

Aula 05 – Entrada e saída

TECNOLOGIA em **Gestão de Dados**

Programação I
Prof. Dr. Wener Sampaio



ReUni
DIGITAL

CEAD
Centro de Educação
Aberta e a Distância

UFPI
UNIVERSIDADE
FEDERAL DO PIAUÍ

MEC

Sumário

- Saída
 - O comando print()
 - Com vírgulas
 - Com marcadores
 - Tipos de marcadores
- Entrada
 - O comando input()
 - Transformações de tipos

Saída

- `print()`: imprime na saída padrão
 - `print(x)`

```
print('tome suco')  
print('de laranja')  
print(  
    print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
```

- Múltiplos valores
 - `value`: o que você quer imprimir.
 - `sep`: caractere separador. Padrão = espaço.
 - `print('tome suco','de laranja', sep='...')`
 - `print('tome suco','de laranja', sep='')`

Saída

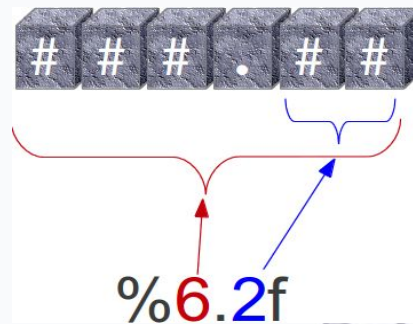
- Múltiplos valores
 - end: caractere terminador. Padrão = \n.
`print('tome suco')`
`print('de laranja', end='\n\n')`
`print('com limão', end='\n\n')`
 - file: Direciona a saída. Padrão = shell
`arq = open('c:/teste/meu_arq.txt','w')`
`print('Tome logo esse suco!', file=arq)`
`arq.close()`

Saída

- Múltiplos valores
 - flush = liberar forçadamente o buffer.
 - Flush = False - Imprimir quando puder. Usa buffer.
 - Flush = True - Imprimir agora. Sem buffer.

Saída

- Múltiplos valores
 - 1º modo: separar por vírgula
 - x, y, z = 10, 20, 30
 - print('altura =', x, 'largura=', y, 'profundidade=', z)
 - 2º modo: Utilizar marcadores
 - print('Hoje é %d de %s de %d.' % (31, 'fevereiro', 2999))
 - print('Hoje é %5d de %s de %d.' % (31, 'fevereiro', 2999))
 - De forma geral
 - %[flags][largura][.precisão]tipo



Saída

- Múltiplos valores
 - 2º modo:

```
print('%6.2f' % (23.789))  
print('%6.2f' % (0.039))  
print('%6.2f' % (199.8))  
print('%6.2f' % (23))  
print('%6.2f' % (2324.17))
```

Saída:

```
=====
23.79
 0.04
199.80
23.00
2324.17
>>> |
```

Saída

- Múltiplos valores
 - 2º modo:

Conversão	Descrição
d, i	Decimal com sinal.
u	Decimal sem sinal.
o	Octal sem sinal
x	hexadecimal sem sinal (minúsculo).
X	hexadecimal sem sinal (maiúsculo).
f, F	Ponto flutuante.
e	Ponto flutuante em notação científica (minúsculo).
E	Ponto flutuante em notação científica (manúsculo).
g	Mesmo que "e" se $-4 < \text{expoente} < \text{precisão}$, cc "f".
G	Mesmo que "e" se $-4 < \text{expoente} < \text{precisão}$, cc "F".
c	caractere simples (inteiro ou caractere).
r	String (criadas por repr()).
s	String (criadas por str()).
%	Caractere %.

Saída

- Múltiplos valores
 - 2º modo:

```
print("a)%10.3e"% (356.08977))
print("b)%10.3E"% (356.08977))
print("c)%10o"% (25))
print("d)%10.3o"% (25))
print("e)%10.5o"% (25))
print("f)%5x"% (47))
print("g)%5.4x"% (47))
print("h)%5.4X"% (47))
print("i)Apenas o sinal %% " % ())
```

Saída:

```
a) 3.561e+02
b) 3.561E+02
c)          31
d)          031
e)         00031
f)         2f
g) 002f
h) 002F
i)Apenas o sinal %
```

Saída

- Múltiplos valores
 - 2º modo:

Flag	Descrição
#	Usado com o, x ou X gera os precedentes 0, 0o, 0O, 0x ou 0X em octais e hexadecimais
0	Preencher com zeros à esquerda.
-	Alinhamento à esquerda.
	se não tem -, então é preenchido com espaços.
+	O sinal ("+" do "-") precedirá o valor

Saída

- Múltiplos valores
 - 2º modo:

```
print("a) %#5X" % (47))
print("b) %5X" % (47))
print("c) %#5.4X" % (47))
print("d) %#5o" % (25))
print("e) %+d" % (42))
print("f) % d" % (42))
print("g) %+d" % (-42))
print("h) %+2d" % (42))
print("i) % 2d" % (42))
print("j) %2d" % (42))
print("k) %05d" % (42))
print("l) %-5d" % (42))
print("m) %5d" % (42))
```

Saída:

```
a)  0X2F
b)   2F
c) 0X002F
d) 0o31
e) +42
f)  42
g) -42
h) +42
i)  42
j) 42
k) 00042
l) 42
m)   42
```

Entrada

- `input()`: recebe um dado informado pelo usuário.
`usuario = input("Login:")`
`senha = input("Senha:")`
`print('%s tem a senha %s' %(usuario, senha))`
- Atenção: `input` retorna strings.
`x=input('Informe um número:')`
`print(type(x))`
- Conversão de string para numérico
`x=int(input('Informe um número:'))`
`print(type(x))`
- Conversão de numérico para string
`x=str (10)`
`print(type(x))`

Conversões

- Inteiros
 - `int(x)` → x numérico ou string
 - `int(x, base)`
 - Exemplos
 - `Y=int('101')`
 - `print(int('101',2))`
- Reais
 - `float(x)` → x numérico ou string
 - `float('+1.23')` → 1.23
 - `float(' -12345\n')` → -12345.0
 - `float('1e-003')` → 0.001
 - `float('+1E6')` → 1000000.0
 - `float('-Infinity')` → -inf

Conversões

- Complexo
 - `complex(real)` → numérico ou string (sem espaço)
 - `complex(real, imag)` → apenas numéricos
 - `complex('3+4j')`
 - `complex(3,4)`
- Binário
 - `bin(x)` → inteiro
 - a saída é uma string
- Booleano
 - `bool(x)` → 0 ou 1
 - `bool(1)`
 - `bool('false')`
- String
 - `str(x)` → converte x em string
 - `str('114+6')`