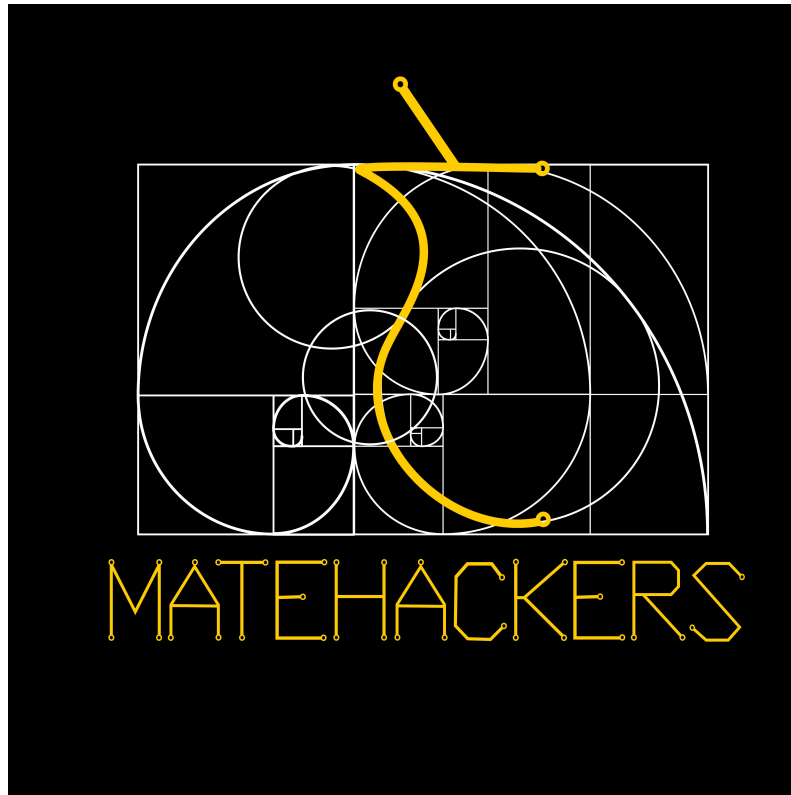


Os primeiros passos em programação usando python!

TODO Para aprender é necessário quase saber [0/15]

TODO Recadinhos [0/6]

- ☐ Obrigado pela presença!



- ☐ Reflexões sobre a ontologia de perguntas bobas.
- ☐ Apresentação baseada no livro <http://learnpythonthehardway.org/>
- ☐ Esse aqui é o <https://matehackers.org/>!

"(...) Você nem imagina o progresso que cada dia me traz. E quando você diz: 'Compartilhe comigo um pouco destes presentes que você acha tão úteis', eu respondo que eu estou ansioso para oferecer todos estes privilégios sobre você, e que eu fico contente em aprender, para que eu possa ensinar.

Nada vai me satisfazer, não importa o quão excelente e benéfico, se eu devo reter o conhecimento sobre tal coisa para mim. E se sabedoria me for dada, sob a expressa condição de que deve ser mantida escondida e não pronunciada, eu recusarei.

Nenhuma boa coisa é agradável de possuir, sem amigos para compartilhá-la."

– **Sêneca**, 4BC~64AD

- ☐ Feedback apreciado. Bônus points for meanies.
- ☐ Pra aprender é necessário quase saber!
- ☐ Case in point: EATDXQVDOSA

TODO Um breve devaneio sobre a linha de comando [0/3]

- ☐ Texto é a melhor interface.
 1. Eficiente
 2. Maior denominador comum entre seres humanos e computadores.
 3. Baixa complexidade.
- ☐ Comandos básicos
 1. **cd**: "change directory" - troca o diretório atual
 2. **ls**: "list" - lista o conteúdo do diretório
 3. **mkdir**: "make directory" - cria um novo diretório
 4. **pwd**: "print working directory" - mostra o diretório atual
- ☐ Novos apelidos para velhos amigos
 1. Você pode chamar o **diretório atual** de `.`
 2. Você pode chamar o **diretório acima do atual** de `..`
 3. Você pode chamar o diretório `/home/<SEU NOME DE USUÁRIO>` de `~`
 4. Você pode chamar o **diretório raiz** de `/`. Todos os outros diretórios estão abaixo dele.
- Exemplos:

```
ls ~
```

```
ls .
```

- ☐ Exercício:
 - Criar um diretório no pasta **home** do seu usuário chamada **programacao** usando a linha de comando.

TODO Primeiro programa: Helo World! [0/5]

- Criando um programa em python do zero!
- ☐ Criando um arquivo texto: **arquivo.txt**
- O que acontece quando outros programas abrem este arquivo?
 1. **firefox arquivo.txt**
 2. **python arquivo.txt**
- ☐ Fazendo algo útil?
 - Modificando o arquivo.txt: **import this**
- ☐ Renomeando o **arquivo.txt** para **programa.py**
 1. **MV**: "move" - move ou renomeia arquivos
- ☐ O seu primeiro comando em python: **print()**
 - **print("Hello World")**
- ☐ Exercício:
 - Modificar o programa.py e deixar somente a linha: **print("Hello World")**

TODO Fazendo de conta? [0/2]

- ☐ A primeira coisa interessante que pode ser feita com uma linguagem de programação são operações aritméticas.
 - Adição: +
 - Subtração: -

- Multiplicação: *
- Divisão: /
- Menor que: <
- Maior que: >
- Menor ou igual: <=
- Maior ou igual: >=
- Igual: =

- ☐ Exemplos

```
print(2+2)
print(9/2)
print(9/2.5)
print(1*7)
print(-5-(-2))
print(10>9)
print(15<12)
print(15<=15)
```

TODO Lembrando o passado. [0/2]

- ☐ Aparentemente computadores conseguem lembrar das coisas!
- ☐ Existe um comando especial em python para lembrar das coisas:
<NOME> = <COISA A SER LEMBRADA>
- Exemplos:

```
# Uso de variáveis e memória.
my_name = "Alan"
my_age = 31
my_height = 1.79 # metros
my_weight = 80 # kilos
my_eyes = "Castanhos"
my_hair = "Escuros"

print(my_name)
print(my_height)
print(my_hair)
print(my_weight)
```

TODO Processando textões. [0/1]

- ☐ - 'Você quis dizer "Springs" - Google' (Novo nome para lembrar: Strings)
- ☐ Strings são sequências de caracteres entre aspas.

```
# Como lidar com strings.
# %d : use com digitos
# %s : use com strings
# %r : use quando você quer só uma _representação_ do valor
print("Vamos falar sobre %s" % my_name)
print("Eu tenho %d metros de altura" % my_height)
print("Ele pesa %d kilos" % my_weight)
print("Pra falar a verdade isso nem é muito pesado")
print("Ele tem os olhos %s e os cabelos %s" % (my_eyes, my_hair))

# Essa linha é complicada
print("Se eu somar %d, %d, e %d isso da %d." % (my_age, my_height, my_weight, my_age +
```

TODO Como interagir com o mundo [0/1]

- ☐ Novo nome para lembrar input

```
# --- Needs a dedicated buffer ---
# Fazendo perguntas ao usuário.
user_age = input("Quantos anos você tem ")
user_weight = input("Quanto você pesa ")
user_height = input("Quanto você mede ")
print("Então você tem %r anos, pesa %r kilos e mede %r metros!" % (user_age, user_weight,
```

- ☐ Exercício: Fazer um programa que pergunte o peso e a altura do usuário e devolva o Índice de Massa Corporal (SE VIREM)
- ☐ Exercício: Fazer um programa que pergunte o peso e a altura e devolva quantas pessoas dessa altura eu preciso empilhar pra chegar até a lua. Diga também quantas pessoas são necessárias para chegar no peso da lua.

TODO Não se -reprima-, digo, repita. [0/4]

- ☐ Dilema #1: Humanos não gostam de fazer trabalho repetitivo.

- ☐ Dilema #2: Humanos tem uma memória de curto prazo pequena.
- ☐ Solução: Compartmentalizar (Funções)

```
# exemplo de chamada de função
print("Hello World")

# exemplo de definição de função
def minha_funcao(mensagem):
    print("Esta é minha mensagem: %s" % mensagem)

# como chamar a função que você definiu?
minha_funcao("FUNCAOOOOOOO SAO DO CORASSSAUUUMMM")
```

- ☐ Exercício: Modificar o código anterior e remover as partes repetidas.

TODO É Lógico. [0/6]

- Introdução a aritimética booleana.
- ☐ O que é lógica?
- ☐ O que são tipos de dados?
 - ☐ Exemplos de tipos de dados:
 - * ☐ Números: **1, 2, -10, 3.0**
 - * ☐ Strings: "Yellow Submarine"
 - * ☐ Booleans : True, False
- ☐ Operadores lógicos em python e suas Tabelas verdades
- ☐ AND

A	B	A and B
True	True	True
True	False	False
False	True	False
False	False	False

- ☐ OR

A	B	A or B
True	True	True
True	False	False
False	True	False
False	False	False

- ☐ NOT

A	not A
True	False
False	True

- ☐ Examples:

```
eu_nasci_ontem = True
voce_consegue_me_enganar = False

print(eu_nasci_ontem and voce_consegue_me_enganar)
print(eu_nasci_ontem or voce_consegue_me_enganar)
print(eu_nasci_ontem and not voce_consegue_me_enganar)
```

TODO Ta na hora de tomar uma decisão. [0/1]

- ☐ SE o curso for bom ENTÃO eu vou senão eu não vou.

```
if eu_nasci_ontem:
    print("Nossa cara, que novo vc")
else:
    print("Nem eu")
```

TODO O mundo da loops. [/]

- ☐ Novo problema: Humanos não gostam de escrever um comando de cada vez.
`bart.gif`
- ☐ Duas maneiras de resolver o problema: FOR e WHILE
- ☐ FOR executa um comando por um número de vezes finita.
- ☐ WHILE executa um comando enquanto uma expressão for True

- ☐ Exercício: (Escrever um simulador de Bart Simpson. (Imprimir 100 vezes na tela: "Family Guy é melhor que Simpsons"))
- ☐ Nova função: `range(n)` - retorna uma lista de valores de 0 até n.

Com o for

```
for i in range(5):
    print("Simpsons é melhor que Family Guy")
```

Com o while

```
contador = 0
while contador < 5:
    print("Family Guy é melhor que Simpsons")
    contador = contador + 1
```

TODO Listas [0/4]

- ☐ Agrupar coisas em sequências é extremamente útil: Permite tratar muitas coisas como uma só
- ☐ `range()` cria uma lista
- ☐ Uma lista está declarada da seguinte forma:

```
lista_vazia = []
lista_de_amigos = [ "Pedro", "Maria", "João" ]
lista_de_idades = [ 24, 23, 33 ]
```

```
for idade in lista_de_idades:
    for amigo in lista_de_amigos:
        print("Meu amigo %s tem %s anos." % (amigo, idade))
```

- ☐ Exercício: Criar uma lista de gêneros e corrigir o programa acima para que ele diga "Meu amigo..." e "Minha amiga..." corretamente.

TODO Arquivos [0/5]

- ☐ Arquivos são uma outra maneira de interagir com o mundo externo.
- ☐ Na prática, tudo é um arquivo.

- ☐ Arquivos possuem um ciclo de vida natural que **deve** ser respeitado:
Abrir -> Ler/Escriver -> **Fechar**
- ☐ Duas funções básicas: `read()` e `write()`
- ☐ Exemplo:

```
with open('arquivo.txt') as f:
    print(f.readline())
```

```
with open('arquivo.txt', 'w') as f:
    f.write("Nananana")
```

```
import os
```

```
for dirpath, dirname, filenames in os.walk('.'):
    print(dirpath, filenames)
```

TODO Comunidade [0/2]

- ☐ Excelente comunidade com muito código aberto.
- ☐ Exemplo

```
import requests
```

```
r = requests.get("https://matehackers.org")
```

TODO Vai planeta [0/1]

- ☐ VOCÊ CHEGOU ATÉ AQUI! PARABÉNS!