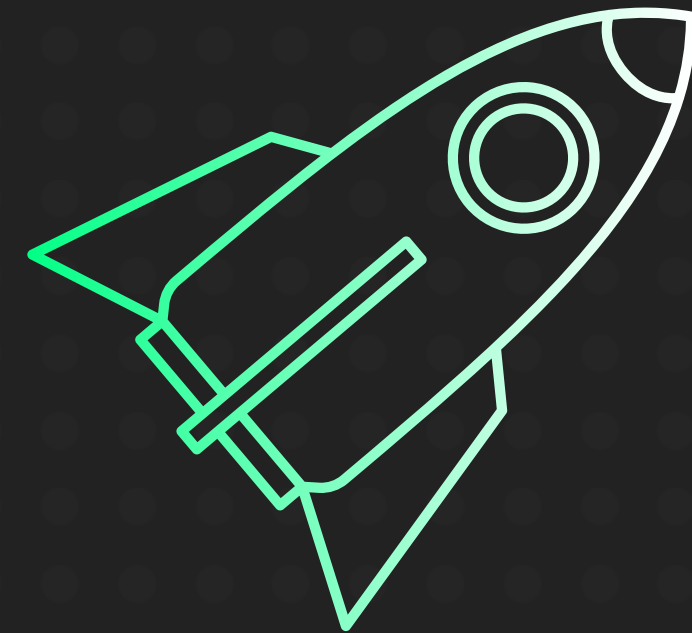
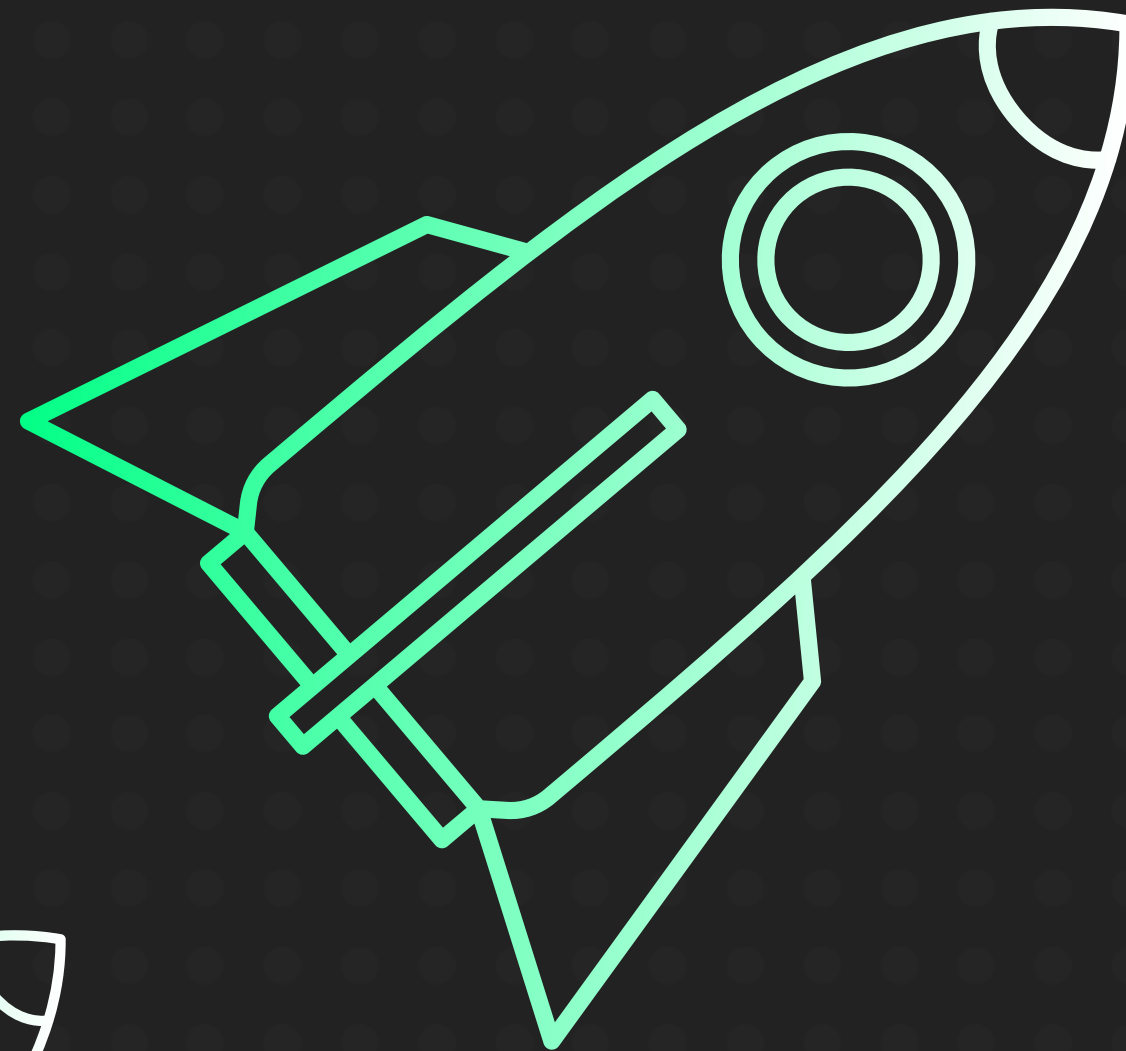


PYTHON

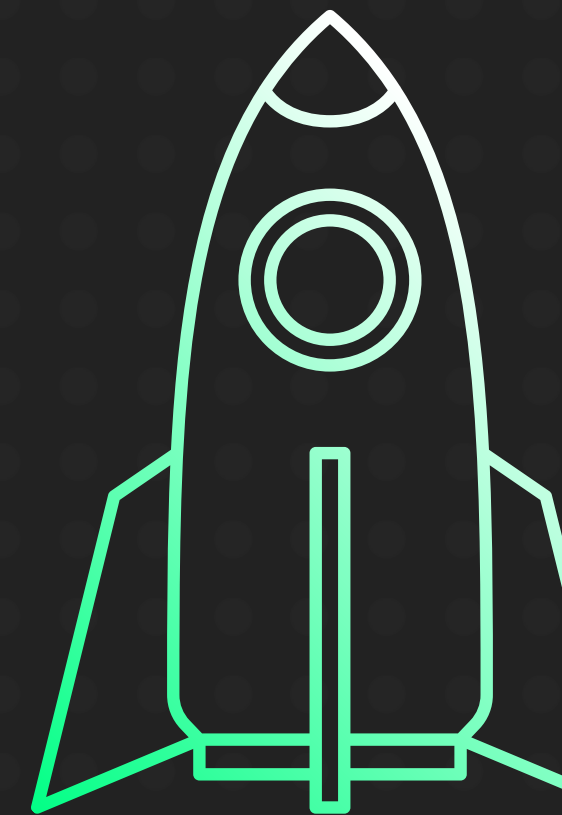
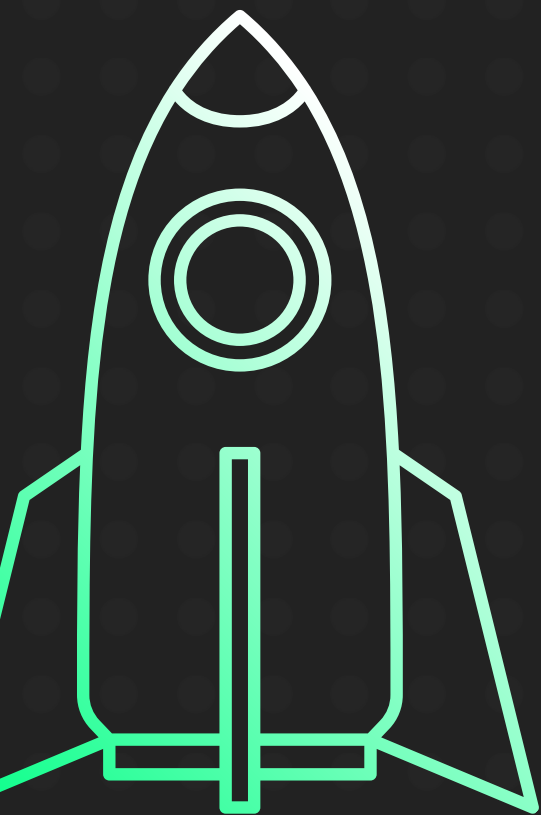
WORLD I



MIGUEL MENDES

SOBRE

Compilação de aprendizados do curso de Python para
consulta em momentos posteriores.

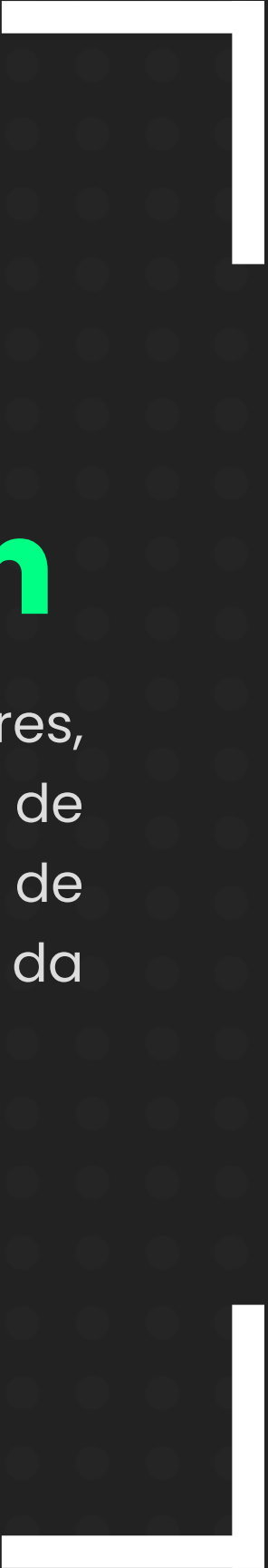




```
import this
```

The Zen of Python

É uma coleção de 19 princípios orientadores, na forma de poema, com uma série de aforismos, para escrever programas de computador que influenciam o design da linguagem de programação Python.



```
print('Miguel é lindo! xD')
```

print

Comando usado para inserir texto na tela.
Observação: dentro dos parenteses pode ser usada aspas simples (') ou aspas duplas (").

```
print(4 * 4)
```

print

Para realizar cálculos, insira a conta dentro de print sem nenhum tipo de aspas.

```
print('2' + '2')
```

print

Nesse cenário vemos que os números não resultaram em uma soma. Caso execute o resultado será 22. Isso se chama concatenação, resultado conseguido quando se "junta" duas strings.

```
print(n.isnumeric())
```

print

Condição que avalia se o valor da variável é número.

```
print(n.isalpha())
```

print

Condição que avalia se o valor da variável é algarismo.


```
print(n.isupper())
```

print

Comando que diz se o valor da variável está em maiúsculo.

```
print(n.lower())
```

print

Comando que diz se o valor da variável está em minúsculo.

```
print(n.isalnum())
```

print

Condição que avalia se o valor da variável é algarismo e número.

```
print(nome[7])
```

print

Nesse print colocamos a variável e um valor entre colchetes para exibir a letra equivalente a 7 na sequência de caracteres da String.

```
print(nome[7:13])
```

print

Nesse print segue o mesmo princípio, porém agora irá exibir os caracteres de 7 a 12, ignorando a última casa.

```
print(nome[7:13:2])
```

print

Nesse print exibirá os caracteres de 7 a 12, com salto de 2 em 2.

```
print(nome[:7])
```

print

Nesse print como não exibi valor de início, por padrão, começará do 0 e mostrará a sequência de caracteres de 0 a 6.

```
print(nome[14:])
```

print

Nesse print o valor de início é 14 e seguirá até o último caracter, independente do tamanho da sequência.


```
print(nome[0::4])
```

print

Nesse print começará do zero, irá até o final, com salto de 4.

```
print(nome.count('e', 0, 13))
```

print

Mostra a quantidade de caracteres colocado como valor, como no exemplo acima. É possível usar o fatiamento.

```
print(nome.find('igu'))
```

print

Mostra onde começa determinado conjunto de caracteres. Se a função retornar o valor (-1) significa que a sequência de caracteres não existe dentro da String.

```
print('Mendes' in nome)
```

print

Confirma se a palavra em questão está na String.

```
print(nome.replace('Mendes', 'Black'))
```

print

Substitui em "segundo plano" alguma palavra na string. Não altera a string propriamente, apenas altera a exibição.

```
print(nome.upper())
```

print

Deixa todos os caracteres em maiúsculo.

```
print(nome.lower())
```

print

Deixa todos os caracteres em minúsculos.

```
print(nome.capitalize())
```

print

Pega todos os caracteres da String e somente a primeira letra de toda String fica maiúscula.


```
print(nome.title())
```

print

Pega todas as primeiras letras depois de um espaço e coloca elas maiúsculas.

```
print(nome.strip())
```

print

Remove todos os espaços inúteis no começo e fim da String. Se configurar "print(nome.rstrip())" remove somente os espaços do lado direito e "print(nome.lstrip())" somente os espaços do lado esquerdo.

```
print(nome.split())
```

print

Ocorre uma divisão dentro da sua String considerando os espaços. Ele gera tecnicamente uma lista com cada elemento será separado pelo espaço.

```
print(' '.join(nome))
```

print

Junta todos os caracteres novamente.

```
print(len(nome))
```

print

Lê a quantidade de caracteres da sua String.

```
variavel = input('Digite seu nome: ')
```

input

Quando queremos que a nossa entrada de dados parta da interação com usuário, usamos a função `input()`, ou seja, é um artifício em programação que utilizamos quando precisamos que o usuário passe ao programa algum tipo de dado.

```
n1 = float(input('Digite um valor: '))
```

input

Para atribuirmos um valor ao input, podemos incluir na linha seu tipo de variável, otimizando o código.

```
tempo = int(input('Digite a idade do seu  
carro: '))
```

```
if tempo <= 3:  
    print('Seu carro é novo!')  
else:  
    print('Seu carro é velho!')  
print('-----FIM-----')
```

```
# print('Carro novo' if tempo <= 3 else  
'Carro velho') --- MODO SIMPLIFICADO
```

if else

A estrutura if-else é um conceito fundamental na programação que permite que um programa tome decisões com base em condições específicas. Essa estrutura condicional permite que partes específicas do código sejam executadas ou ignoradas, dependendo se uma determinada condição é verdadeira ou falsa.


```
resultado = valor_entrada * valor_entrada  
print('O resultado é {}'.format(resultado))
```

.format()

Utiliza os marcadores de posição '{}' para indicar onde os dados formatados serão inseridos. Para limitar as casas decimais de um valor no .format(), nas chaves digite {:.2f}.

```
n1 = 7
n2 = 77
cores = {'limpa': '\033[m',
        'azul': '\033[34m',
        'amarelo': '\033[33m',
        'pretoebranco': '\033[30m'}

print('Os número são {}{}{} e {}{}{}.'
      .format(cores['pretoebranco'], n1,
              cores['limpa'], cores['azul'], n2,
              cores['limpa']))
```

cores in Python

Existem diversas maneiras de colocar cores no seu código em Python. Ao lado, é uma delas, de modo mais organizado.

```
import math
```

import

É um módulo que nos dá acesso a várias funções matemáticas, como raiz quadrada, seno, arredondamentos, etc. Para chamar um ou dois métodos específicos digite: "from math import (método), (método)"

```
import math
```

```
num = int(input('Digite um número: '))  
raiz = math.sqrt(num)
```

```
print('A raiz quadrada de {} é {:.2f}'  
      .format(num, raiz))
```

import

Dentro do módulo de funções matemáticas usamos "math.sqrt" para Raiz Quadrada.

```
from math import trunc
```

```
n1 = float(input('Digite um número  
decimal: '))  
r1 = trunc(n1)
```

```
print('O número {} digitado em sua  
porção inteira é {}'.format(n1, r1))
```

import

Função matemática que arredonda números decimais para sua porção inteira.

```
import random
```

import

Para gerar números aleatórios em Python utilizamos o módulo random.

**MUITO
OBRIGADO!**