

**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
BAHIA

LINGUAGEM DE PROGRAMAÇÃO

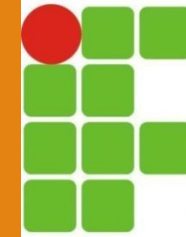
INTRODUÇÃO À LP



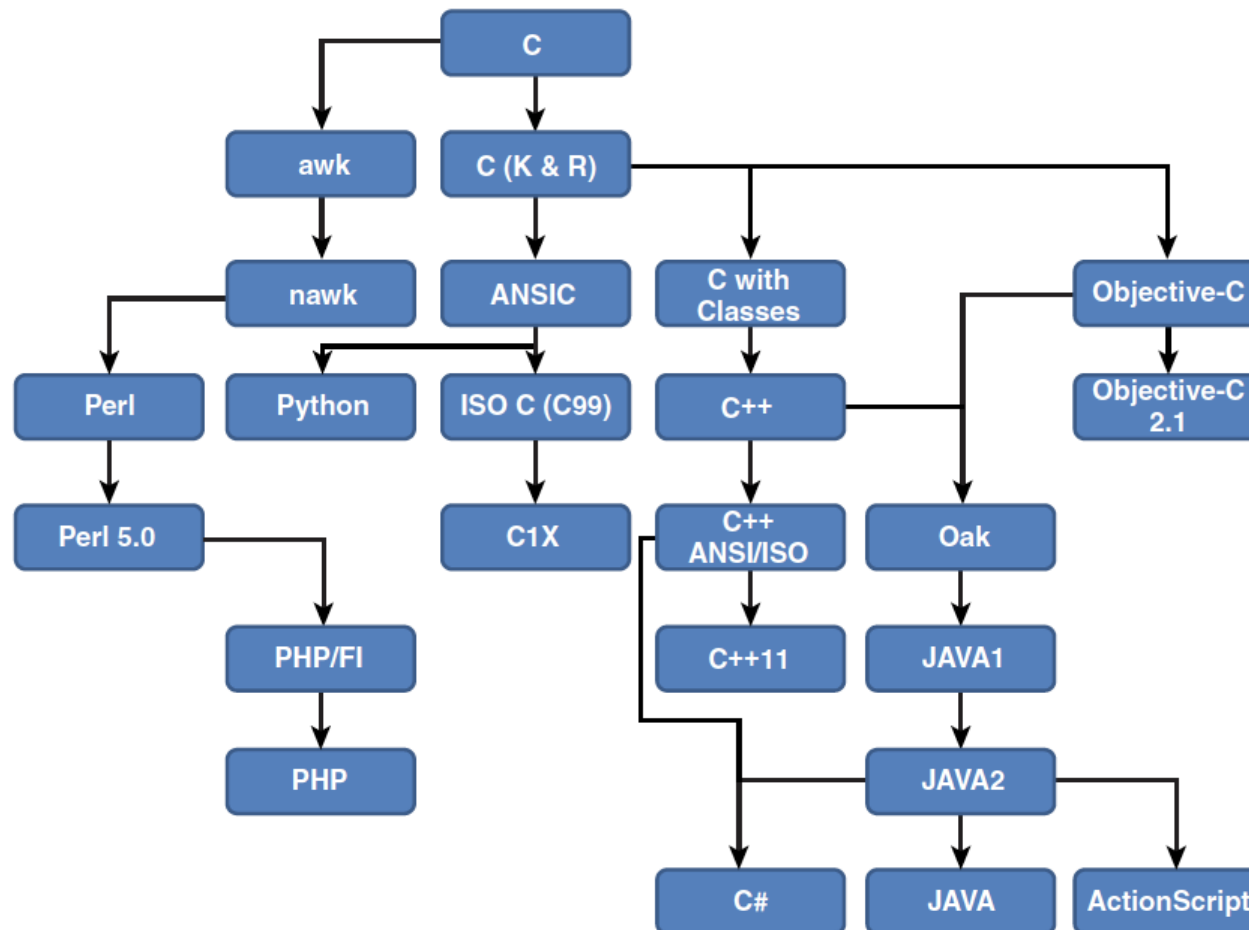
INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA

- Linguagem C – criada em 1972
 - LP de alto nível
 - LP procedural (modularização)
 - Estrutura simples
 - Permite acesso de baixo nível à memória
 - Permite a implantação de programas utilizando linguagem Assembly
 - Linguagem criada para incentivar a programação multiplataforma
- IDEs
 - Dev C++
 - Visual Studio Code
 - Code::Blocks

INFLUÊNCIA DE C EM OUTRAS LINGUAGENS



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA



INFLUÊNCIA DE C EM OUTRAS LINGUAGENS



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA

- Imprimindo de 1 a 10

```
C  
for(i = 1; i <= 10;i++)  
{  
    printf("%d\n",i);  
}
```

```
PHP  
for ($i = 1; $i <= 10; $i++)  
{  
    echo $i;  
}
```

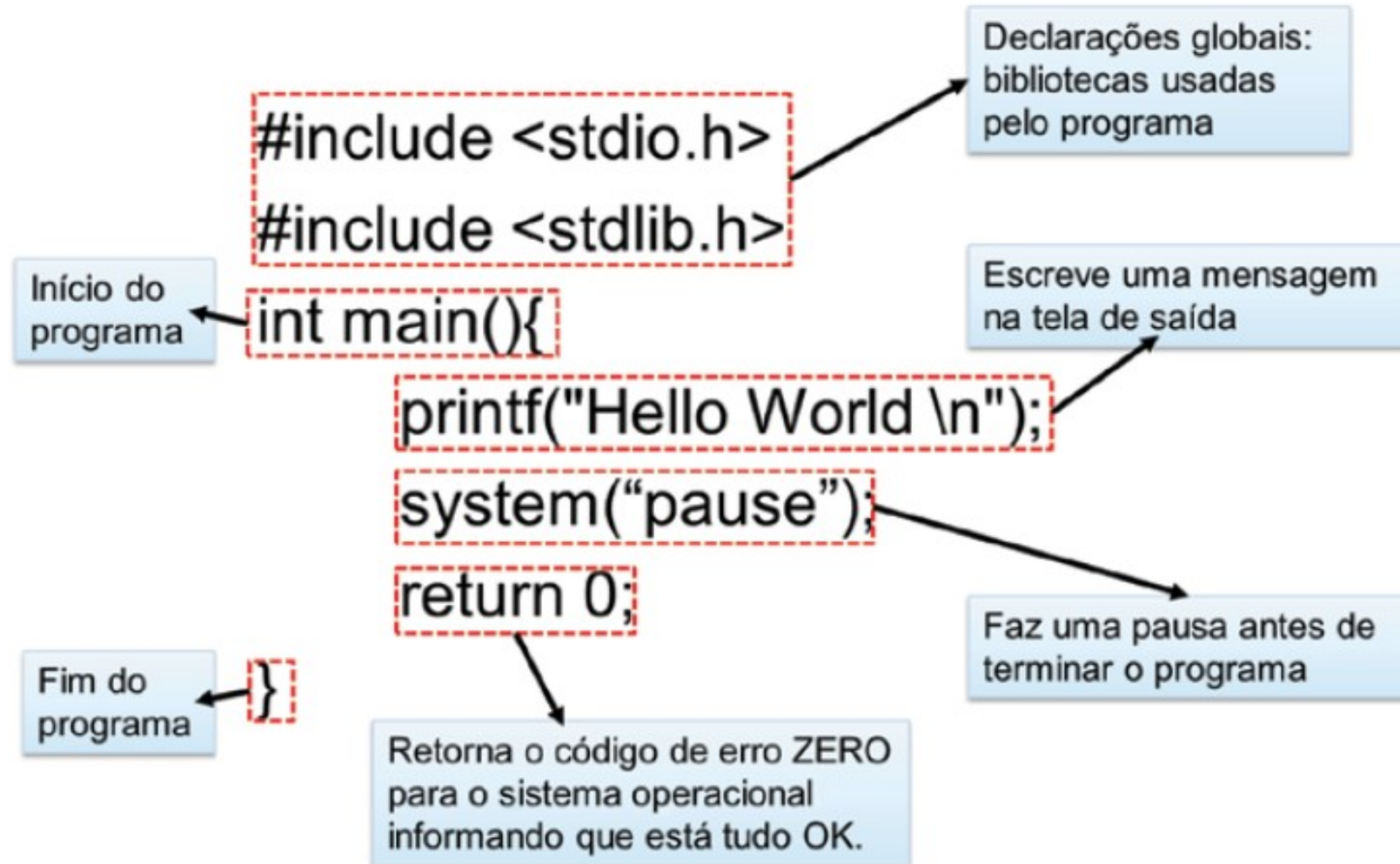
```
Java  
for(int i=1;i<=10;i++)  
{  
    System.out.println(i);  
}
```

```
Perl  
for($i = 1; $i<=10; $i++)  
{  
    print $i;  
}
```

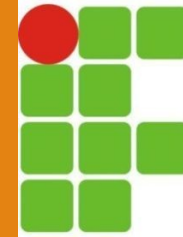
ESQUELETO DE UM PROGRAMA EM C



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA



INDENTAÇÃO DO CÓDIGO



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA

- É o espaçamento (ou tabulação) colocado antes de começar a escrever o código
- É uma convenção de escrita de código
- Auxilia na leitura, interpretação e manutenção do código
- Em C, a indentação não é obrigatória, mas é importante para o entendimento do código.
- OBS: Sempre usem indentação

Exemplo

```
01  #include <stdio.h>
02  #include <stdlib.h>
03  int main(){printf("Hello World \n"); system("pause"); return 0;}
```

COMENTÁRIOS

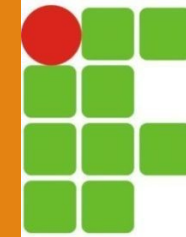


INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA

Exemplo: comentários no programa

```
01  #include <stdio.h>
02  #include <stdlib.h>
03  int main(){
04      /*
05      A funcao printf()
06      serve para
07      escrever na tela
08      */
09      printf("Hello World \n");
10      //faz uma pausa no programa
11      system("pause");
12      return 0;
13  }
```

VARIÁVEIS



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA

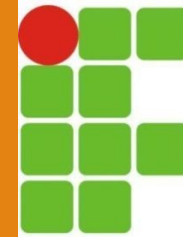
- Uma variável é uma entidade capaz de representar um valor ou uma expressão
- Na computação, variável é uma posição de memória, onde um determinado dado ou valor é armazenado para ser lido ou alterado
- Antes de ser usada, a variável deve ser declarada
- Sintaxe da declaração de uma variável:

```
tipo_da_variável nome_da_variável;
```

- Ex: `int x = 10;`

Memória		
endereço	variável	conteúdo
119		
120		
121	int x	10
122		
123		
124		

NOMES DE VARIÁVEIS



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA

- É um conjunto de caracteres que podem ser letras, números ou underscore(_)
- Deve sempre iniciar com uma letra ou underscore

Exemplo: nomeando variáveis

comp!	.var	int	1var	1cont	-x	Va-123	
cont	Cont	Val_123	_teste	Int1	Cont1	X	

NOMES DE VARIÁVEIS

- É um conjunto de caracteres que podem ser letras, números ou underscore(_)
- Deve sempre iniciar com uma letra ou underscore
- A linguagem C é case sensitive (int Soma, int SOMA e int soma)
- Palavras-chaves não podem ser utilizadas

Lista de palavras-chave da linguagem C

auto	double	int	struct	break	else	long	switch
case	enum	if	typeof	continue	float	return	while
union	const	for	short	unsigned	char	extern	signed
void	default	do	sizeof	volatile	goto	register	static

TIPOS DE VARIÁVEIS

- A linguagem C possui 5 tipos de dados básicos

Tipo	Bits	Intervalo de valores
char	8	-128 A 127
int	32	-2.147.483.648 A 2.147.483.647
float	32	1,175494E-038 A 3,402823E+038
double	64	2,225074E-308 A 1,797693E+308
void	8	sem valor

TIPOS DE VARIÁVEIS

- A linguagem C possui 5 tipos de dados básicos

Tipo	Bits	Intervalo de valores
char	8	-128 A 127
int	32	
float	32	
double	64	
void	8	

permite armazenar em um único byte (8 bits) um número inteiro muito pequeno ou o código de um caractere do conjunto de caracteres da tabela **ASCII**:

```
char c = 'a';  
char n = 10;
```

TIPOS DE VARIÁVEIS

- A linguagem C possui 5 tipos de dados básicos

Tipo	Bits	Intervalo de valores
char	8	-128 A 127
int	32	-2.147.483.648 A 2.147.483.647
float	32	
double	64	
void	8	

Esse tipo de dados permite armazenar um número inteiro (**sem parte fracionária**). Seu tamanho depende do processador em que o programa está rodando e é tipicamente 16 ou 32 bits:

```
int n = 1459;
```

TIPOS DE VARIÁVEIS

- A linguagem C possui 5 tipos de dados básicos

Tipo	Bits	Intervalo de valores
char	8	-128 A 127
int	32	-2.147.483.648 A 2.147.483.647
float	32	1,175494E-038 A 3,402823E+038
double	64	2,225074E-308 A 1,797693E+308

Esses tipos de dados permitem armazenar um valor real (**com parte fracionária**), também conhecido como **ponto flutuante**. A diferença entre eles é a sua precisão:

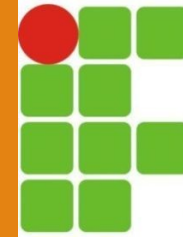
Tipo ••**float**: precisão simples.

Tipo ••**double**: dupla precisão.

São úteis quando queremos trabalhar com intervalos de números reais realmente grandes.

A linguagem C usa o padrão numérico americano, ou seja, a parte decimal

IMPRIMINDO A VARIÁVEL



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA

- `printf()` → função de saída / escrita de dados

`printf("tipos de saída", lista de variáveis)`

- “tipos de saída”: conjunto de caracteres que especifica o formato dos dados a serem escritos e/ou o texto a ser escrito.
- lista de variáveis: conjunto de nomes de variáveis, separados por vírgula, que serão escritos.

Exemplo: escrevendo um texto na tela

```
01  #include <stdio.h>
02  #include <stdlib.h>
03  int main(){
04
05      printf("Esse texto sera escrito na tela");
06      system("pause");
07      return 0;
    }
```

Saída	Esse texto sera escrito na tela
-------	---------------------------------

IMPRIMINDO A VARIÁVEL

- Quando queremos escrever dados formatados na tela usamos a forma geral da função, a qual possui os tipos de saída.
- Eles especificam o formato de saída dos dados que serão escritos pela função `printf()`.
- Cada tipo de saída é precedido por um sinal de %, e um tipo de saída deve ser especificado para cada variável a ser escrita.



← `printf("%tipo_de_saída", expressão);`



← `printf("%tipo1 %tipo2", expressão1, expressão2);`

IMPRIMINDO A VARIÁVEL

- Quando queremos escrever dados formatados na tela usamos a forma geral da função, a qual possui os tipos de saída.
- Eles especificam o formato de saída dos dados que serão escritos pela função printf().
- Cada tipo de saída é precedido por um sinal de %, e um tipo de saída deve ser especificado para cada variável a ser escrita.



← printf(" %tipo_de_saída", expressão);

Alguns tipos de saída

%c	escrita de um caractere (char)
%d ou %i	escrita de números inteiros (int ou char)
%u	escrita de números inteiros sem sinal (unsigned)
%f	escrita de número reais (float ou double)
%s	escrita de vários caracteres
%p	escrita de um endereço de memória
%e ou %E	escrita em notação científica

IMPRIMINDO A VARIÁVEL



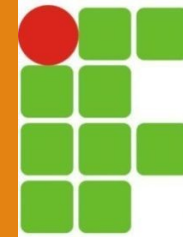
INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA

Exemplo: escrita de dados na linguagem C

```
01  #include <stdio.h>
02  #include <stdlib.h>
03  int main(){
04  int x = 10;
05      //Escrita de um valor inteiro
06      printf("%d\n",x);
07      float y = 5.0;
08      //Escrita de um valor inteiro e outro real
09      printf("%d%f\n",x,y);
10      //Adicionando espaço entre os valores
11      printf("%d %f\n",x,y);
12      system("pause");
13      return 0;
14  }
```

Saída
10
105.000000
10 5.000000

IMPRIMINDO A VARIÁVEL



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA



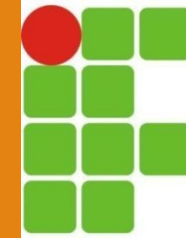
Junto ao tipo de saída, pode-se adicionar texto e não apenas espaços.

```
01  #include <stdio.h>
02  #include <stdlib.h>
03  int main(){
04      int x = 10;
05      printf("Total = %d\n",x);
06      printf("%d caixas\n",x);
07      printf("Total de %d caixas\n",x);
08      system("pause");
09      return 0;
10  }
```

Saída

```
Total = 10
10 caixas
Total de 10 caixas
```

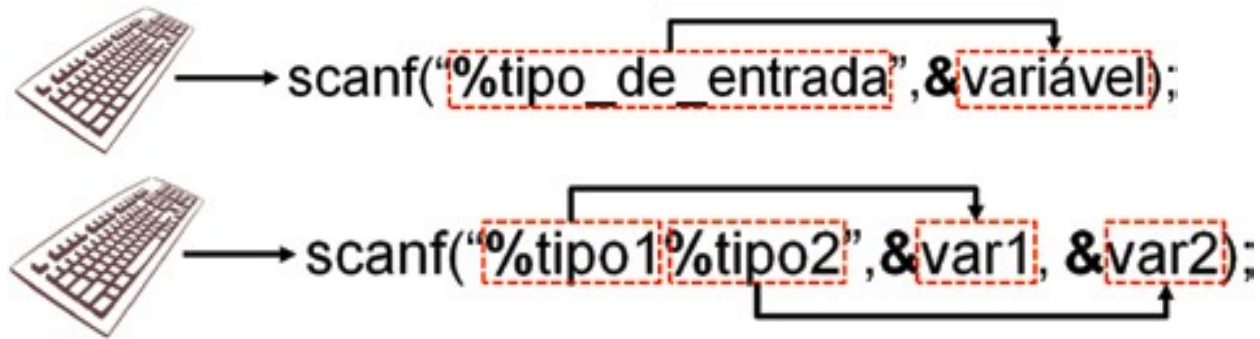
LENDO A VARIÁVEL DO TECLADO



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA

- `scanf()` → função de entrada/ leitura de dados em C
- A forma geral da função `scanf()` é:

`scanf("tipos de entrada", lista de variáveis)`
- Os tipo de entrada especificam o formato de entrada dos dados que serão lidos pela função `scanf()`.
- Cada tipo de entrada é precedido por um sinal de %, e um tipo de entrada deve ser especificado para cada variável a ser lida.



Alguns tipos de saída

<code>%c</code>	leitura de um caractere (char)
<code>%d</code> ou <code>%i</code>	leitura de números inteiros (int ou char)
<code>%f</code>	leitura de número reais (float ou double)
<code>%s</code>	leitura de vários caracteres