

Processamento da Informação

Apresentação da disciplina,
algoritmos sequenciais e módulos

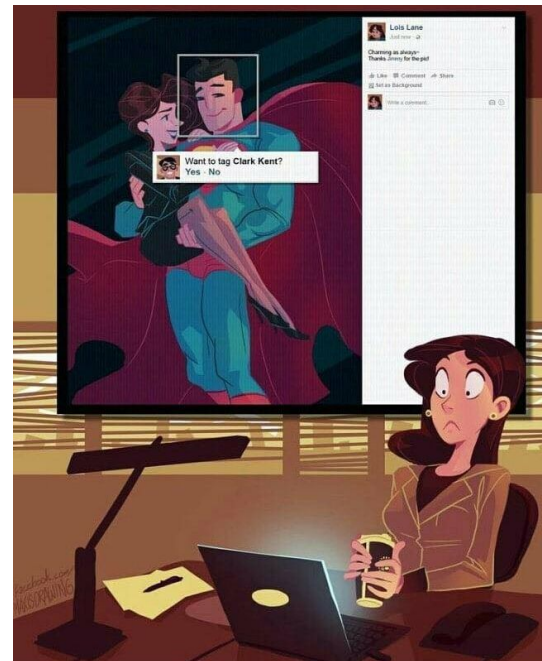
Profa. Debora Medeiros
CMCC-UFABC

Apresentação

- Debora Medeiros
 - debora.medeiros@ufabc.edu.br
- Sala 511, torre 2, bloco A, campus Santo André
- Áreas de pesquisa:
 - Mineração de dados
 - Mineração de textos
 - Aprendizado de máquina
 - Agrupamento semi-supervisionado
 - Otimização baseada em população

Por que aprender a programar?

- Geral
 - Buscadores
 - Classificação de imagens
 - Redes sociais
 - Organização
 - Eficiência em transmissão de dados
 - Planejamento de transporte de cargas
 - Uso do espaço
 - Rotas
 - ChatGPT



Por que aprender a programar?

- Medicina
 - Análise do comportamento de entidades biológicas (proteínas, genes) em diferentes circunstâncias do organismo
 - Desenvolvimento de tratamentos, medicamentos, diagnósticos
 - Apoio ao diagnóstico por sintomas
 - Identificação de tumores em imagens de tomografia

Por que aprender a programar?

- Meio ambiente
 - Detecção de desmatamento via imagens de satélite
 - Predição de adaptação de espécies
- Comércio
 - Gerenciamento de estoque, previsão de compras
 - Análise de padrão dos clientes

Por que aprender a programar?

- Mesmo que vocês não planejem atuar nessa função
- Saber é importante para:
 - Executar
 - Manter
 - Interagir com quem vai executar ou manter
- Programar auxilia a desenvolver o pensamento lógico.
- Programar lhe torna mais independente.

Por que
aprender a
programar?

- Não sou
só eu que
estou
dizendo
xD

Coding Is the Must-Have Job Skill of the Future

7.1k
SHARES

2985

Share

2372

Tweet

403

Share

1079

in

218

StumbleUpon

41

Pin

WHAT'S THIS?

Why Programming Is The Core Skill Of The 21st Century

It's never been easier, more accessible, or more essential to learn coding skills.



How Coding Can Boost Everyone's Career

“Everybody in this country should learn how to program a computer... because it teaches you how to think.”

- Steve Jobs

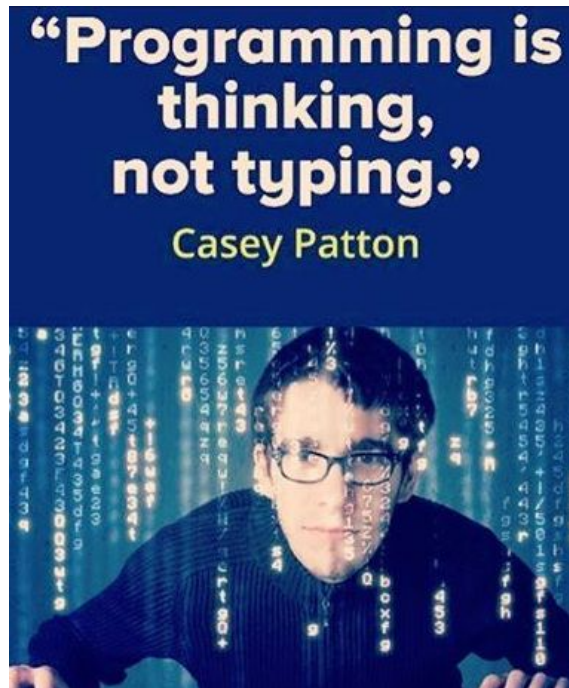
<https://www.youtube.com/watch?v=Dv7gLpW91DM>

Sobre a disciplina

- Objetivos:
 - Apresentar os fundamentos sobre manipulação e tratamento da Informação, principalmente por meio da explicação e experimentação dos conceitos e do uso prático da lógica de programação.
- Competências:
 - Compreender os conceitos fundamentais a respeito da manipulação e tratamento da Informação.
 - Entender a lógica de programação de computadores.
 - Desenvolver algoritmos básicos para modelar e solucionar problemas de natureza técnico-científica.

Sobre a disciplina

- Combinação de teoria e prática de maneira inseparável.
- Análise de um problema
- Planejar a abordagem
- Implementar uma solução



Ferramentas

Comunicação:

- Moodle (moodle.ufabc.edu.br)
 - Material e atividades
- Discord
 - Fórum permanente
 - Plantões de dúvidas
 - Quintas, 17h30-19h30
 - Também na sala 511-2 (SA)
 - Convite:
<https://discord.gg/ad4TnHDSxH>

Programação:

- Spyder
 - Instalação para Linux e Windows
 - Incluso no pacote Anaconda
- Ferramentas online
 - Replit (<https://replit.com/>)
 - Google collab (<https://colab.research.google.com/>)
 - Colaborativas

Apresentação das ferramentas

- Moodle
- Discord
- Spyder
- Google collab
- Replit

Construção de programas

- Conjunto de instruções
 - Ordem sequencial lógica
 - Algoritmo

Algoritmo

- Algoritmo:
 - Sequência lógica são passos executados até atingir um objetivo ou solução de um problema.
- Exemplos de situações onde uma sequência lógica de passos é necessária:
 - Fazer um bolo
 - Construir um robô para explorar um local desconhecido
 - Trocar uma lâmpada

Exemplo de algoritmo

Trocando uma lâmpada:

- 1.

Exemplo de algoritmo

Trocando uma lâmpada:

1. Pegue uma lâmpada nova
2. Pegue uma escada
3. Posicione a escada embaixo da lâmpada
4. Suba a escada
5. Retire a lâmpada queimada
6. Coloque a lâmpada nova

Exemplo de algoritmo

Trocando uma lâmpada:

1. Pegue uma lâmpada nova
2. Pegue uma escada
3. Posicione a escada embaixo da lâmpada
4. Suba a escada
5. Retire a lâmpada queimada
6. Coloque a lâmpada nova

E se não tem uma lâmpada nova?

E se a linguagem de programação não entende este comando, apenas “suba 1 degrau”?

Exemplo de algoritmo

Trocando uma lâmpada:

1. Pegue uma lâmpada nova
2. Pegue uma escada
3. Posicione a escada embaixo da lâmpada
4. Suba a escada
5. Retire a lâmpada queimada
6. Coloque a lâmpada nova

E se não tem uma lâmpada nova?

E se a linguagem de programação não entende este comando, apenas “suba 1 degrau”?

Podemos ficar pensando em se's até amanhã!

Exemplo de algoritmo

Trocando uma lâmpada:

1. Pegue uma lâmpada nova
2. Pegue uma escada
3. Posicione a escada embaixo da lâmpada
4. Suba a escada
5. Retire a lâmpada queimada
6. Coloque a lâmpada nova

E se não tem uma lâmpada nova?

E se a linguagem de programação não entende este comando, apenas “suba 1 degrau”?

É importante ter em mente que o computador não interpreta nada, ele vai seguir literalmente a sequência de instruções!

Exemplo de algoritmo

Trocando uma lâmpada:

1. Pegue uma lâmpada nova
2. Pegue uma escada
3. Posicione a escada embaixo da lâmpada
4. Suba a escada
5. Retire a lâmpada queimada
6. Coloque a lâmpada nova

E se não tem uma lâmpada nova?

E se a linguagem de programação não entende este comando, apenas “suba 1 degrau”?

Dica: assistam o vídeo de um pai ensinando lógica de programação para os filhos (<https://www.youtube.com/watch?v=FN2RM-CHkul&t=10s>)

Exemplo de algoritmo

Trocando uma lâmpada:

1. Pegue uma lâmpada nova
2. Pegue uma escada
3. Posicione a escada embaixo da lâmpada
4. Suba a escada
5. Retire a lâmpada queimada
6. Coloque a lâmpada nova

Instruções de caráter
informal: algoritmo

Traduzindo para
comandos
compreensíveis por uma
máquina: programa

Algoritmos e programas

- Para que um computador desempenhe uma tarefa é necessário que uma sequência de ações (algoritmo) seja especificada de uma **forma compreensível** pela máquina.
- Um **programa de computador** nada mais é que um **algoritmo** escrito de forma **compreensível pelo computador**.
- Ações especificadas de maneira “formal”.

Linguagem de programação

- É um conjunto limitado de:
 - Símbolos (comandos, identificadores, caracteres, etc)
 - Regras de sintaxe (descrevem de forma precisa ações)

No exemplo da troca de lâmpada:

- pegar
- lâmpada nova
- escada

No exemplo da troca de lâmpada:

- o comando **pegar** deve ser seguido do identificador de um objeto

Linguagem de programação

- É um conjunto limitado de:
 - **Símbolos** (comandos, identificadores, caracteres, etc)
 - **Regras de sintaxe** (descrevem de forma precisa ações)
- Exemplo
 - **print** seguido de parênteses

Linguagem de programação

Linguagem de máquina	Compreendida pelo computador. Dependente da arquitetura do computador
Linguagem de baixo nível	Utiliza mnemonicos para a representação de ações elementares Ex. Assembler
Linguagem de alto nível	Utiliza instruções próximas da linguagem humana Ex. C, Java, Python, PHP

Linguagem de programação

Linguagem de máquina	Compreendida pelo computador. Dependente da arquitetura do computador
Linguagem de baixo nível	Utiliza mnemonicos para a representação de ações elementares Ex. Assembler
Linguagem de alto nível	Utiliza instruções próximas da linguagem humana Ex. C, Java, Python, PHP

No exemplo da troca de lâmpada

- Levantar o pé
- Apoiar o pé no degrau

Enquanto houver degrau acima:

- Subir no degrau de cima

- Subir a escada

Linguagem de programação

Linguagem de máquina	Compreendida pelo computador. Dependente da arquitetura do computador
Linguagem de baixo nível	Utiliza mnemonicos para a representação de ações elementares Ex. Assembler
Linguagem de alto nível	Utiliza instruções próximas da linguagem humana Ex. C, Java, Python, PHP

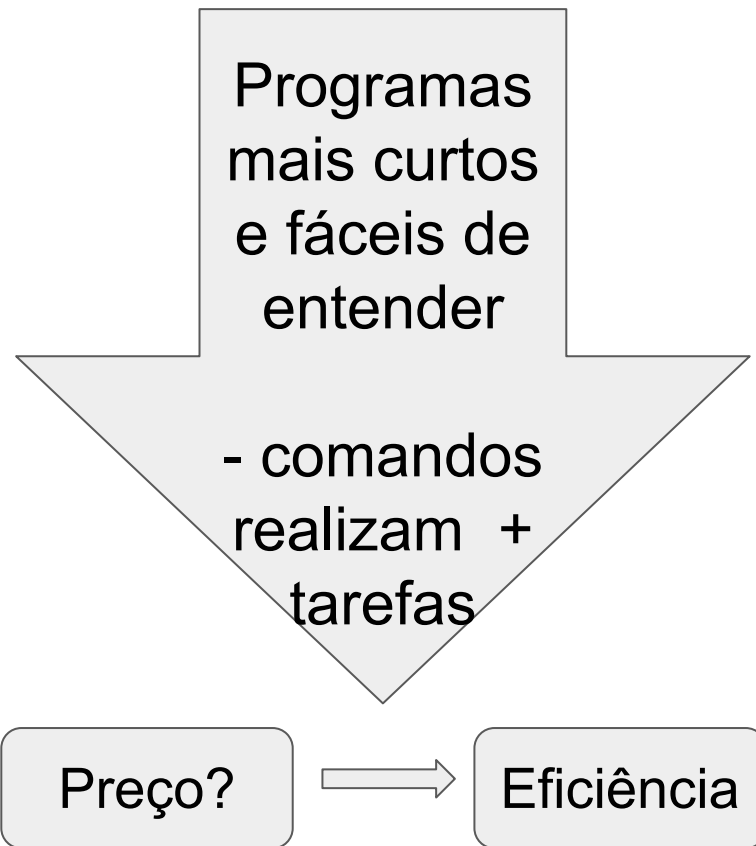
Programas
mais curtos
e fáceis de
entender

- comandos
realizam +
tarefas

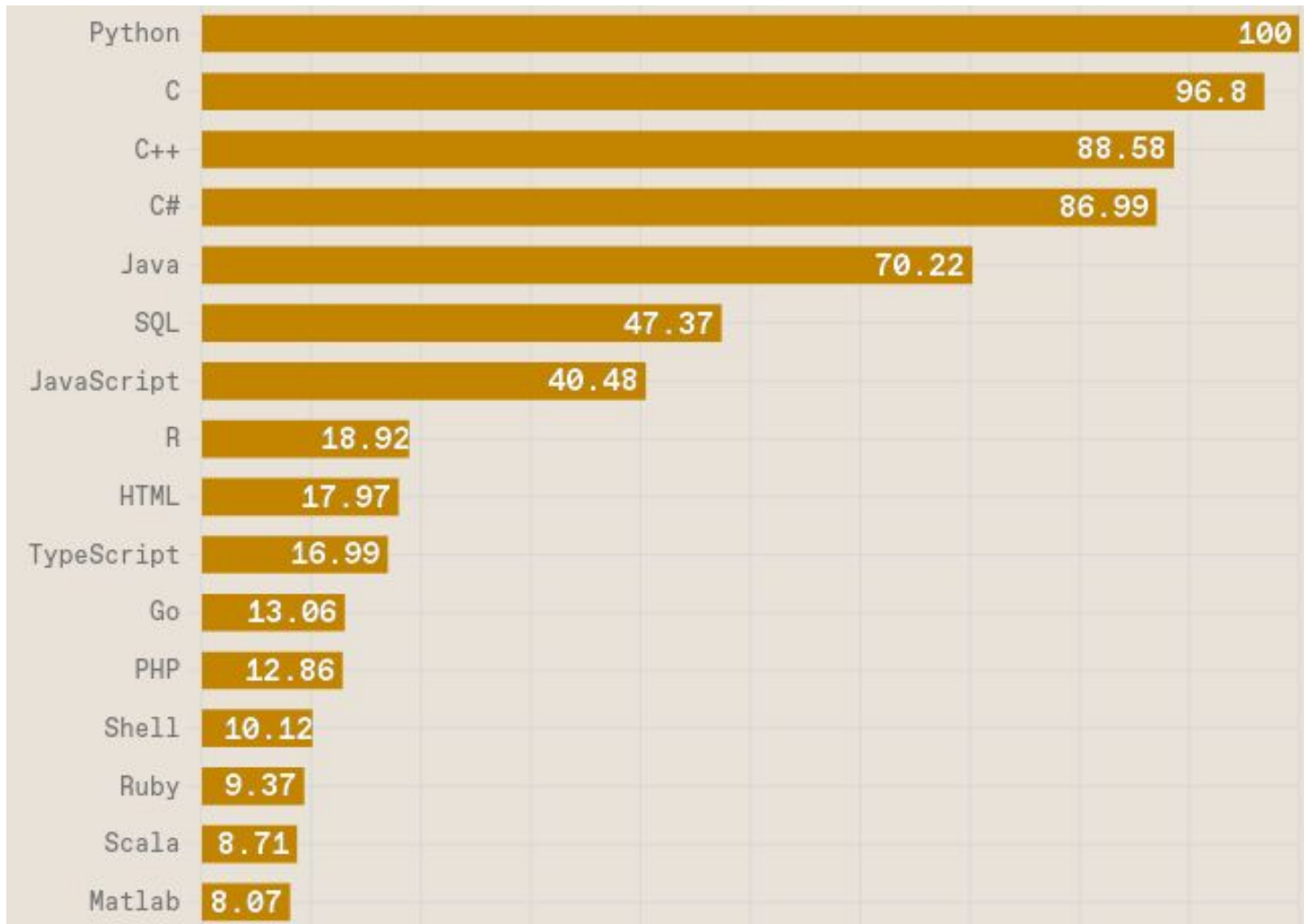
Preço?

Linguagem de programação

Linguagem de máquina	Compreendida pelo computador. Dependente da arquitetura do computador
Linguagem de baixo nível	Utiliza mnemonicos para a representação de ações elementares Ex. Assembler
Linguagem de alto nível	Utiliza instruções próximas da linguagem humana Ex. C, Java, Python, PHP



Linguagens mais utilizadas

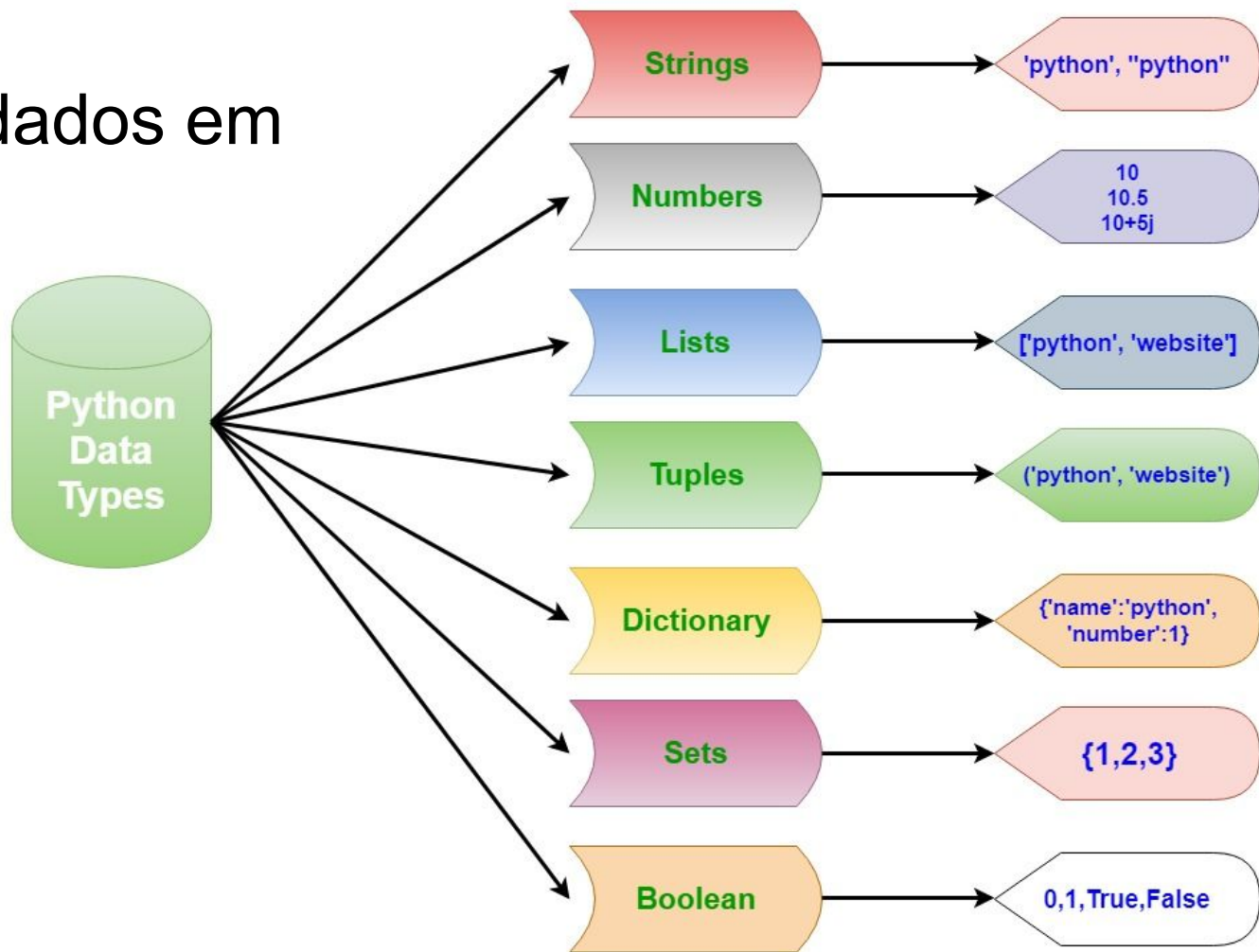


Quem usa python?

- Wikipedia
- Google
- Yahoo!
- CERN
- NASA
- Facebook
- Amazon
- Instagram
- Spotify



Tipos de dados em Python



Tipos de dados em Python

Tipos de dados primitivos:

- Inteiros (int)
 - 3, 8, 3000
- Ponto flutuante (float)
 - 2.5, 3.33, 20.0
- String (str)
 - Sequência de caracteres, ex.: “olá”, “Maria”, “200”
- Listas (list)
 - [“may”, 4, “2021”]
- Booleano (bool)
 - Valor lógico: **True** ou **False**

Precedência de operadores

Qual seria o resultado da execução das seguintes instruções?

- $4 + 5 + 6 / 3$
- $3 ** 2 + 2$
- $-2 ** 4$

Precedência de operadores

Qual seria o resultado da execução das seguintes instruções?

- $4 + 5 + 6 / 3 = 11$
- $3 ** 2 + 2 = 11$
- $-2 ** 4 = -16$

Precedência de operadores

Operador	Operação
* *	Exponenciação
-	Negação
* / // %	Multiplicação Divisão Divisão inteira Módulo
+ -	Soma subtração

Um pouco mais de variáveis e tipos

```
var = 5    # comando de atribuição (atribuindo o valor 5 à variável "var")  
var = 134  # valor inteiro  
var = 4.2  # valor decimal  
var = "hello world" # texto
```

```
print(0.1 + 0.1) #0.2  
print(0.54 * 4) #2.16
```

```
print(0.1 + 0.2) #0.30000000000000004
```

Um pouco mais de variáveis e tipos

```
nome = "Maria"
sobrenome = "Medeiros"
print(nome + sobrenome) #MariaMedeiros
print(nome + " " + sobrenome) #Maria Medeiros
print("Olá " + nome + ", tudo bem?") #Olá Maria, tudo bem?
```

Entrada de dados

```
nome = input("Como é o seu nome? ")  
print("Olá, " + nome)  
ano = input("Em que ano estamos? " )  
proxAno = ano + 1
```

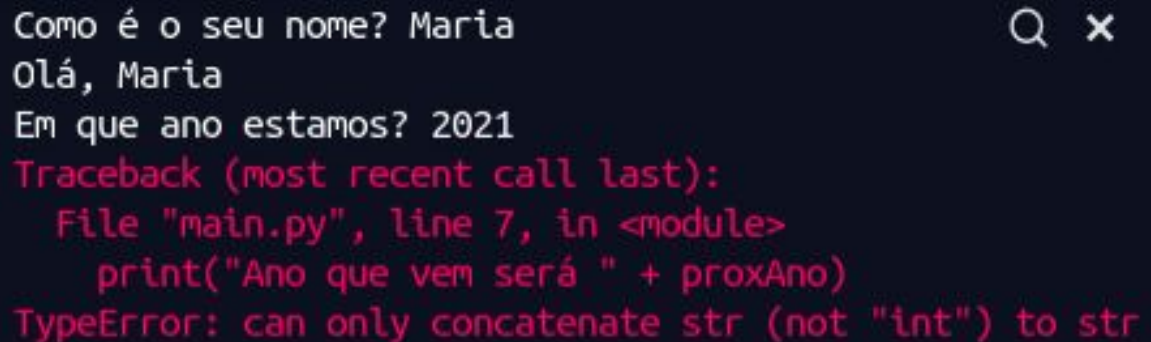
```
Como é o seu nome? Maria  
Olá, Maria  
Em que ano estamos? 2021  
Traceback (most recent call last):  
  File "main.py", line 6, in <module>  
    proxAno = ano + 1  
TypeError: can only concatenate str (not "int") to str
```

Entrada de dados

```
nome = input("Como é o seu nome? ")
print("Olá, " + nome)
ano = int(input("Em que ano estamos? "))
proxAno = ano + 1
print("Ano que vem será " + proxAno)
```

Entrada de dados

```
nome = input("Como é o seu nome? ")  
print("Olá, " + nome)  
ano = int(input("Em que ano estamos? "))  
proxAno = ano + 1  
print("Ano que vem será " + proxAno)
```

A screenshot of a terminal window with a dark background. It shows the execution of a Python script. The first two prompts are answered: 'Como é o seu nome?' with 'Maria' and 'Olá, Maria'. The third prompt 'Em que ano estamos?' is answered with '2021'. Then, a red traceback message appears, indicating a 'TypeError: can only concatenate str (not "int") to str' at line 7 of 'main.py'.

```
Como é o seu nome? Maria  
Olá, Maria  
Em que ano estamos? 2021  
Traceback (most recent call last):  
  File "main.py", line 7, in <module>  
    print("Ano que vem será " + proxAno)  
TypeError: can only concatenate str (not "int") to str
```

Entrada de dados

```
nome = input("Como é o seu nome? ")
print("Olá, " + nome)
ano = int(input("Em que ano estamos? "))
proxAno = ano + 1
print("Ano que vem será " + str(proxAno))
```

```
Como é o seu nome? Maria
Olá, Maria
Em que ano estamos? 2021
Ano que vem será 2022
❏
```


Um programinha com entrada e saída

Conversor de temperatura

```
C = int(input("Digite a temperatura em Celsius: "))  
# a expressão a seguir converte uma temperatura de  
# Celsius para Fahrenheit  
F = C * 9/5 + 32  
print("A temperatura em Fahrenheit é: " + str(F));
```

```
Digite a temperatura em Celsius: 30  
A temperatura em Fahrenheit é: 86.0  
❏
```

Um pouquinho mais de prática

- Receitas
- Caixa eletrônico

Um pouquinho mais de atribuições

- Quais dos comando abaixo vocês acham que daria erro:
 - $6 * \text{-----}8$
 - $8 = \text{alunos}$
 - $(((((4 ** 3)))))$
 - $(-(-(-(-5))))$
 - $4 += 7 / 2$

Um pouquinho mais de atribuições

- Quais dos comando abaixo vocês acham que daria erro:

- $6 * \text{-----}8$

- **8 = alunos**

- $(((((4 ** 3)))))$

- $(-(-(-(-5))))$

- **4 += 7 / 2**

Aula passada

- Quais dos comando abaixo vocês acham que daria erro:

- $6 * \text{-----}8$

- **8 = alunos**

- (((((4 ** 3))))

- (-(-(-(-5))))

- **4 += 7 / 2**

- Problema: números recebendo valores
- += está ok

Operadores de atribuição compostos

- Exemplos

- $X += 1 \rightarrow X = X + 1$

- $X -= 5 \rightarrow X = X - 5$

- $X *= 7 \rightarrow X = X * 7$

- $X /= 87 \rightarrow X = X / 87$

- Também temos os operadores compostos:

- $**=$

- $//=$

- $/=$

- $\%=$

Modularização

- Emprego de módulos/funções/métodos
 - Ferramentas com entrada, saída e um procedimento específico
- Nas aulas anteriores chegamos a usar algumas funções:
 - `min(5, 8, 2) → 2`
 - `round(2.567, 2) → 2.57`

```
In [9]: help(round)
Help on built-in function round in module builtins:

round(number, ndigits=None)
    Round a number to a given precision in decimal digits.

    The return value is an integer if ndigits is omitted or None.  Otherwise
    the return value has the same type as the number.  ndigits may be negative.
```

- Agora vamos aprender a fazer nossas próprias funções

Definição de uma função

Palavra reservada usada para definir uma função

4 espaços indicam que o comando pertence a um bloco (indentação)

A palavra reservada return termina a função e indica qual é o valor de saída

```
def media(x, y):  
    z = (x + y) / 2  
    return z
```

```
print(media(4, 7))
```

Entrada

Procedimento

Saída

Uso

Importante

Cada função cria um ambiente específico para suas variáveis

```
def media(x, y):  
    z = (x + y) / 2  
    return z
```

```
print(media(4, 7)) #5.5  
print(z)
```

O que será
impresso aqui?

Importante

Cada função cria um ambiente específico para suas variáveis

```
def media(x, y):  
    z = (x + y) / 2  
    return z  
  
print(media(4, 7)) #5.5  
print(z)
```

Traceback (most recent call last):

```
File "<ipython-input-67-893fa9747e51>", line 7, in <module>  
    print(z)
```

NameError: name 'z' is not defined

Algumas funções matemáticas

```
n = -4
```

```
print(pow(n,2)) #16
```

```
print(abs(n)) #4
```

```
import math
```

```
print(math.sin(n)) #0.7568024953079282
```

```
print(math.sqrt(n))
```

```
Traceback (most recent call last):
```

```
File "<ipython-input-29-4a4581bbbc88>", line 7, in <module>  
    print(math.sqrt(n))
```

```
ValueError: math domain error
```

```
print(math.sqrt(abs(n))) #2.0
```

Vantagens

- Encapsular trechos de código que se repetem
 - Agilidade
- Organização
 - Facilitar teste do código
 - Menos incidência de erros
- Tornar o código mais legível

Outro exemplo

- Bhaskara

$$\Delta = \sqrt{b^2 - 4ac}$$

$$x' = \frac{-b - \sqrt{\Delta}}{2a}$$

$$x'' = \frac{-b + \sqrt{\Delta}}{2a}$$

Outro exemplo

- Bhaskara

$$\Delta = \sqrt{b^2 - 4ac}$$

$$x' = \frac{-b - \sqrt{\Delta}}{2a}$$

$$x'' = \frac{-b + \sqrt{\Delta}}{2a}$$

```
def bhaskara(a, b, c):  
    delta = b**2 - 4*a*c  
    xl = (-b - delta**0.5)/(2*a)  
    xll = (-b + delta**0.5)/(2*a)  
    #ops, e agora? retorno xl ou xll?
```

Tuplas

- Conjunto pré-determinado de elementos

```
x = 3.987
print(type(x)) #<class 'float'>
y = (x, x)
print(type(y)) #<class 'tuple'>
z = (1.2, 0)
print(type(z)) #<class 'tuple'>
print(z[0]) #1.2
w = 1, "um"
print(type(w)) #<class 'tuple'>
a, b = z
print(a) #1.2
print(b) #0
```

Continuando Bhaskara

```
def bhaskara(a, b, c):  
    delta = b**2 - 4*a*c  
    x1 = (-b - delta**0.5)/(2*a)  
    x2 = (-b + delta**0.5)/(2*a)  
    return x1, x2
```

```
x1, x2 = bhaskara(1, 0, -1)  
print(x1) #-1.0  
print(x2) #1.0
```


Exemplo do caixa eletrônico

```
valor = int(input("Informe o valor (múltiplos de 5 reais): "))

notas100 = valor // 100
valor = valor - (notas100 * 100)

notas50 = valor // 50
valor = valor - notas50 * 50

notas20 = valor // 20
valor = valor % 20 #equivalente a valor = valor - notas20 * 20

notas10 = valor // 10
valor = valor % 10

notas05 = valor // 5

print("Você receberá " + str(notas100) + " notas de 100, "
      + str(notas50) + " notas de 50, " + str(notas20)
      + " notas de 20, " + str(notas10) + " notas de 10 e "
      + str(notas05) + " notas de 5")
```

Exemplo do caixa eletrônico

```
def xCedulas (cedula, quantia):  
    x = quantia // cedula  
    restante = quantia % cedula  
    return x, restante
```

```
valor = int(input("Informe o valor (múltiplos de 5 reais): "))
```

```
notas100, valor = xCedulas(100, valor)  
notas50, valor = xCedulas(50, valor)  
notas20, valor = xCedulas(20, valor)  
notas10, valor = xCedulas(10, valor)  
notas05, valor = xCedulas(5, valor)
```

```
print("Você receberá " + str(notas100) + " notas de 100, "  
      + str(notas50) + " notas de 50, " + str(notas20)  
      + " notas de 20, " + str(notas10) + " notas de 10 e "  
      + str(notas05) + " notas de 5")
```

Referências

- Material do prof. Jesús P Mena-Chalco (UFABC)
- Material do prof. Thiago Covões (UFABC)