



Introdução à linguagem C

Programa Principal

```
#include <stdio.h>
int main()
{
    int a,b;

    printf("Digite um valor: ");
    scanf("%i", &a);

    b = (2*a)+1;

    printf("Novo valor = %i", b);
    return 0;
}
```

Programa Principal

`#include <stdio.h>` Diretiva de compilação

`int main()` Programa principal
`{` função **main**

`int a, b;` Declaração de variáveis

```
printf("Digite um valor: ");  
scanf("%i", &a);
```

```
b = (2*a) + 1;
```

```
printf("Novo valor = %i", b);  
return 0;
```

}

Instruções da função **main**: corpo do programa

Tipos de dados

Existem 5 tipos de dados nativos:

int	float	char	double	void
-----	-------	------	--------	------

Usados para definir funções e na **declaração de variáveis**. Exemplos:

```
int a, b;
```

```
float x, y = 3.5;
```

```
char A = '1', B = 's';
```

Atribuição:

```
x = (2*a) + 1;  
y = 2*x / 4 + 5 / 2;
```

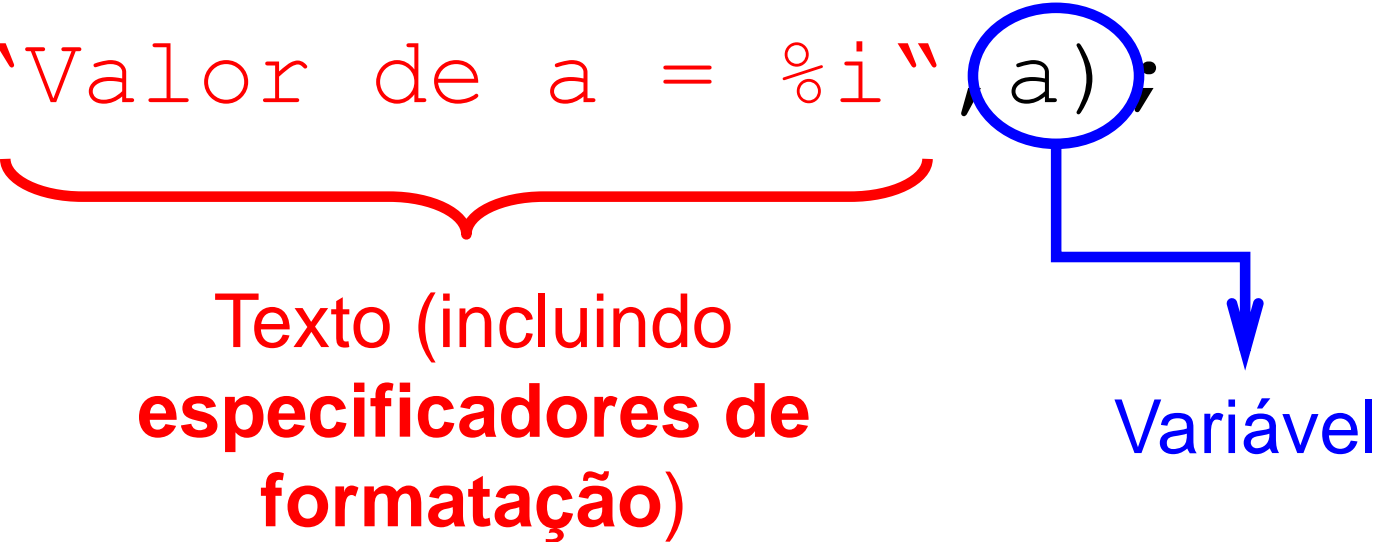
Toda expressão resulta em valor numérico:

```
z = 2*5 / 4 + 5 / 2;  
y = (3 > 4);    Resulta igual a 0, por ser falsa
```

Comparação: **y = (x==3);** Mais tarde...

Exibição:


```
printf ("Valor de a = %i", a);
```



The diagram illustrates the components of the `printf` function. A red curly brace under the string `"Valor de a = %i"` is labeled "Texto (incluindo especificadores de formatação)". The variable `a` is circled in blue, and a blue arrow points from it to the label "Variável".

Leitura:

```
scanf ("%f", &x);
```



The diagram illustrates the components of the `scanf` function. A red curly brace under the string `"%f"` is labeled "Especificador de formatação". The address `&x` is circled in blue, and a blue arrow points from it to the label "Endereço da variável".

Exibição de mais de um valor (variável):

```
printf("a = %i e b = %i", a, b);
```

Pulando linhas na exibição:

```
printf("Estudo de caso \n");
```

Observações:

- Para **int**, **float** e **char**: **%i**, **%f** e **%c**.
- Tanto ***printf*** quanto ***scanf*** são funções da “biblioteca” **stdio.h**.