




---

---

---

---

---

---

---

---

### Imprimindo informações

- `printf(string de formatação, variáveis)`
  - Biblioteca **stdio.h**
  - Caracteres especiais não são suportados
  - Caracteres de formatação

Símbolo	Descrição
%c	Caractere
%d, %i	Número inteiro (com sinal)
%u	Número inteiro (sem sinal)
%e, %E	Notação científica
%f	Ponto flutuante
%lf	Ponto flutuante longo
%o	Octal
%s	String
%x, %X	Hexadecimal
%p	Ponteiro
%%	Símbolo de percentual

```

int x = 50;
printf("%d", x);           // 50
printf("%f", x);           // 0.000000
printf("%o", x);           // 62
printf("%x", x);           // 32

printf("A variavel x vale: %d", x);
// A variavel x vale: 50

int y = 30;
printf("%d + %d = %d", x, y, x+y);
// 50 + 30 = 80
  
```

---

---

---

---

---

---

---

---

### sprintf

- Idêntico ao `printf`
- Não exibe na tela, mas sim, aloca em uma variável de texto
- Utilizada para converter números em texto

```

char meuBuffer[50];
int x = 14;

sprintf(meuBuffer, "%d", x);           // meuBuffer conterá "14"
sprintf(meuBuffer, "%d * 2 = %d", x, x*2); // meuBuffer conterá "14 * 2 = 28"
  
```

---

---

---

---

---

---

---

---

## Formatando valores numéricos



- `%[a].[b]símbolo`
  - **a**: quantidade de espaços (casas) deve ocupar
  - **b**:
    - Inteiros: quantidade mínima de números que deve aparecer, preenchendo com zeros a esquerda
    - Ponto flutuante: quantidade de casas decimais
    - Strings: quantidade máxima de caracteres

```
printf("%3.5f\n", 123.456789); // 123.45679
printf("%12.6f\n", 123.456789); // _123.456789

printf("%3.5d\n", 12); // 00012
printf("%2.3d\n", 12); // 012
printf("%10.1d\n", 12); // _12
printf("%10.3d\n", 12); // _012

printf("%10.15s\n", "Teste"); // _Teste
printf("%10.15s\n", "Teste de string digitada"); // Teste de string
```

---

---

---

---

---

---

---

---

## Lendo informações



- `scanf(string, endereços de memória)`
  - Biblioteca **stdio.h**
  - Caracteres especiais não são suportados
  - Leitura até o espaço em branco ou tecla enter

Símbolo	Descrição
<code>%c</code>	Caractere
<code>%d, %i</code>	Número inteiro (com sinal)
<code>%u</code>	Número inteiro (sem sinal)
<code>%e, %E</code>	Notação científica
<code>%f</code>	Ponto flutuante
<code>%F</code>	Ponto flutuante longo
<code>%o</code>	Octal
<code>%s</code>	String
<code>%x</code>	Hexadecimal
<code>%p</code>	Ponteiro
<code>%[]</code>	Conjunto de caracteres

```
int x;
printf("Digite um numero inteiro: ");
scanf("%d", &x);
printf("O valor digitado foi: %d", x);
```

---

---

---

---

---

---

---

---

## Gerando números randômicos



- Funções
  - `rand()`
    - Biblioteca **stdlib.h**
    - Retorna um flutuante randômico entre 0 e 1
  - `srand(semente)`
    - Biblioteca **stdlib.h**
    - Define uma nova semente para a função `rand`
  - `time()`
    - Biblioteca **time.h**
    - Retorna quantidade de segundos desde 00:00 01/01/70

---

---

---

---

---

---

---

---

Manipulando datas



- Biblioteca **time.h**
- Tipo de dado **time\_t**
  - Instante (data e hora)
- Estrutura **struct tm**
  - time\_t organizado
- Função **time**
  - Captura o time\_t atual
- Função **localtime**
  - time com fuso horário
- Função **strftime**
  - Formata data e hora

Símbolo	Descrição
%a	Dia da semana abreviado
%A	Dia da semana
%b	Mês abreviado
%B	Mês
%c	Data e hora completa
%d	Dia do mês (1-31)
%H	Hora (24)
%I	Hora(12)
%j	Dia do ano (1-366)
%m	Mês (1-12)
%M	Minuto (0-59)
%p	AM, PM
%S	Segundos (0-59)
%U	Número da semana (a partir de domingo)
%w	Dia da semana (0-6) (a partir de domingo)
%W	Número da semana (a partir de segunda-feira)
%x	Data completa
%X	Horário completo
%y	Ano (yy)
%Y	Ano (yyyy)
%Z	Fuso horário

---

---

---

---

---

---

---

---



---

---

---

---

---

---

---

---