

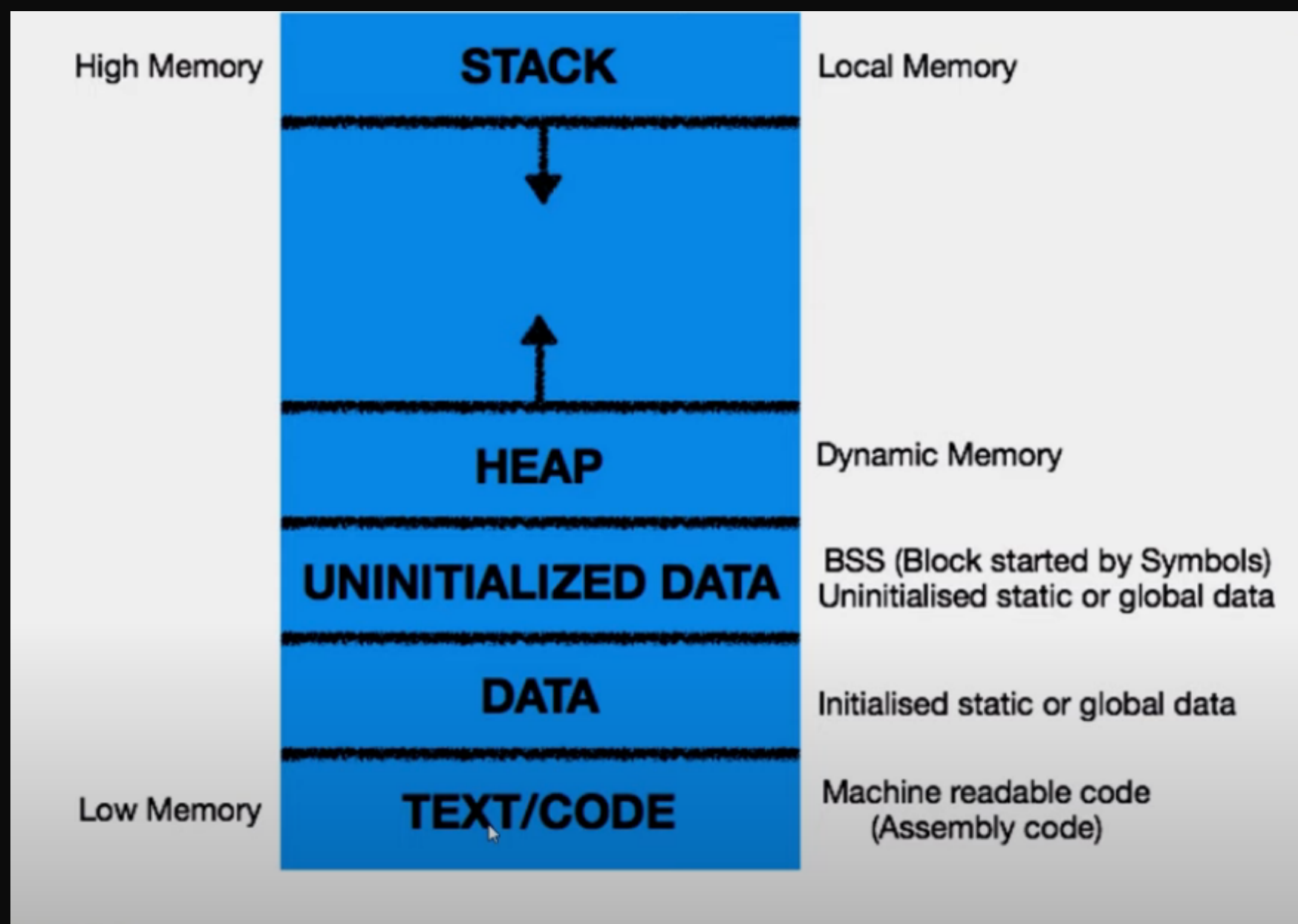


# <Introdução a Python>

COMPILE



# Alocação de Memória em Python



Memória alocada por um processo

Em mais linguagens mais **leves** e **fortemente tipadas** é comum a alocação estática.

Em Python, tudo é dinâmico. Por baixo dos panos, todo o **gerenciamento** de memória é feito pela própria máquina no **HEAP**.

Ou seja, adeus malditos **ponteiros**!





Relembrando...

# Condicional IF e ELSE

Estrutura excludente para tomada de decisões

O **ELSE** só acontece quando a condição do **IF** não passa

O **ELSE** trabalha com booleanos

**Escolpos Definidos  
Pela indentação!!!!**



Se **x** for igual a **y**, então mostre "**Paçoca!**", senão mostre "**Chocolate!**".

```
if x == y:  
    print("Paçoca!")  
else:  
    print("Chocolate!")
```





Relembrando...

# Laços: **While** e **For**

#programa que imprime os números de 1 a 10 usando um loop while

```
num = 1
while num <= 10:
    print(num)
    num += 1
```

#programa que imprime os números de 1 a 10 usando um loop for

```
for num in range(1, 11):
    print(num)
```





Relembrando...

# Funções

# Definindo uma função que verifica se um número é par ou ímpar

```
def par_ou_impar(numero):  
    if numero % 2 == 0:  
        return "O número é par."  
    else:  
        return "O número é ímpar."
```

# Chamando a função com um número de exemplo

```
resultado = par_ou_impar(7)
```

# Exibindo o resultado

```
print(resultado)
```





Relembrando...

# Entrada e Saída de Dados

Toda a entrada de dados no Python são strings! Exigem "cast"

```
nome = input("Digite o seu nome: ")
print("Olá,", nome)
idade = 25
print("Meu nome é", nome, "e eu tenho", idade, "anos.")
```

## Usando "map" e "cast"

```
num1, num2 = map(int, input("Digite dois números inteiros separados por espaço: ").split())
resultado = num1 + num2
print("A soma de", num1, "e", num2, "é", resultado)
```





# Exercício - Plotagem Simples de Gráfico

Suponha que você tenha os seguintes dados de vendas de uma loja de roupas em diferentes meses do ano:

**vendas** = [5000, 6000, 4500, 7000, 8000, 6500, 7500, 9000, 10000, 8500, 9500, 11000]

Você quer analisar a tendência de vendas ao longo do ano e identificar quais foram os meses de maior e menor vendas. Para fazer isso, você decide plotar um gráfico de linhas com os dados de vendas.

Dica: usem **matplotlib**





# Exercício - Detecção de anagramas

**Escreva um programa em Python que leia duas strings do usuário e determine se elas são anagramas (ou seja, se as letras de uma podem ser rearranjadas para formar a outra). Se as strings forem anagramas, o programa deve exibir uma mensagem informando isso. Caso contrário, o programa deve exibir uma mensagem informando que as strings não são anagramas. Certifique-se de que o programa funcione corretamente mesmo que as strings tenham letras maiúsculas e minúsculas diferentes (por exemplo, "Casa" e "saca" devem ser consideradas anagramas).**







# Exercício - Jogo da adivinhação

**Escreva um programa em Python que simule um jogo de adivinhação em que o usuário tem que adivinhar um número de 1 a 100 em no máximo 7 tentativas. O programa deve gerar um número aleatório entre 1 e 100 e o usuário deve tentar adivinhar qual é o número. Se o usuário acertar o número, o programa deve exibir uma mensagem informando que ele acertou e quantas tentativas foram necessárias. Se o usuário não acertar o número em 7 tentativas, o programa deve exibir uma mensagem informando que ele perdeu e qual era o número secreto. Além disso, o programa deve informar ao usuário se o número que ele digitou é maior ou menor do que o número secreto.**





# Questões do Beecrowd!

**Fazer as questões do BeeCrowd que  
separamos na última aula mandamos  
no whatzap (se der tempo)**





**Obrigado pela atenção!**