
    printf("Numero: %3d\n", num );
    return 0;
}
```

Saída

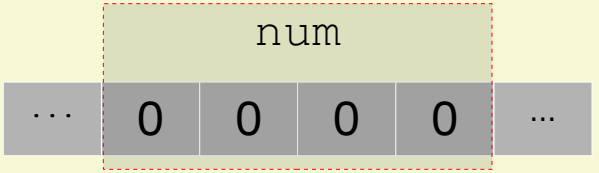
```
Numero: 0
Numero:   0
```

Formato de impressão

%d: número inteiro – 4 bytes

%nd: número de espaços reservados

Segmento da memória



Função printf()

main.c

```c
#include <stdio.h>

int main()
{
    int num = 74;
    printf("Numero: %d\n", num );
    printf("Numero: %3d\n", num );
    return 0;
}

```

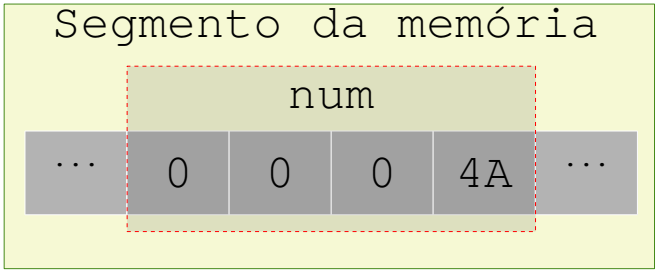
Saída

```
Numero: 74
Numero:  74
```

Formato de impressão

%d: número inteiro – 4 bytes

%nd: número de espaços reservados



74 = 0b01001010 = 0x4A

Função printf()

main.c

```c
#include <stdio.h>

int main()
{
    int num = 74;
    printf("Numero: %d\n", num );
    printf("Caractere: %c\n", num );
    return 0;
}
```

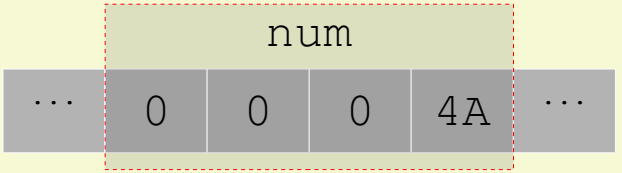
Saída

```
Numero: 74
Caractere: J
```

Formato de impressão

%c: caractere – símbolo gráfico convencional para:
- Digitos Numéricos('0','1','2',...)
- Letras('A','B','C',...'a','b','c',...)
- Código de controle(\n,\t,...)
- Símbolo especial

Segmento da memória



74 = 0b01001010 = 0x4A

Função scanf()

main.c

```c
#include <stdio.h>

int main()
{
    int num = 0;
    printf("Numero: %d\n", num );
    scanf("%d", &num );
    printf("Numero: %d\n", num );
    return 0;
}
```

Saída

```
Numero: 0
```

Entrada

```
74<Enter>
```

Saída

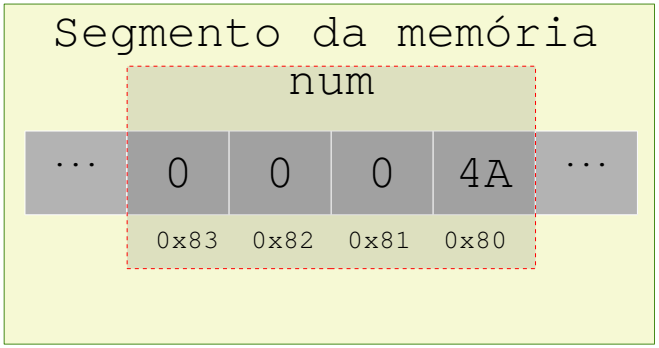
```
Numero: 74
```

scanf()

Lê a entrada padrão, que geralmente, é o teclado

%d: tipo de dado de entrada: inteiro

&<var>: endereço da variável que receberá o dado lido.



74 = 0b01001010 = 0x4A

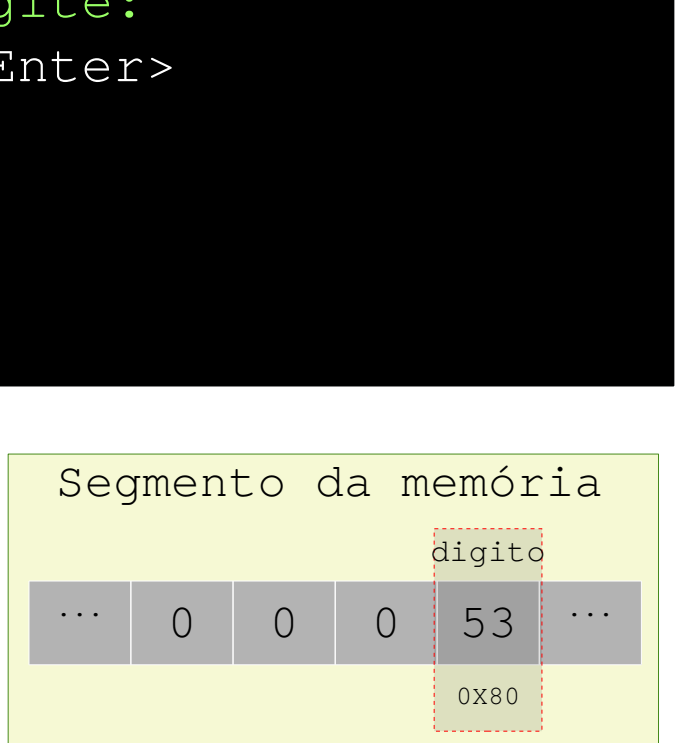
Função scanf()

main.c

```c
#include <stdio.h>

int main()
{
    char digito;
    printf("Digite: \n");
    scanf("%c", &digito );
    printf("%c\n", digito );
    return 0;
}

```

Saída

```
Digite:
S<Enter>
S
```



scanf()

%c: tipo de dado de entrada: caractere

83 = 0b01010011 = 0x53

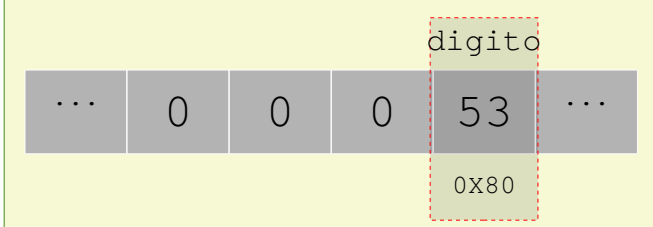
Função scanf()

main.c

```c
#include <stdio.h>

int main()
{
    char digito;
    printf("Digite: \n");
    scanf("%c", &digito );
    printf("%c\n", digito );
    return 0;
}

```

Saída

```
Digite:
SENAI<Enter>
S
```



scanf()

%c: tipo de dado de entrada: caractere

83 = 0b01010011 = 0x53

Função scanf()

main.c

```c
#include <stdio.h>

int main()
{
    char palavra[6];
    printf("Digite: \n");
    scanf("%s",  palavra );
    printf("%s\n", palavra );
    return 0;
}

```

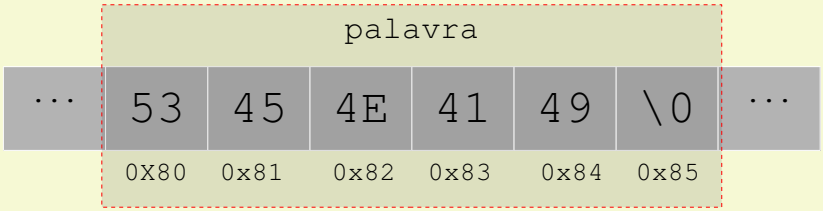
Saída

```
Digite:
SENAI<Enter>
SENAI
```

scanf()

%s:String – cadeia de caracteres

Segmento da memória



Função scanf()

main.c

```c
#include <stdio.h>

int main()
{
    char palavra[6];
    printf("Digite: \n");
    scanf("%s",  palavra );
    printf("%s\n", palavra );
    return 0;
}

```

Saída

```
Digite:
SENAI<Enter>
SENAI
```

scanf()

%s:String – cadeia de caracteres

Segmento da memória

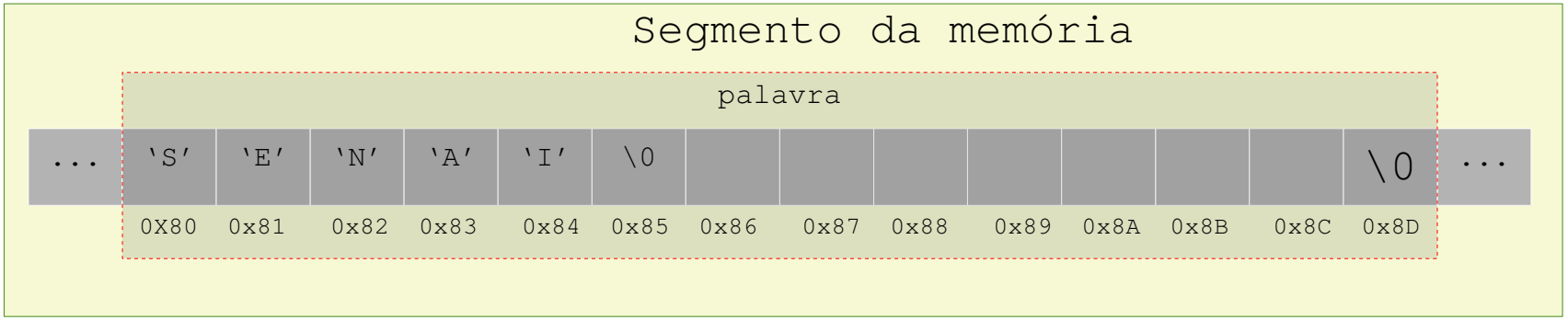
Função scanf()

main.c

```c
#include <stdio.h>

int main()
{
    char escola[14];
    printf("Escola: \n");
    scanf("%s", escola );
    printf("%s\n", escola );
    return 0;
}

```

Saída

```
Escola:
SENAI Jandira<Enter>
SENAI
```



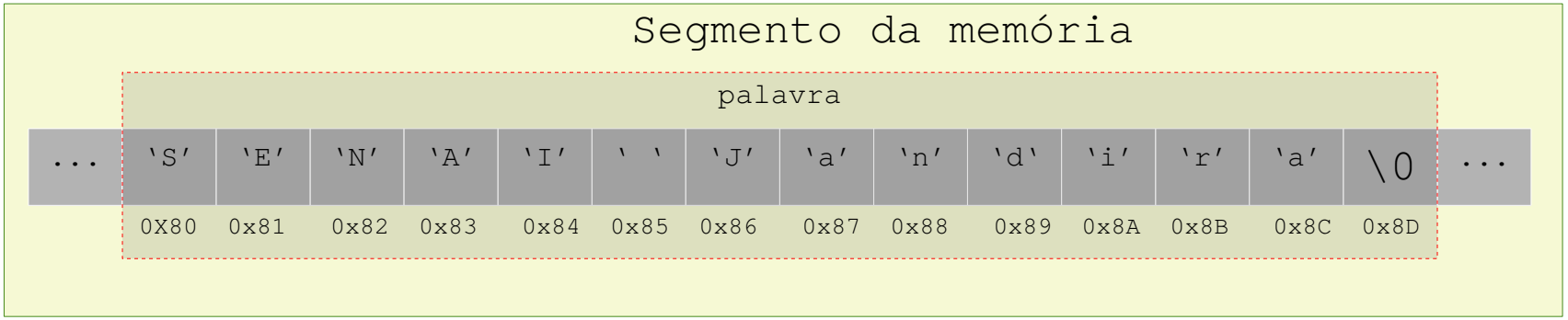
Função scanf()

main.c

```c
#include <stdio.h>

int main()
{
    char escola[14];
    printf("Escola: \n");
    scanf("%13[^\n]", escola );
    printf("%s\n", escola );
    return 0;
}

```

Saída

```
Escola:
SENAI Jandira<Enter>
SENAI Jandira
```



Site: **josewrpereira**.github.io/ddp/

E-mail: **josewrpereira**@outlook.com