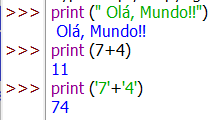
Para baixar o pyhton no seu computador você pode acessar via (<https://www.python.org/downloads/>) e nele baixar o python e a idle para programar sem utilizar uma IDE.

*Primeiros Comandos:*

**Operação básica:**

Print (‘Olá,Mundo!’) -> mostrar na tela, são apenas mensagens.

Se você quiser fazer operações, no python é apenas colocar os números!

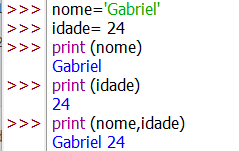


No python eu posso indentar com ‘,’ também. Exemplo é indentar uma variável com uma string.

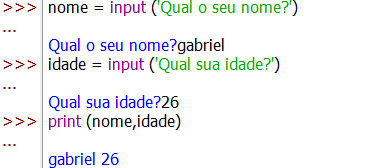


Escreva sempre as **variáveis em letras minúsculas**!

No python todas as variáveis são um objeto.



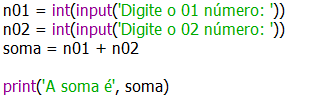
Como eu pergunto para o usuário? Utilizo Input!



**Criando Scripts**:

Vou em file e new file na idle.

Para executar é só ir em run.



*IDE utilizada no curso: PyCharm.*

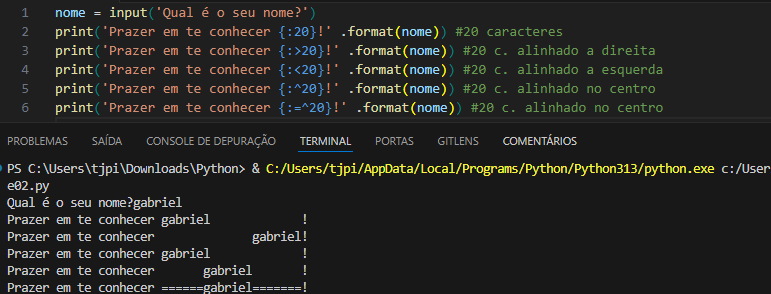
No python tem as saídas formatadas! O que seria isso? Ex:



No caso, nas ‘{}.format()‘ sairá o nome formatado.

Ou colocar assim print(f'Olá {nome}! Seja bem-vindo!')

*Outras coisas que da para fazer usando o ‘{}.format()’:*

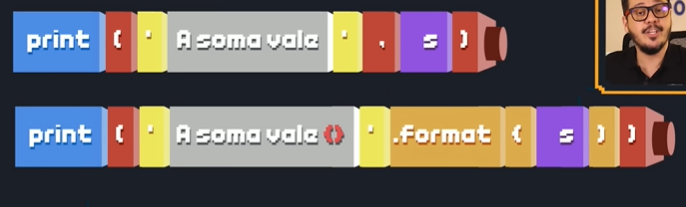


**Tipos Primitivos:**

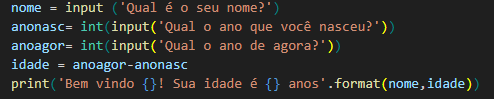
1. **int()** – nº inteiro.
2. **float()** – nº real.
3. **bool()** – valores lógicos.
4. **str()** – caracteres ou string.

Para verificar o tipo é apenas colocando print(type()).



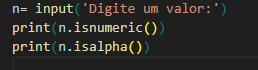


Ex:

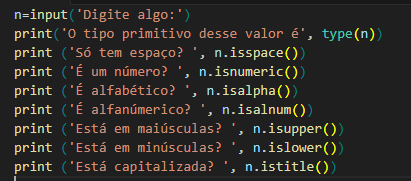


Para saber se um valor pode ser convertido em numérico utiliza-se **‘.isnumeric()’**

Para saber se um valor pode ser convertido em letra utiliza-se **‘.isalpha()’**



Existe vários .is, apenas verifique o que melhor se aplique.



**Operadores Aritméticos:**

1. +
2. –
3. \*
4. /
5. \*\*
6. //
7. %

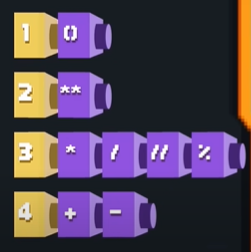


**Igual para Python se escreve ‘==’**

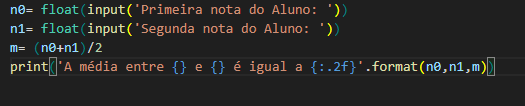


Potencia também pode ser feito utilizando a função **pow().**

**Ordem de precedência**:



EX:



*.2f == duas casas decimais* ***após o ponto flutuante****.*

**Módulos:**

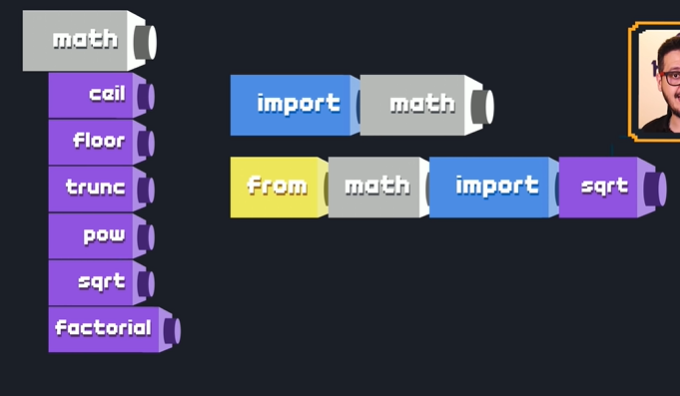
São importações que podem ser feitas de pacotes de fora.

EX: import bebida

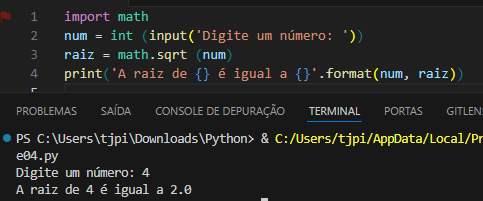
Mas se eu não quiser todas as bebidas? Eu utilizo from bebida import vodka

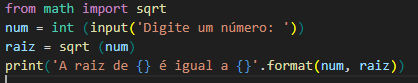
Uma grande biblioteca que é utilizada é a *math*:

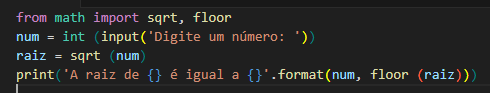
* ceil – Arredondar para cima
* floor – Arredondar para baixo
* trunc – Eliminar da virgula para frente
* pow – Potencia
* sqrt – Calcular raiz quadrada
* factorial – Calcular fatorial.
* hypot – Calcular hipotenusa.



Se eu quiser importar mais que a sqrt é só colocar ‘,’ e a que eu quero importar.

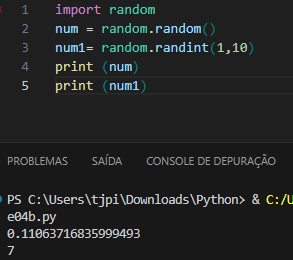


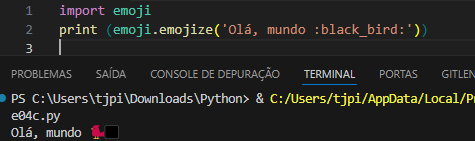




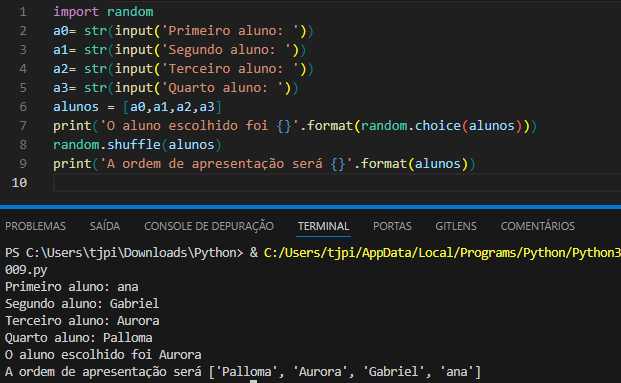
Bibliotecas podem ser importas: [*https://docs.python.org/3.13/library/index.html*](https://docs.python.org/3.13/library/index.html)- No qual esses são módulos feitos pelo Python. OU <https://pypi.org/> - Módulos feitos pela comunidade.

*Ex: random (da biblioteca padrão) e emoji (biblioteca da comunidade)*

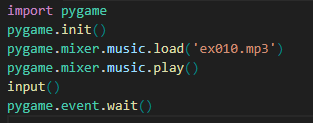




Desafio:



Tocando musica no Python:



**Manipulando cadeia de caracteres:**



Operações que podem ser feitas:

1. **Fatiamento** – Pegar pedaços dela.
   1. frase[9] ele vai pegar só o V
   2. frase[9:13] ele vai pegar o V até o E (excluindo o 13), sempre 1 a menos no final.
   3. frase[9:21:2] ele vai pegar o V até o N pulando de 2 em 2.
2. **Analise** – Saber informação sobre a frase.
   1. *len*(frase) – saber a quantidade de caracteres (21 caracteres)
   2. frase.*coun*t*(‘o’)* – contar quantas vezes aparece o ‘o’ (3 caracteres)
      1. frase.count(‘o’,0,13) – pedindo para contar 0 até o 13 (excluindo o 13).
   3. frase.*find*(‘deo’) –aonde encontrou o texto mencionado. # Se ele responder -1 é pq não existe.
   4. ‘curso’ *in* frase – verificar se existe a palavra curso na frase. (true ou false)
3. **Transformação** – mudar atravês de metodos.
   1. frase.*replace* (‘Python’,’Android’) – troca a palavra 0 por 1.
   2. frase.*upper*() – colocar tudo em maisculo.
   3. frase.*lower*() – colocar tudo em minusculo.
   4. frase.*capitalize*() – coloca só a primeira letra em maisculo.
   5. frase.*title*() – coloca tudo após os espaços em maisculo.



* 1. frase.*strip*() – remove espaços inuteis (0:2 e 17,18)
  2. frase.*rstrip*() – remove do lado direito.
  3. frase.lstrip() – remove do lado esquerdo.

1. **Divisão**
   1. frase.split() – divisão na string entre os espaços.



* 1. ‘-‘.join(frase) – juntar as strings.

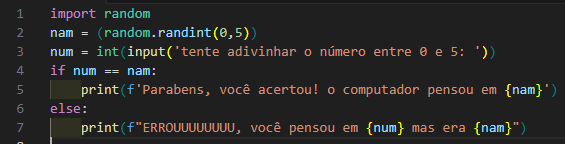
**Condições:**



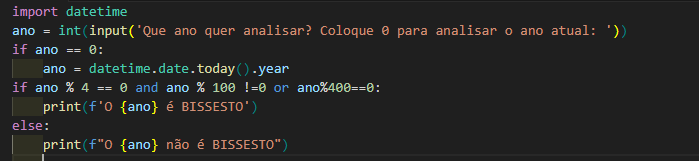
Mas tem como fazer uma condição simplificada:



D001:



D002:



**Cores no terminal:**

Sempre que quer utilizar cores utiliza os codigos da cores: \033[(aqui o codigo da cor)m







*Estilos de Texto*

0: Resetar todas as configurações

1: Negrito/Brilhante

2: Escurecido (dim)

3: Itálico

4: Sublinhado

5: Piscando

7: Inversão (troca cor de fundo e texto)

8: Oculto

9: Tachado

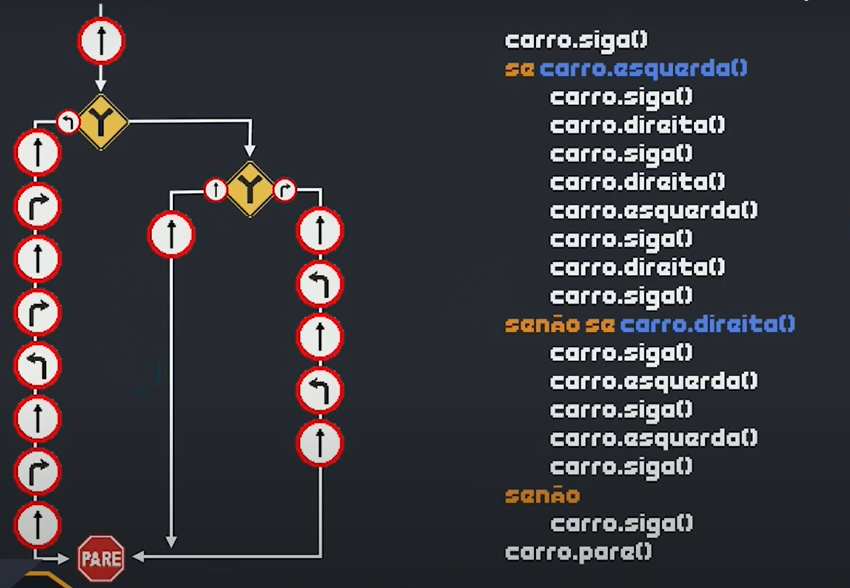
*Cores de Texto*

|  |  |  |
| --- | --- | --- |
| **Cor** | **Texto Normal** | **Texto Brilhante** |
| Preto | \033[30m | \033[90m |
| Vermelho | \033[31m | \033[91m |
| Verde | \033[32m | \033[92m |
| Amarelo | \033[33m | \033[93m |
| Azul | \033[34m | \033[94m |
| Magenta (Roxo) | \033[35m | \033[95m |
| Ciano (Azul Claro) | \033[36m | \033[96m |
| Branco | \033[37m | \033[97m |

*Cores de fundo*

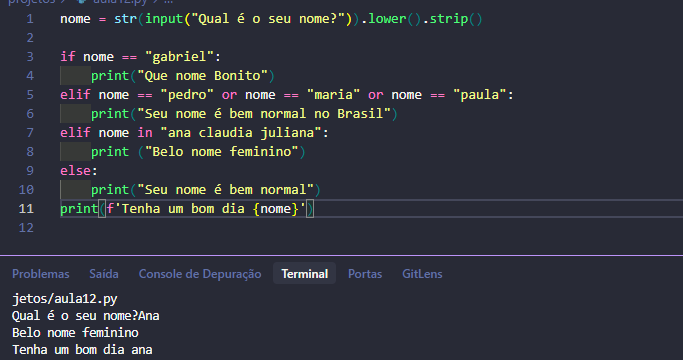
|  |  |  |
| --- | --- | --- |
| **Cor** | **Fundo Normal** | **Fundo Brilhante** |
| Preto | \033[40m | \033[100m |
| Vermelho | \033[41m | \033[101m |
| Verde | \033[42m | \033[102m |
| Amarelo | \033[43m | \033[103m |
| Azul | \033[44m | \033[104m |
| Magenta (Roxo) | \033[45m | \033[105m |
| Ciano (Azul Claro) | \033[46m | \033[106m |
| Branco | \033[47m | \033[107m |

**Condições Aninhadas:**









Desafios:

