



# CK0211 - Fundamentos de Programação: Conceitos Básicos

Emanuele Santos

Bibliografia: Ascencio, Cap. 1, Seções 1.5 a 1.11

# Objetivos

- Apresentar o conceito de variável
- Conhecer os tipos de dados mais comuns usados em linguagens de programação
- Aprender como formar identificadores
- Apresentar uma visão geral de algumas linguagens de Programação
- Familiarizar-se com o ambiente de programação no Linux
- Codificar alguns algoritmos em Python

# Conceito de variável

- Um algoritmo e, posteriormente, um programa, recebem dados, que precisam ser armazenados no computador para serem utilizados no processamento
- Uma variável representa uma posição na memória e pode ter **tipo** (inteiro, caractere, real), tamanho (16, 32 bits, ...) e **nome** definidos
- Utiliza-se o termo variável pois o seu **valor** pode ser alterado durante a execução do programa no qual ela é utilizada
- Exemplos: temperatura em Fahrenheit, temperatura em Celsius, média, etc.

# Analogia com gavetas

- O armazenamento é feito na **memória**
- Podemos imaginar essa “memória” como sendo um armário repleto de gavetas
- As gavetas seriam os locais físicos responsáveis por armazenar as informações
- As variáveis são diferenciadas através de **identificadores**
- Uma **variável** pode guardar apenas um **valor** de cada vez, normalmente as informações são sempre do mesmo **tipo**

# Formação de identificadores

- Nomes utilizados para referenciar variáveis, funções ou vários outros objetos definidos pelo programador
- Regras básicas para a formação de identificadores são:
  - Os caracteres que podem ser utilizados:
    - Dígitos, letras maiúsculas e minúsculas, e sublinhado \_
    - O primeiro caractere não pode ser dígito
    - Não são permitidos espaços em branco e caracteres especiais (@, \$, +, -, %, !)
    - Nem palavras reservadas (keywords)

# Formação de Identificadores

- Exemplos de identificadores válidos
  - A
  - a
  - nota
  - NOTA
  - X5
  - nota\_1
  - \_nota
  - tempo

# Palavras reservadas em Python

## 2.3.1. Keywords

The following identifiers are used as reserved words, or *keywords* of the language, and cannot be used as ordinary identifiers. They must be spelled exactly as written here:

|        |          |         |          |        |
|--------|----------|---------|----------|--------|
| False  | class    | finally | is       | return |
| None   | continue | for     | lambda   | try    |
| True   | def      | from    | nonlocal | while  |
| and    | del      | global  | not      | with   |
| as     | elif     | if      | or       | yield  |
| assert | else     | import  | pass     |        |
| break  | except   | in      | raise    |        |

[http://docs.python.org/reference/lexical\\_analysis.html](http://docs.python.org/reference/lexical_analysis.html)

# Formação de identificadores em Python

- A versão 3 de Python já aceita caracteres acentuados
- Mas como a maioria das linguagens ainda não aceita, então recomenda-se não utilizá-los em identificadores

# Formação de Identificadores

- Por que os identificadores abaixo são inválidos?

5b

A:B

e 12

and

x-y

while

nota/2

# Tipos de dados

- 3 tipos básicos
  - Numérico (Inteiro ou Real)
  - Literal ou Caractere
  - Lógico

# Tipos Numéricos: Inteiro

- Toda e qualquer informação numérica que pertença ao conjunto dos números **inteiros** (negativa, nula, positiva)
- Exemplos
  - 35
  - 0
  - -78

# Tipos Numéricos: Real

- Toda e qualquer informação numérica que pertença ao conjunto dos números **reais** (negativa, nula, positiva)
- Exemplos
  - 5.5
  - -34.89
  - 0.0

Atenção para o delimitador de casas decimais: em programas de computador se utiliza a notação inglesa (.)

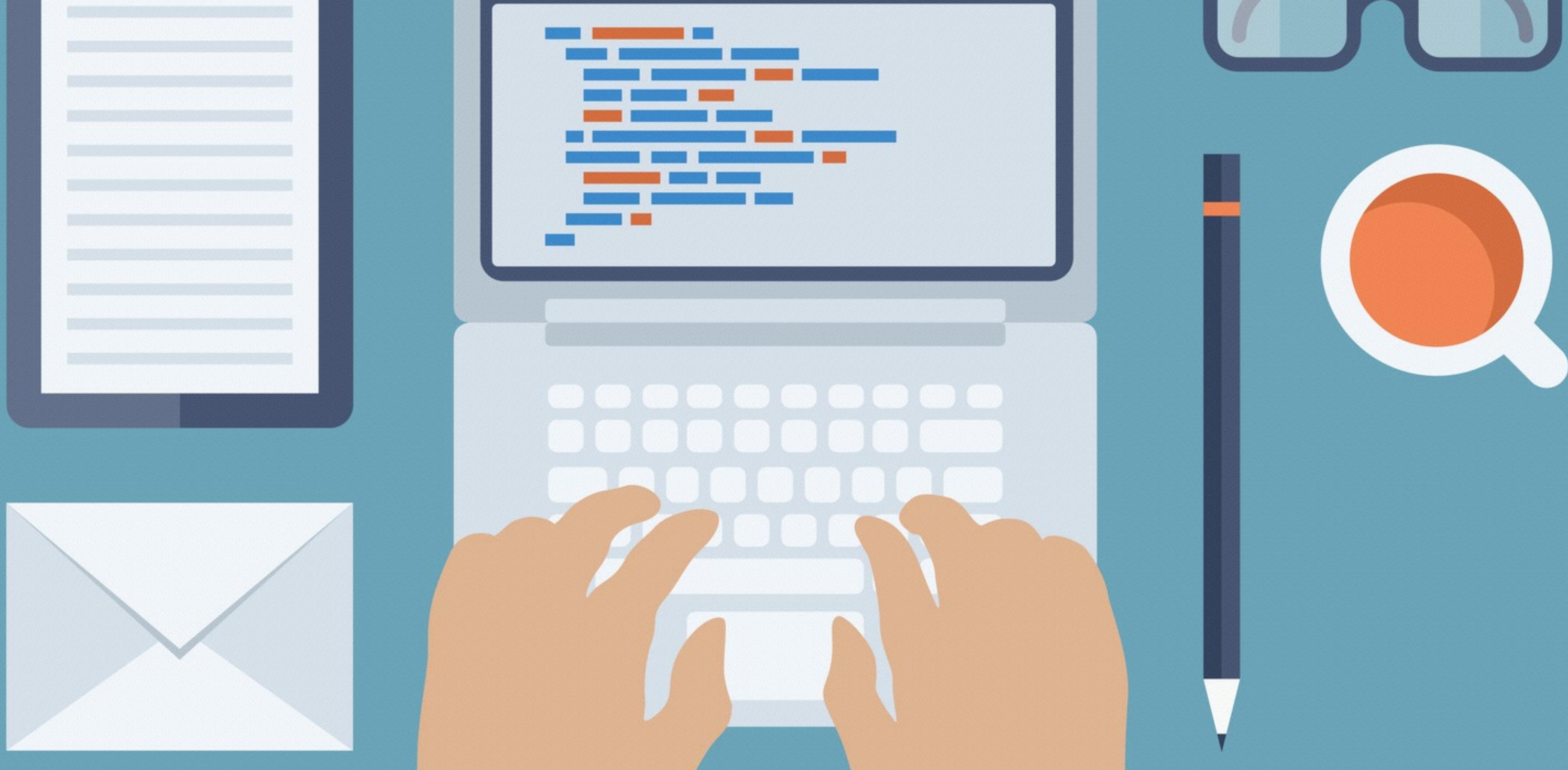
# Tipo Literal ou Caractere

- Toda e qualquer informação constituída de caracteres alfanuméricos (dígitos, letras e símbolos)
  - Exemplos
    - “aluno”
    - “1 + 5”
    - ‘A’
- Em Python:**

  - qualquer símbolo pode ser usado dentro da cadeia de caracteres (string), inclusive caracteres acentuados
  - não existe diferença entre caractere e cadeia de caracteres; um caractere é uma cadeia de tamanho 1
  - Aspas simples ou duplas podem ser utilizadas para delimitar strings

# Tipo Lógico

- São também chamados de dados booleanos (vindos da álgebra de Boole) e podem assumir os valores verdadeiro ou falso.
- True
- False



# PARTE PRÁTICA

# O que vamos fazer...

- Compilar e executar um programa em C
- Escrever o programa equivalente em Python
- Implementar o programa que converte temperatura Fahrenheit para Celsius em Python
- Implementar outros programas

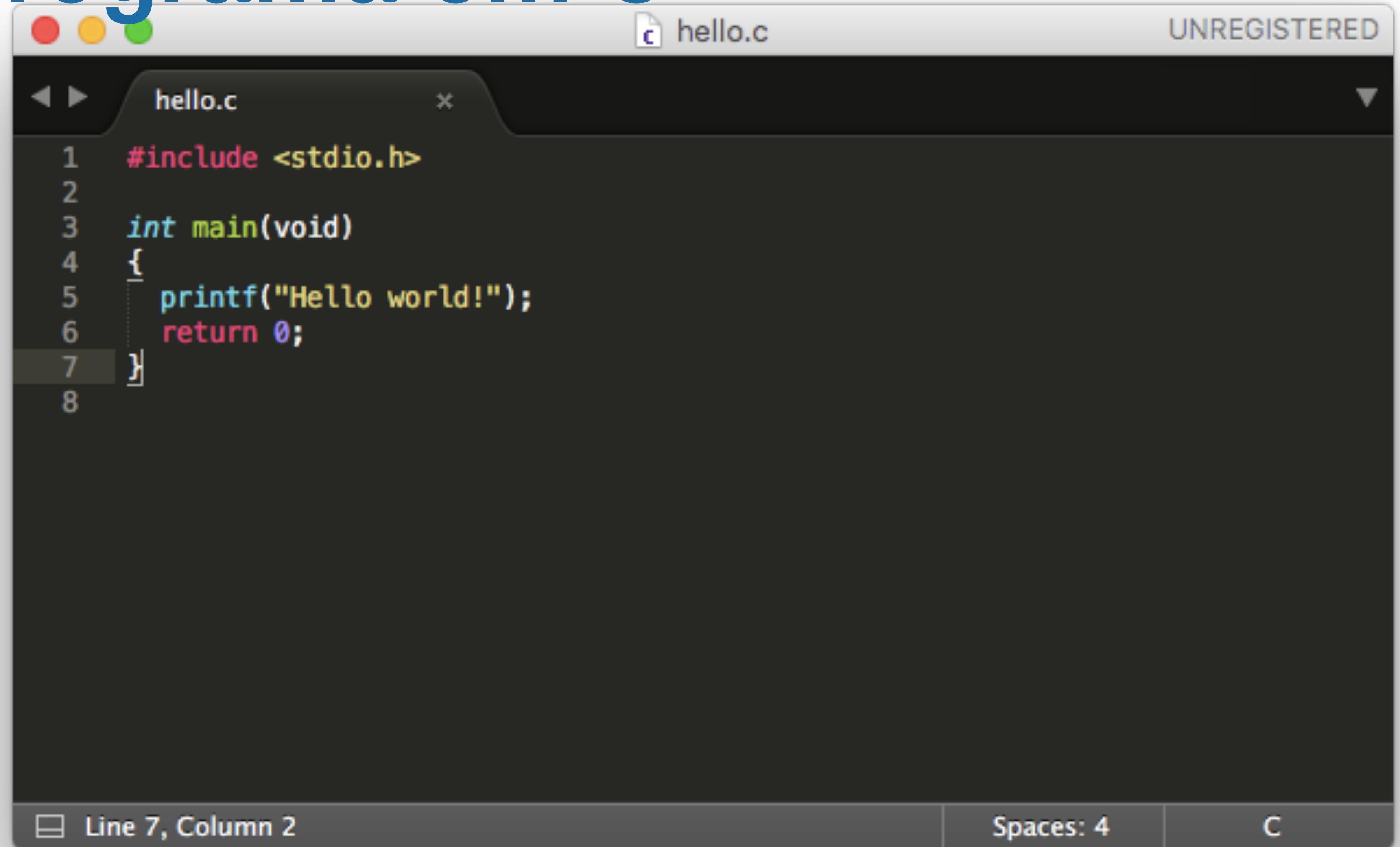
# Por onde começar?

- No Linux:
  - Crie uma pasta no seu diretório de usuário para guardar os arquivos da disciplina (Por exemplo, **fup**)
  - Abra um terminal e digite

```
cd fup
subl hello.c
```

# O primeiro programa em C

- Digite o código ao lado no seu arquivo hello.c
- Depois salve e feche o arquivo



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

The image shows a screenshot of a code editor window titled "hello.c". The code editor has a dark theme with syntax highlighting. The code itself is a simple "Hello World" program. The status bar at the bottom indicates "Line 7, Column 2".

# Para compilar e executar o programa

- No terminal, dentro da pasta fup digite:

```
gcc -o hello hello.c  
./hello
```

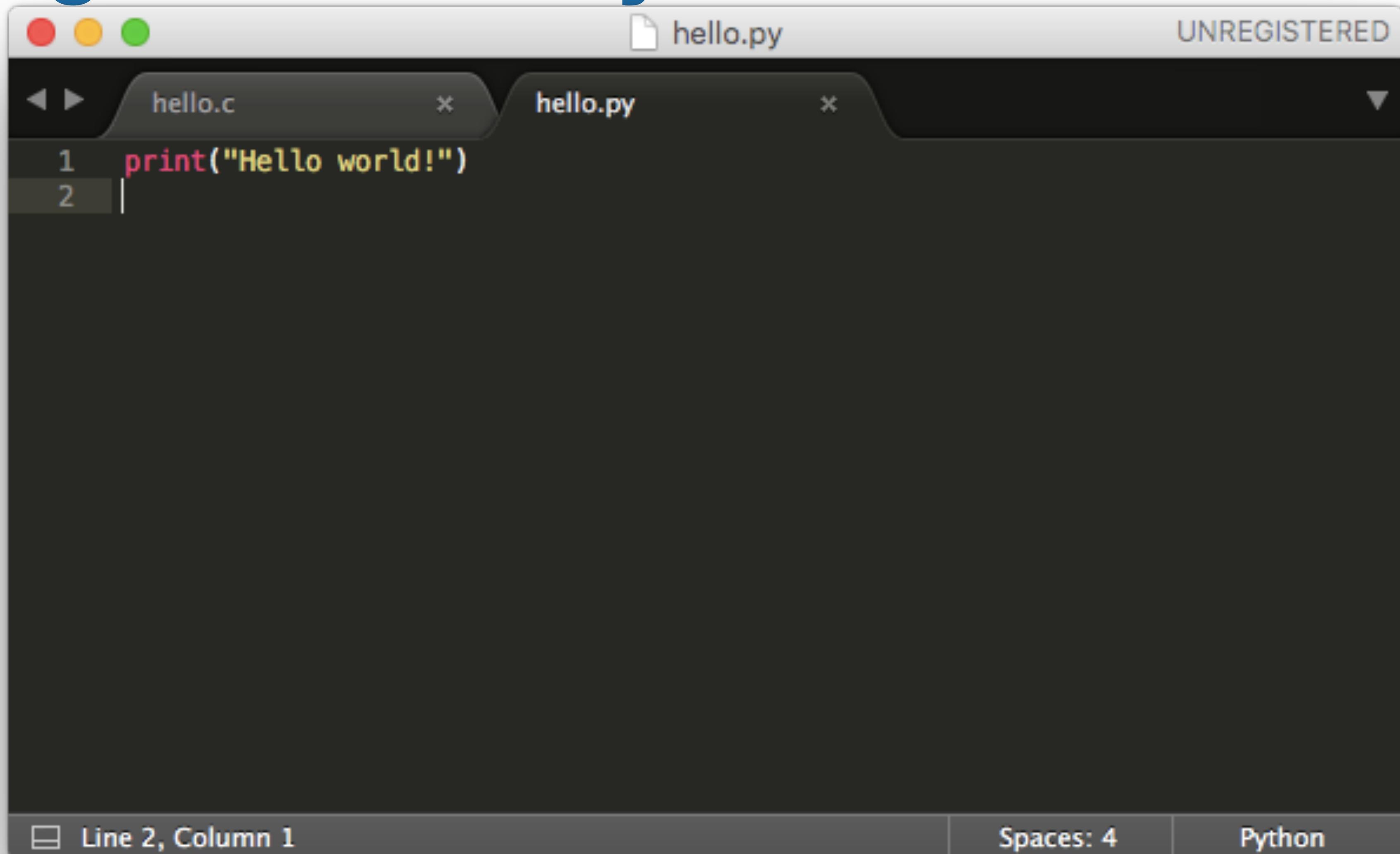
# O mesmo programa em Python

- Na mesma janela do terminal, digite:

```
subl hello.py
```

# O primeiro programa em Python

- Digite o código ao lado no seu arquivo hello.py
- Depois salve e feche o arquivo



```
1 print("Hello world!")
```

# Para executar o programa

- No terminal, dentro da pasta fup digite:

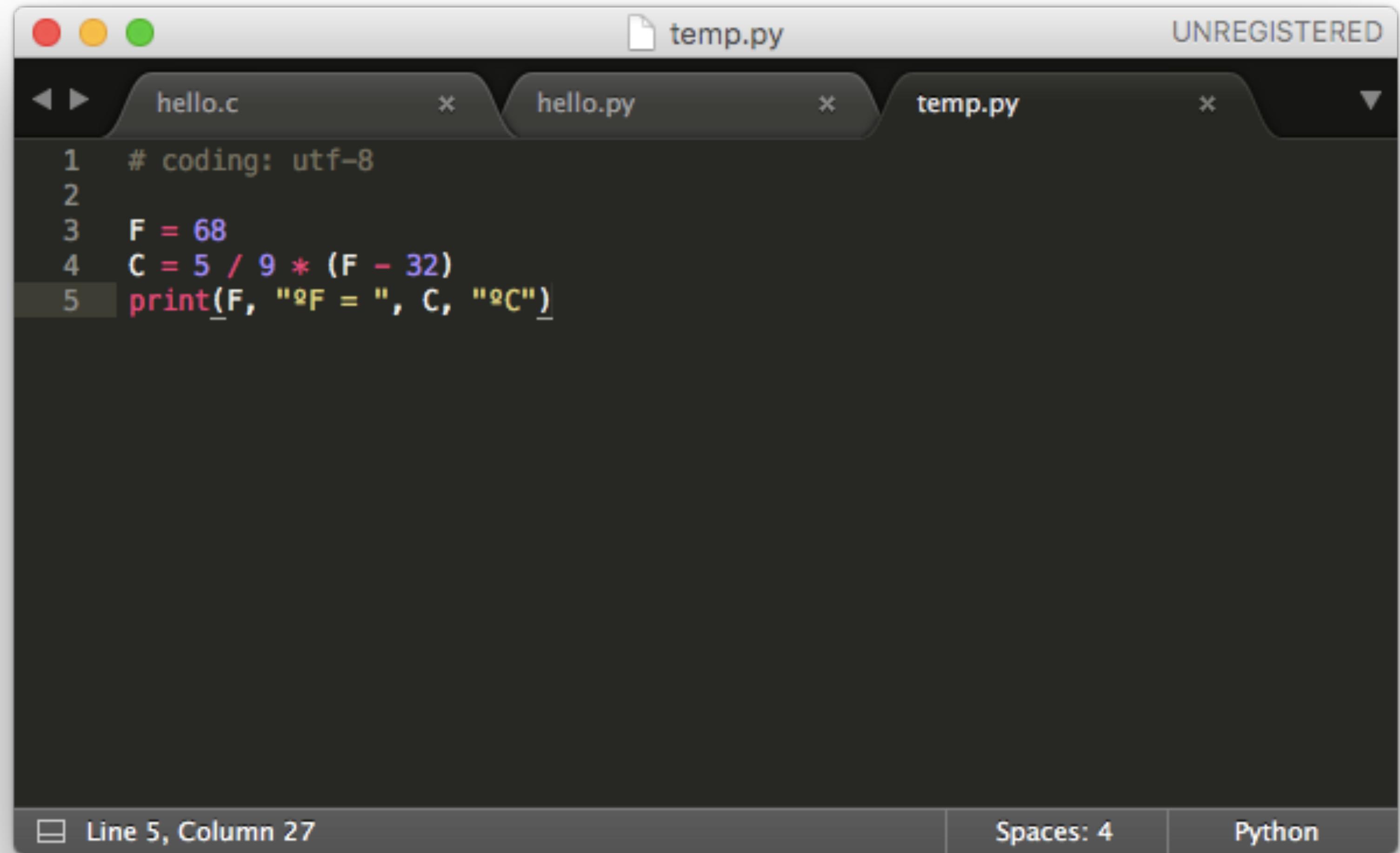
```
python3 hello.py
```

# Convertendo temperaturas

- Para os próximos exemplos, abra o editor de textos Sublime diretamente a partir do menu iniciar do Linux (é só digitar Sublime na barra de texto)
- Depois crie um arquivo chamado temp.py dentro da pasta fup

# Convertendo temperaturas

- Digite o código ao lado e salve o arquivo



```
# coding: utf-8
F = 68
C = 5 / 9 * (F - 32)
print(F, "°F = ", C, "°C")
```

Line 5, Column 27      Spaces: 4      Python

# Para executar o programa

- No terminal, dentro da pasta fup digite:

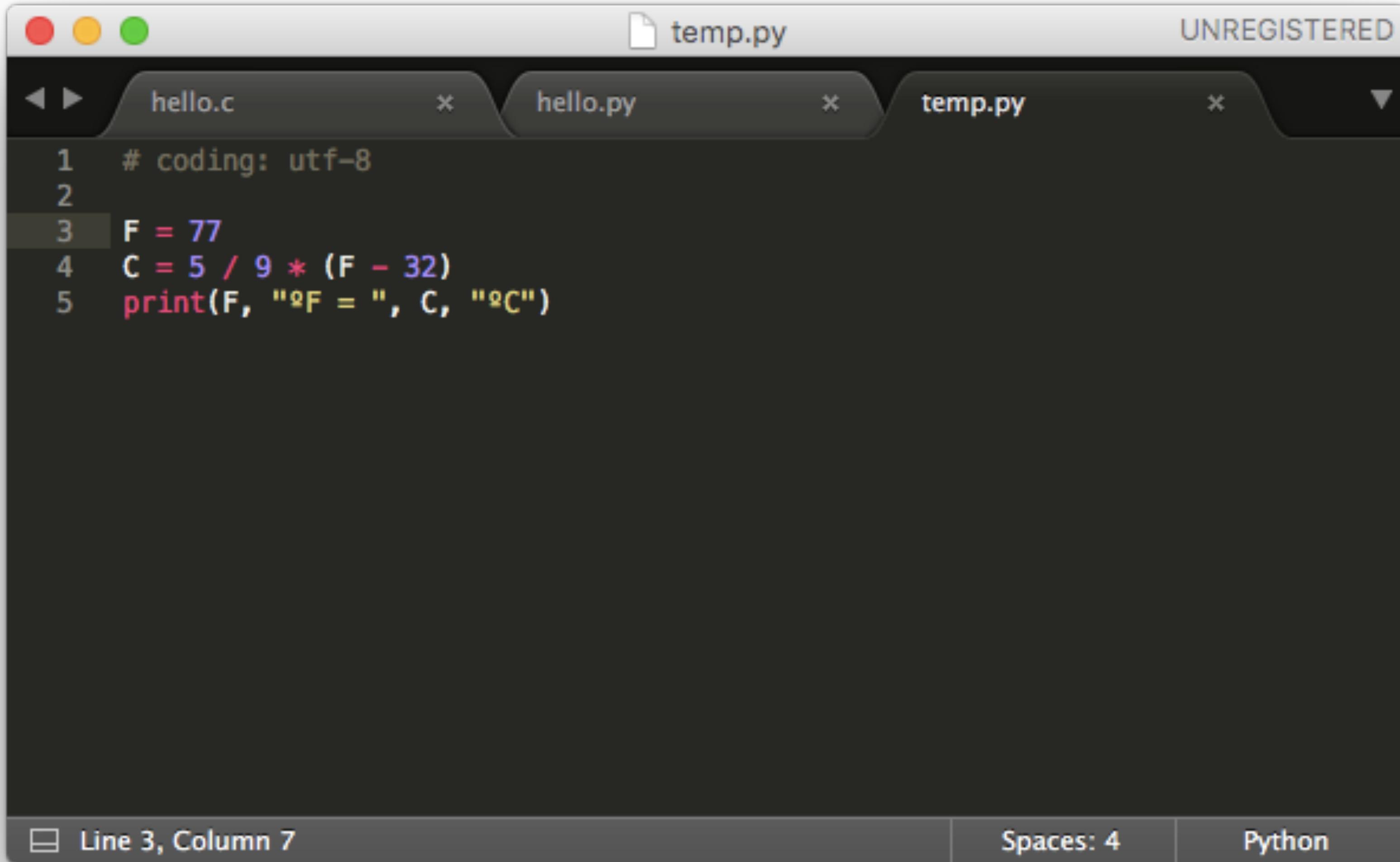
```
python3 temp.py
```

- Você deve ver o seguinte resultado:

```
68 °F = 20.0 °C
```

# Convertendo temperaturas

- Para mudar a temperatura, é só fazer F receber um outro valor
- Depois salve o programa e execute-o novamente



```
# coding: utf-8
F = 77
C = 5 / 9 * (F - 32)
print(F, "°F = ", C, "°C")
```

Line 3, Column 7      Spaces: 4      Python

# Para executar o programa

- No terminal, dentro da pasta fup digite:

```
python3 temp.py
```

- Você deve ver o seguinte resultado:

```
77 °F = 25.0 °C
```

# Entrada de dados

- Usaremos a função **input**
- **input** mostra um literal passado na função e retorna um literal contendo o que foi digitado pelo usuário (sempre do tipo texto)
- guardamos o valor retornado pela função input em uma variável

```
nome = input('Qual o seu nome? ')
print('Bom dia,', nome)
```

# Entrada de dados

- Para forçar que os valores digitados pelo usuário sejam outros além de literal, precisamos converter o valor retornado pela função **input**
- Usamos
  - `int(...)` para converter para inteiro
  - `float(...)` para converter para real

```
idade = int(input("Quantos anos voce tem? "))
```

# Exercício 1

- Use a função input para pedir ao usuário que entre com a temperatura. Converta a entrada para float antes de guardar o resultado em F
- Salve e execute o seu programa

```
Entre com a temperatura em Fahrenheit: 86
86.0 °F = 30.0 °C
```

# Exercício 2

- Faça um programa que multiplica 2 números inteiros entrados pelo uauário
- Crie um novo arquivo chamado mult.py
- Depois altere o programa para multiplicar 2 números reais

# Exercício 3

- Faça um programa que recebe duas notas de um aluno e mostra a média das notas
- Crie um novo arquivo chamado media.py

# Exercício 4

- Faça um programa que calcula o Índice de Massa Corporal (IMC) de uma pessoa a partir de sua altura em metros e o seu peso em Kg, como dado na fórmula:

$$imc = \frac{peso}{altura \times altura}$$

- Crie um novo arquivo chamado imc.py