

MODELO DE PLANEJAMENTO/PLANO DE AULA PARA CADA ENCONTRO

IDENTIFICAÇÃO		
Instituição Concedente: Colégio Impacto Centro de Ensino LTDA – ME Professor: Bruno Camargo Braga Componente Curricular: Física Unidade temática: Investigação Científica e Processos Criativos. Tema de estudo: Primeira interação com a linguagem de programação Python		
Grupos 3 e 4: 2º ano do Ensino Médio	Tempo: 2 horas	Data: 25/04/2025
COMPETÊNCIAS E HABILIDADES		
(EMIFCG03) Utilizar informações, conhecimentos e ideias resultantes de investigações científicas para criar ou propor soluções para problemas diversos; (EMIFCG05) Questionar, modificar e adaptar ideias existentes e criar propostas obras ou soluções criativas, originais ou inovadoras, avaliando e assumindo riscos para lidar com as incertezas e colocá-las em prática.		
OBJETIVOS DE APRENDIZAGEM		
Utilizando a linguagem de programação Python (versão 3.11), ser capaz de realizar: <ol style="list-style-type: none"> 1. Operações numéricas; 2. Manipulação de textos e uso de comentários; 3. Operações com listas (tanto de números quanto de textos). 		
RECURSOS/MATERIAIS DIDÁTICOS		
Quadro, smartphone, Google Classroom, Google Colab, Pydroid 3.		
PROCEDIMENTOS METODOLÓGICOS		
<p>1º momento (5 min): Saudações iniciais, chamada e anotação da ata de sala</p> <p>2º momento (25 min): O convite dos alunos na turma do Google Classroom deveria ser feito previamente, porém possivelmente muitos alunos não responderam ao convite. Tome esse começo da aula para terminar de colocar todos os alunos presentes na turma, e ajude-os a baixar os aplicativos necessários – Google Classroom e Pydroid 3 (para quem usar Android). Indique aos alunos com iphone como usar o <i>notebook</i> do Google Colab.</p>		

No começo da aula será discutido brevemente o formato adotado das aulas, como os alunos serão avaliados e os temas que serão abordados nas aulas. Após essa introdução é explicado como que os alunos terão acesso ao Python 3, seguindo o modelo sobre o uso direto e o uso local.

Uso direto sem instalação (aplicação pela nuvem):

Tanto no computador quanto no celular podemos utilizar o Python sem instalar qualquer aplicativo, conseguimos isso devido a sites, que, dentre os mais confiáveis, temos o **Google Colab**.

O problema de opções como o Google Colab é a dependência de conexão com a internet, além de possíveis falhas devido ao congestionamento da conexão ou super lotação do servidor. Para não lidar com esse possível problema, precisamos entender como instalar localmente o Python e todas as bibliotecas não padrões necessárias às aulas.

Uso local (instalando aplicativo e bibliotecas):

- **No celular, caso seja Android**, existem aplicativos personalizados disponíveis no Google Play, como por exemplo o **Pydroid 3**. O aplicativo já vem com a opção "Pip", que ajuda a instalar as bibliotecas de terceiros, como é o caso do *matplotlib* (uso para construir gráficos).
- **Em um computador Windows**, é possível executar o Python 3 direto no terminal PowerShell ou com auxílio de programas como o **Visual Studio Code** (recomendado 8GB+ de RAM). No Visual Studio Code vá em Extensões e instale Python e o Pip Manager.
- **Em um computador Linux**, é fácil de usar o Python 3, pois nas versões mais recentes do Ubuntu (e similares tipo o Linux Mint) o compilador de Python já vem pré-instalado no sistema operacional, e podemos checar sua versão com o comando:

```
$ python --version
```

Caso não tenha o Python 3 no seu Linux, é simples de instalar com os comandos seguintes:

```
$ sudo apt update
$ sudo apt install build-essential
$ sudo apt install python3.11
```

Por fim, para instalar o pip (instalador de bibliotecas):

```
$ sudo apt-get install python3-pip
```

3º momento (45 min): De início a avaliação pedagógica. Explique aos alunos como usar o Google Forms, a partir do link disponibilizado no Classroom.

4º momento (15 min): De início a uma breve introdução de como usar o Python, explique o que são comentários e operações básicas.

Deve-se discutir com os alunos alguns como o Python aceita os comandos, seu formato e algumas funções básicas. No Python e muitas outras linguagens similares, os comandos são lidos por linha, de cima para baixo. Podemos escrever

comentários no código sem atrapalhar os comandos, para isso utilizamos o símbolo #. Tudo que é escrito após o # é ignorado. Colocamos apenas para pessoas lerem, explicando o objetivo do comando ou deixando algum recado tipo (esse código não faz tal coisa, ex: conta uvas, mas não consegue contar melões);

Para imprimir qualquer resultado obtido pela execução do código, na tela do seu computador ou celular, usamos o comando **print()**. Nesta aula será disponibilizado o código do professor aos alunos (em anexo).

Deve-se destacar aos alunos que podemos de fato escrever comentários em português no código, mas tudo fora dos comentários deve seguir o molde da língua Inglesa. Assim, não podemos usar acentuações, o "ç" e quanto aos números devemos usar ponto ao invés da vírgula para as casas decimais (Errado: 8,5 | Correto: 8.5).

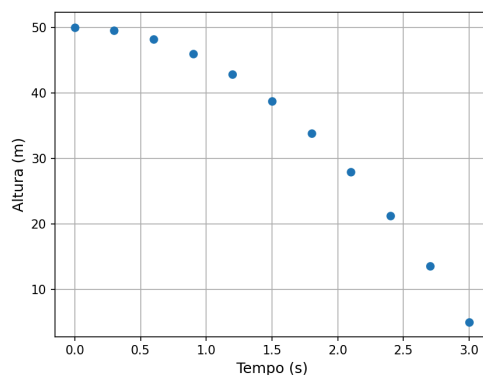
Será apresentado aos alunos como tratar três tipos de variáveis: números, textos e listas. Como em qualquer calculadora, o Python aceita algumas operações numéricas básicas (soma, subtração, multiplicação, divisão e potência). Após falar das operações numéricas é recomendado tratar textos e frases e como podemos manipula-los no Python. Por fim, será discutido o agrupamento de dados numéricos com textos por meio das listas. Com o tratamento correto desses dados, podemos resolver qualquer problema de Física e matemática. A lógica de programação será vista nas aulas seguintes.

5º momento (30 min): Com alguns comandos básicos explicados e descritos no quadro, deixe os alunos testarem em seus aparelhos. Circule a sala tirando dúvidas.

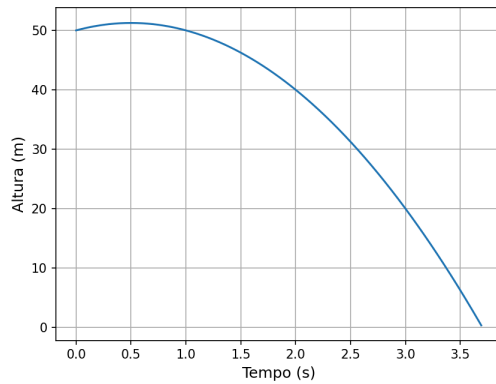
AVALIAÇÃO

Avaliação diagnóstica. Recomenda-se fazer a avaliação pelo Google Forms, mas pode imprimir-la e entregar aos alunos. As questões são descritas a seguir:

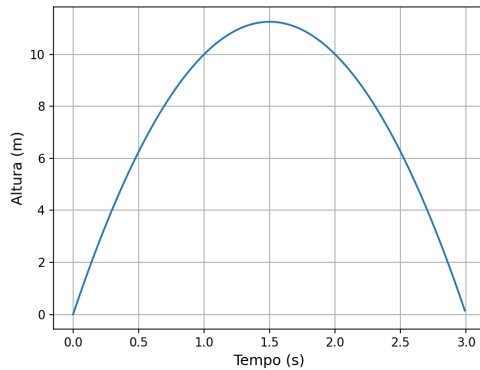
- Observando o gráfico, podemos dizer que:



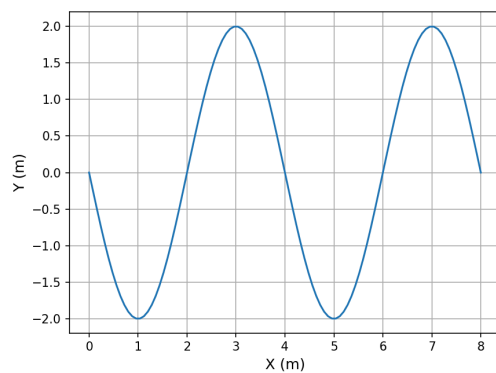
- A posição inicial do corpo é o chão
 - O corpo possui aceleração diferente de zero (**correto**)
 - Após 1,5 segundos o corpo deslocou 40 m da posição inicial
- Observando o gráfico, podemos dizer que:



- A velocidade inicial tem a mesma direção e sentido que a gravidade
 - A velocidade inicial é zero (partiu do repouso)
 - A velocidade inicial tinha sentido oposto à gravidade (**correto**)
3. Observando o gráfico do lançamento de um corpo, podemos dizer que:



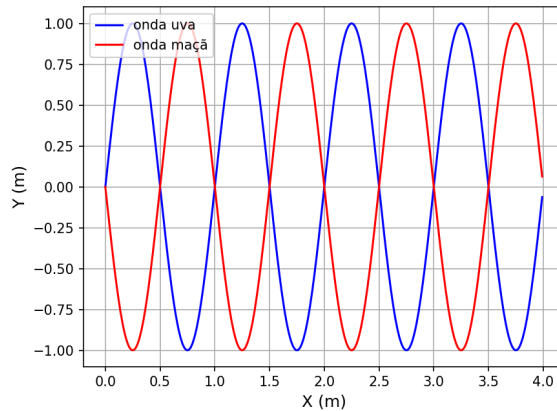
- Após 3 segundos o corpo se deslocou 3 m para a direita
 - No tempo 1,5 s o corpo parou no ar (**correto**)
 - O deslocamento do corpo foi de 22 m
4. Qual a amplitude da onda a seguir?



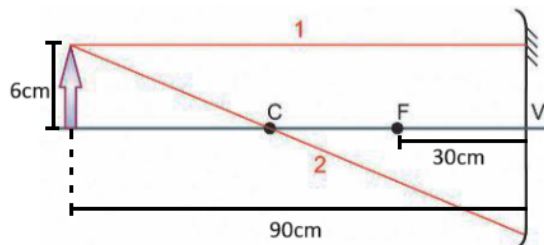
- 4 m
 - 2 m (**correto**)
 - 1 m
5. Qual o comprimento de onda da onda apresentada no exercício anterior?
- 5 m

- b. 2 m
- c. 4 m **(correto)**

6. Considerando que as ondas uva e maçã possuem a mesma frequência e amplitude, o que podemos dizer sobre sua soma?



- a. A amplitude permanece a mesma e a frequência dobra
 - b. A amplitude zera **(correto)**
 - c. A amplitude dobra
7. Resolva a equação: $x + 2y = 25$
- a. $x = 25$, $y = 2$
 - b. se $y=1$, então $x= 20$
 - c. se $y=1$, então $x= 23$ **(correto)**
8. Resolva a equação: $x^2 - 4 = 0$
- a. $x= 4$, $x=2$
 - b. $x=-2$, $x=2$ **(correto)**
 - c. $x=-8$, $x=8$
9. Resolva o sistema de equações:
- $$x^2 - 2y = 0$$
- $$6x + y = -18$$
- a. $x=-6$, $y=18$ **(correto)**
 - b. $x=6$, $y=-18$
 - c. $x=-18$, $y=6$
10. Considere um objeto luminoso colocado sobre o eixo principal de um espelho esférico côncavo. Sabendo que a distância focal do espelho é 30 cm, que a distância do objeto ao espelho é 90 cm e que a altura do objeto é 6,0 cm, calcule a distância da imagem ao espelho e a altura da imagem, ambas em centímetros.



- a. Distância 45 cm; Altura -3 cm **(correto)**
- b. Distância -45 cm; Altura 9 cm
- c. Distância 6 cm; Altura -2 cm

REFERÊNCIAS
G. Van Rossum, F. L. Drake. <i>Python 3 Reference Manual</i> . Scotts Valley, CA: CreateSpace; 2009.
ANEXOS
<p>Será disponibilizado ao alunos via Google Classroom o código a seguir:</p> <p>(Para reduzir o tamanho deste arquivo, coloquei todos os códigos feitos no github e deixo disponível aqui o link de acesso)</p> <p>https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/Aula_1.py</p>

MODELO DE PLANEJAMENTO/PLANO DE AULA PARA CADA ENCONTRO

IDENTIFICAÇÃO		
Instituição Concedente: Colégio Impacto Centro de Ensino LTDA – ME Professor: Bruno Camargo Braga Componente Curricular: Física Unidade temática: Investigação Científica e Processos Criativos. Tema de estudo: Fundamentos de programação Python		
Grupos 3 e 4: 2º ano do Ensino Médio	Tempo: 2 horas	Data: 05/05/2025
COMPETÊNCIAS E HABILIDADES		
(EMIFCG03) Utilizar informações, conhecimentos e ideias resultantes de investigações científicas para criar ou propor soluções para problemas diversos; (EMIFCG05) Questionar, modificar e adaptar ideias existentes e criar propostas obras ou soluções criativas, originais ou inovadoras, avaliando e assumindo riscos para lidar com as incertezas e colocá-las em prática.		
OBJETIVOS DE APRENDIZAGEM		
Utilizando a linguagem de programação Python (versão 3.11), ser capaz de: <ol style="list-style-type: none"> 1. Entender e utilizar estruturas de repetição (código em ciclos); 2. Interagir com o programa durante sua execução (uso do comando <i>input</i>); 3. Entender e utilizar a estrutura condicional <i>if-else</i>; 		
RECURSOS/MATERIAIS DIDÁTICOS		
Quadro, smartphone, Google Classroom, Google Colab, Pydroid 3.		
PROCEDIMENTOS METODOLÓGICOS		
<p>1º momento (10 min): Saudações iniciais, chamada e anotação da ata de sala</p> <p>2º momento (20 min): Revise os comandos vistos no encontro anterior.</p> <p>3º momento (10 min): Apresente o problema da repetição de comandos e a fácil solução, por meio dos blocos cíclicos.</p> <p>Uma das vantagens das linguagens de alto nível, como o Python, está nas funcionalidades cujo objetivo é evitar repetição de comandos, o que deixa o código mais limpo e fácil de entender. A funcionalidade mais comum entre as linguagens</p>		

de programação é a estrutura de repetição (ou muitas vezes chamadas por ciclos/*loops*). Os comandos que iniciam ciclos no Python são: *while* e *for*.

No começo da aula devemos mostrar aos alunos o modo mais trabalhoso de fazer, a exemplo preencher uma lista com dados. Para tal repetimos o comando de adicionar a lista toda vez que precisamos. Porém, com o uso correto de uma estrutura de repetição, obtemos a automação desse processo.

4º momento (20 min): Detalhe os comandos cíclicos “while” e “for”.

A seguir é recomendado que seja explicado tal estrutura, seguindo os comandos:

- Comando **while expressão**: Do inglês "enquanto" a expressão for verdade, repita o código dentro da estrutura. Este comando para ser implementado corretamente precisa de um passo que modifica a expressão toda vez que completa-se um ciclo. Isso é feito para que eventualmente a expressão se torne falsa, e assim o ciclo pare. Ex: Com $i=0$, se a expressão for "*while i<5*", dentro da estrutura deve ser escrito algum passo que aumente 'i', de modo que ao 'i' se tornar 5 ou maior, a estrutura "*while*" termina.
- Comando **for x in lista**: Diferentemente do "*while*", o "*for*" deixa a quantidade de repetições bem definida na expressão escrita após o "*for*" – "*x in lista*". Do inglês significa: repita a estrutura "para" x assumindo os valores da lista um a um. Ex: "*for x in [1,25,30]*": repita a estrutura assumindo que o valor de x será 1 na primeira execução, 25 na segunda e 30 na última.

O comando "*for*" é mais utilizado quando a lista de valores de uma variável é bem conhecida, enquanto que o "*while*" é mais genérico e muitas vezes usado para ciclos com muitas repetições e condições muito raras de parada.

5º momento (30 min): Apresente a possibilidade de interação com o programa, durante sua execução. Termine o assunto falando da necessidade de verificar as respostas antes de usa-las, para isso apresente a estrutura condicional “if-else”.

Para evitar que façamos tudo direto no código em si, existem funções de interação durante a execução do código. A mais comum é o pedido de uma resposta a perguntas pré-definidas. Fazemos isso no Python com o comando **input()**. Ao executar o código ele irá pausar pedindo que seja repassado uma resposta. Ex: dentro de um ciclo pode ter uma linha "*x=input('Forneça um número')*"; significa que toda vez que o ciclo reiniciar o código pausa, imprime a mensagem ('Forneça um número') na tela e espera a resposta. Assim podemos modificar o significado de uma variável durante a execução, podendo por exemplo executar 2 ou 20 repetições de um ciclo usando o mesmo código.

Com comandos tipo o **input()**, teremos uma ou mais variáveis no código que não sabemos o valor e seu tipo (se é um número ou texto). Na mensagem se *input* podemos pedir um tipo de resposta específico, mas isso não garante que o código receberá o que se espera. Para garantir que estamos trabalhando corretamente, precisamos verificar as respostas dadas. Para tal usamos a estrutura condicional "*if-else*". Do inglês "*se-senão*", são blocos que permitem execuções diferentes baseado na verificação feita.

Ex: se x for par faça X1, senão (ele será ímpar) então faça X2; essa verificação é

escrita como:

"if x%2==0:

X1

Else:

X2"

6º momento (30 min): Apresente as atividades, deixe os alunos fazendo e circule pela sala tirando dúvidas.

Crterios de avaliao:

1. *Atingiu o objetivo da atividade: solucionou o problema usando o Python 3 como principal ferramenta – independente das estruturas e mtodos usados, desde que o programa escrito seja funcional (7,0pt);*
2. *Estrutura e mtodos:*
 - a. *O programa est organizado e fcil de entender (1,0pt);*
 - b. *Foi utilizado os mtodos explanados em cada encontro (1,0pt);*
 - c. *O programa est bem comentado, explicando o raciocnio lgico (1,0pt).*

AVALIAO
<p>Atividades recomendadas:</p> <ol style="list-style-type: none"> 1. <i>Faça um código que peça uma seqüência de valores e calcule a média deles.</i> 2. <i>Faça uma código que peça um número inteiro e nos diz se o número é par ou ímpar.</i>
REFERÊNCIAS
<p>G. Van Rossum, F. L. Drake. <i>Python 3 Reference Manual</i>. Scotts Valley, CA: CreateSpace; 2009.</p>
ANEXOS
<p>Será disponibilizado ao alunos via Google Classroom o código a seguir:</p> <p>https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/Aula_2.py</p>

MODELO DE PLANEJAMENTO/PLANO DE AULA PARA CADA ENCONTRO

IDENTIFICAÇÃO		
Instituição Concedente: Colégio Impacto Centro de Ensino LTDA – ME Professor: Bruno Camargo Braga Componente Curricular: Física Unidade temática: Investigação Científica e Processos Criativos. Tema de estudo: Fundamentos de programação Python – Parte 2		
Grupos 3 e 4: 2º ano do Ensino Médio	Tempo: 2 horas	Data: 09/05/2025
COMPETÊNCIAS E HABILIDADES		
(EMIFCG03) Utilizar informações, conhecimentos e ideias resultantes de investigações científicas para criar ou propor soluções para problemas diversos; (EMIFCG05) Questionar, modificar e adaptar ideias existentes e criar propostas obras ou soluções criativas, originais ou inovadoras, avaliando e assumindo riscos para lidar com as incertezas e colocá-las em prática.		
OBJETIVOS DE APRENDIZAGEM		
Utilizando a linguagem de programação Python (versão 3.11), ser capaz de: <ol style="list-style-type: none"> 1. Entender e utilizar estruturas de repetição indefinidas (ciclos infinitos); 2. Definir seus próprios métodos dentro da linguagem (comando <i>def</i>); 3. Trabalhar com funções matemáticas (comando <i>lambda</i>); 		
RECURSOS/MATERIAIS DIDÁTICOS		
Quadro, smartphone, Google Classroom, Google Colab, Pydroid 3.		
PROCEDIMENTOS METODOLÓGICOS		
<p>1º momento (10 min): Saudações iniciais, chamada e anotação da ata de sala</p> <p>2º momento (20 min): Revise os comandos vistos no encontro anterior.</p> <p>3º momento (20 min): Apresente os ciclos “infinitos” e suas vantagens</p> <p>Muitas vezes é impossível de saber quando uma estrutura cíclica deve acabar. Nesses casos usamos o comando "while True", que significa: enquanto a verdade seja verdadeira repita a estrutura. Isso coloca o programa em um ciclo sem fim.</p>		

Para que a execução termine é necessário usar a estrutura condicional "if-else" dentro da estrutura do "while", quando obtermos o resultado esperado precisamos uma maneira de cessar a execução. No Python podemos usar o comando **break** dentro do "if-else", que literalmente quebra a execução do ciclo.

Comece a aula com um exemplo de ciclo indefinido. Um caso simples seria realizar alguma operação qualquer com números que ao terminar imprime o resultado e pede um novo número para reiniciar o processo. Esse ciclo usaria um tipo numérico específico para cessar o ciclo, por exemplo um número primo.

4º momento (20 min): Apresente a definição de métodos e sua vantagem em evitar repetições.

É extremamente comum que, durante a implementação da solução de um problema, exista uma sequência de comandos que é repetida diversas vezes no código. Para evitar o trabalho extra de sempre reescrever a mesma sequência de linhas, podemos definir métodos no Python usando o comando "def" e simplesmente chama-los toda vez que for necessário. Isso não apenas diminui o trabalho de escrita, mas também deixa a leitura do código mais fácil. Isso é feito seguindo o modelo:

```
"def metodo(variavel):
```

```
    ...sequência de comandos"
```

Quando necessário é então chamado no código seguindo o exemplo:

```
"...comandos..
```

```
metodo(x)
```

```
...comandos..
```

```
metodo(y)
```

```
...comandos.."
```

Um exemplo de como apresentar a definição de métodos seria pedir para os alunos calcular o fatorial de vários números, ou várias posições da sequência de Fibonacci. Assim fica evidente a vantagem do comando "def".

5º momento (20 min): No fim da aula apresente a ideia de utilizar funções matemáticas. Isso ajuda muito na construção de gráficos e resoluções de problemas comumente presentes em vestibulares.

Uma das formas de calcular funções matemáticas no Python é usar o comando "lambda". Diferentemente dos métodos vistos até o momento, "lambda" não retorna um valor específico, mas sim uma função/regra.

Ex: o comando "f = lambda x: x**2", faz f receber a função quadrática. Ao chamar os valores f(2), f(3) e f(4) será recebido os valores 4, 9 e 16.

Com o a união dos comandos "def" e "lambda" conseguimos trabalhar com funções mais complexas, como a equação horária da posição no movimento uniformemente variado.

6º momento (30 min): Apresente as atividades, deixe os alunos fazendo e circule pela sala tirando dúvidas.

Crêterios de avaliaç o:

1. Atingiu o objetivo da atividade: solucionou o problema usando o Python 3 como principal ferramenta – independente das estruturas e m todos usados, desde que o programa escrito seja funcional **(7,0pt)**;
2. Estrutura e m todos:
 - a. O programa est  organizado e f cil de entender **(1,0pt)**;
 - b. Foi utilizado os m todos explanados em cada encontro **(1,0pt)**;
 - c. O programa   bem comentado, explicando o racioc nio l gico **(1,0pt)**.

Nesta aula focaremos nos conte dos dos livros did ticos da escola:

1. Velocidade de propaga  o de um pulso transversal em meios unidimensionais e homog neos (ondas em cordas). De modo resumido veremos as equa  es de velocidade:
 - a. $v = \sqrt{\frac{T}{\mu}}$, onde T   a intensidade da for a tra  o na corda e μ   a massa espec fica da corda;
 - b. $v = \lambda \cdot f$, onde λ   o comprimento de onda e f a frequ ncia.
2.  ptica: Espelhos planos.

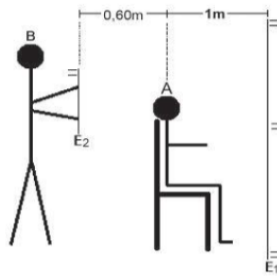
AVALIA  O

Atividades recomendadas:

1. Quando jogamos uma pedra em um lago de  guas calmas, s o produzidas ondas per dicas que percorrem 5 m em 10 s. Sendo a dist ncia entre duas cristas sucessivas igual a 40 cm, teremos que a frequ ncia e a velocidade de propaga  o dessas ondas s o, respectivamente, iguais a?

Entregue um c digo que fa a a solu  o do problema e imprime a resposta.

2. Considerando que ao usar espelhos planos, a dist ncia da imagem ao espelho   igual a dist ncia do espelho ao objeto.
Um barbeiro B segura um espelho plano E2, de espessura desprez vel, paralelamente a outro espelho plano E1, t b m de espessura desprez vel, permitindo que seu cliente A observe, no espelho E1, o seu corte de cabelo na parte posterior da cabe a.



Determine a menor dist ncia entre o cliente e a imagem que ele observa da sua nuca no espelho E1, considerando que a cabe a do cliente t b m possui dimens es desprez veis.

Entregue um c digo que fa a a solu  o do problema e imprime a resposta.

Dica: use o método "def".

REFERÊNCIAS

G. Van Rossum, F. L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace; 2009.

ANEXOS

Será disponibilizado ao alunos via Google Classroom o código a seguir:

https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/Aula_3_base.py

https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/2Ano/Aula_3.py

MODELO DE PLANEJAMENTO/PLANO DE AULA PARA CADA ENCONTRO

IDENTIFICAÇÃO		
Instituição Concedente: Colégio Impacto Centro de Ensino LTDA – ME Professor: Bruno Camargo Braga Componente Curricular: Física Unidade temática: Investigação Científica e Processos Criativos. Tema de estudo: Construindo gráficos no Python		
Grupos 3 e 4: 2º ano do Ensino Médio	Tempo: 2 horas	Data: 16/05/2025
COMPETÊNCIAS E HABILIDADES		
(EMIFCG03) Utilizar informações, conhecimentos e ideias resultantes de investigações científicas para criar ou propor soluções para problemas diversos; (EMIFCG05) Questionar, modificar e adaptar ideias existentes e criar propostas obras ou soluções criativas, originais ou inovadoras, avaliando e assumindo riscos para lidar com as incertezas e colocá-las em prática.		
OBJETIVOS DE APRENDIZAGEM		
Utilizando a linguagem de programação Python (versão 3.11), ser capaz de: <ol style="list-style-type: none"> 1. Utilizar bibliotecas não padrões do Python; 2. Construir gráficos a partir de uma lista de pontos; 		
RECURSOS/MATERIAIS DIDÁTICOS		
Quadro, smartphone, Google Classroom, Google Colab, Pydroid 3.		
PROCEDIMENTOS METODOLÓGICOS		
<p>1º momento (10 min): Saudações iniciais, chamada e anotação da ata de sala</p> <p>2º momento (20 min): Revise os comandos vistos no encontro anterior.</p> <p>3º momento (20 min): Apresente e ajude os alunos a usar a biblioteca necessária para geração de gráficos.</p> <p>Para gerar gráficos precisamos de uma biblioteca específica. Caso o aluno esteja usando aplicativos como o <i>Google Colab Notebook</i>, todas as bibliotecas já estão instaladas e ele não precisa de nenhum trabalho extra. Se o aluno estiver programando localmente (sem internet), ele precisará instalar o matplotlib e o numpy. O <i>numpy</i> é necessário para algumas funções do <i>matplotlib</i>.</p>		

Recomenda-se instalar o **Pip**, pois esse programa faz o trabalho de instalar corretamente novas bibliotecas no seu interpretador local do Python. O guia de instalação do Pip está no material "Como utilizar o Python 3" da Aula 1. Com o Pip instale as bibliotecas: **matplotlib**; **numpy**.

4º momento (20 min): Comece a aula guiando os alunos a como usar o “matplotlib” para construir gráficos.

A forma mais simples de construção de gráfico pelo matplotlib é utilizando uma lista de pontos a serem imprimidos no gráfico. Primeiramente precisamos que a biblioteca monte a interface do gráfico. Então chamamos a biblioteca com:

```
"import numpy as np
```

```
import matplotlib.pyplot as plt"
```

Assim deixamos mais fácil chama-la com a abreviação plt, e usamos então o comando:

```
"fig = plt.figure(dpi=150)", que cria a interface com resolução de 150 linhas.
```

Com a interface pronta, podemos definir a lista de pontos (com os métodos vistos nas aulas anteriores) e nomear os eixos do gráfico usando:

```
"plt.ylabel("nome do eixo y", fontsize=12)
```

```
plt.xlabel("nome do eixo x", fontsize=12)"
```

Se necessário podemos colocar uma grade no gráfico, para facilitar a visualização dos valores de cada ponto. Para isso usamos o comando "plt.grid(True)".

Para adicionar os pontos no gráfico, precisamos colocar separadamente os valores de x e de y de cada ponto. Ex: Para adicionar os pontos (1,2), (-5,3) e (2,0); use o comando:

```
"plt.plot([1,-5,2], [2,3,0], 'o')"
```

O símbolo 'o' no final indica que é para o gráfico apresentar os pontos, se não utilizarmos ele o gráfico é contruído usando uma linha. Essa linha simplesmente liga os pontos e os omite, então com poucos pontos fica uma representação incorreta. O gráfico apresentaria corretamente uma curva se tivesse muitos pontos (mais de 100).

Para o interpretador do Python entender que deve-se mostrar o gráfico na tela, usamos o comando: "plt.show()".

5º momento (20 min): De um exemplo de gráfico na aula. O exemplo de gráfico apresentado na sala de aula deve ser pertinente a um conteúdo visto no ano escolar do grupo ou de um conteúdo recorrente e que os alunos possuem dificuldade. Nesse grupo usaremos ondas em uma corda.

Utilizamos 400 pontos podemos retirar a opção 'o' do *plt.plot*. Utilizaremos uma função de onda definida como $y = \text{sen}(2\pi \cdot f \cdot x)$, onde f é a frequência da onda. Inicialmente apresentaremos uma onda no gráfico, e posteriormente, como colocar duas ondas no mesmo gráfico (facilita a comparação entre elas). Para isso devemos gerar as listas de 400 pontos (x,y) para cada onda e simplesmente escrever o

comando *plot* para cada onda em sequência:

```
plt.plot(listax_onda1, listay_onda1)
plt.plot(listax_onda2, listay_onda2)
```

Para melhor diferenciar as ondas, podemos nomear cada onda com a função *label* de *plot* e escolher as cores de cada onda com a função *color*. A diferença entre as ondas escolhidas foi apenas na frequência, que assumirá os valores 1Hz e 0,5Hz.

6º momento (30 min): Apresente a atividade, deixe os alunos fazendo e circule pela sala tirando dúvidas.

Critérios de avaliação:

1. *Atingiu o objetivo da atividade: solucionou o problema usando o Python 3 como principal ferramenta – independente das estruturas e métodos usados, desde que o programa escrito seja funcional (7,0pt);*
2. *Estrutura e métodos:*
 - a. *O programa está organizado e fácil de entender (1,0pt);*
 - b. *Foi utilizado os métodos explanados em cada encontro (1,0pt);*
 - c. *O programa é bem comentado, explicando o raciocínio lógico (1,0pt).*

AVALIAÇÃO
<p>Atividade recomendada:</p> <ul style="list-style-type: none"> A partir das definições da Aula 4 e dada a onda $o = \text{Onda}(1.2, 0.25, \text{np.pi}/2)$, qual é o seu comprimento de onda e sua velocidade de propagação? <p><u>Entregue um código que faça a solução do problema e gere o gráfico da onda. Adicione o seu código e a imagem do gráfico.</u></p> <p><i>Dica: Pode-se reutilizar o código visto em aula.</i></p>
REFERÊNCIAS
<ol style="list-style-type: none"> G. Van Rossum, F. L. Drake. <i>Python 3 Reference Manual</i>. Scotts Valley, CA: CreateSpace; 2009. J. D. Hunter. <i>Matplotlib: A 2D Graphics Environment</i>. Computing in Science & Engineering. v. 9, n. 3, p. 90-95, 2007.
ANEXOS
<p>Será disponibilizado ao alunos via Google Classroom o código a seguir:</p> <p>https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/2Ao/Aula_4.py</p>

MODELO DE PLANEJAMENTO/PLANO DE AULA PARA CADA ENCONTRO

IDENTIFICAÇÃO		
Instituição Concedente: Colégio Impacto Centro de Ensino LTDA – ME Professor: Bruno Camargo Braga Componente Curricular: Física Unidade temática: Investigação Científica e Processos Criativos. Tema de estudo: Resolvendo equações e/ou sistemas de equações no Python		
Grupos 3 e 4: 2º ano do Ensino Médio	Tempo: 2 horas	Data: 19/05/2025
COMPETÊNCIAS E HABILIDADES		
(EMIFCG03) Utilizar informações, conhecimentos e ideias resultantes de investigações científicas para criar ou propor soluções para problemas diversos; (EMIFCG05) Questionar, modificar e adaptar ideias existentes e criar propostas obras ou soluções criativas, originais ou inovadoras, avaliando e assumindo riscos para lidar com as incertezas e colocá-las em prática.		
OBJETIVOS DE APRENDIZAGEM		
Utilizando a linguagem de programação Python (versão 3.11), ser capaz de: <ol style="list-style-type: none"> 1. Definir variáveis simbólicas e usa-las para montar equações; 2. Expandir e fatorar equações; 3. Resolver equações e sistemas de equações. 		
RECURSOS/MATERIAIS DIDÁTICOS		
Quadro, smartphone, Google Classroom, Google Colab, Pydroid 3.		
PROCEDIMENTOS METODOLÓGICOS		
1º momento (10 min): Saudações iniciais, chamada e anotação da ata de sala 2º momento (20 min): Revise os comandos vistos no encontro anterior. 3º momento (10 min): Apresente uma breve introdução da biblioteca sympy. Semelhante a aula 4, quando precisamos da biblioteca matplotlib de gráficos, aqui precisamos de uma nova biblioteca. Se o aluno estiver programando localmente (sem internet), ele precisará instalar a biblioteca sympy . Os alunos devem lembrar de adicionar no começo do código o comando: "import sympy as sp"		

Essa biblioteca "sympy" foi criada para resolver cálculos matemáticos exatos, ou seja, sem aproximações. Exemplo: a raiz quadrada de 8

- Sem o "sympy" podemos fazer $8^{0.5}$; isso gera o resultado aproximado **2.8284271247461903**.
- Com o "sympy" fazemos `sp.sqrt(8)` e isso gera **$2\sqrt{2}$** , que é a fatoração da raiz de 8.

4º momento (10 min): Descreva o método simbólico do sympy.

Até a aula passada, para descrever uma função como " $f = x + 2y$ " precisaríamos ou associar valores a x e y diretamente ou definir um método f que faz " $x + 2y$ ". Com o 'sympy' podemos definir x e y como símbolos e criar f, usando:

```
"x, y = sp.symbols('x y')
f = x + 2*y"
```

Assim podemos realizar operações como $f - x$ e obter resultados exatos ($f - x = 2y$). Com o "sympy" podemos pedir que ele simplifique expressões com o comando **expand** por exemplo: "`sp.expand((x-5)**2)`" gera " $x^2 - 10x + 25$ ".

Podemos também fazer o contrário e pedir para o "sympy" fatorar as expressões. Por exemplo: se quisermos as raízes da equação " $x^2 + 2x - 15 = 0$ ", podemos fazer "`factor(x**2 + 2*x - 15)`" que gera " $(x-3)(x + 5)$ ". É importante notar que nem sempre o *factor* gera algum resultado, pois o programa pode entender que a forma atual da expressão já está num formato simplês o suficiente. Devido a isso o *factor* resolve apenas os casos mais simples. Para soluções de qualquer equação usamos outro comando, o **solve**.

5º momento (10 min): Apresente o solve, linsolve e nonlinsolve. Comandos para resolver equações e sistemas de equações.

Além de equações, o comando "solve" pode receber uma lista de equações vinculadas em um único sistema e procurar possíveis soluções.

Exemplos de uso do **solve**:

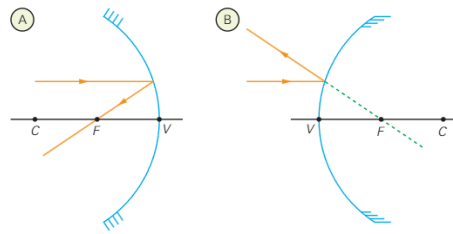
- Para $x^2 - 2 = 0$, escrevemos "`sp.solve(x**2-2, x)`" que gera " $[-\sqrt{2}, \sqrt{2}]$ ";
- Para $x^2 - 16 = 9$, escrevemos "`sp.solve(sp.Eq(x**2-16,9), x)`" que gera " $[-5, 5]$ ";
- Para o sistema $\{x - 2y = 0, x + y = 18\}$, escrevemos "`sp.linsolve([x-2*y, sp.Eq(x+y,18)], (x,y))`".
- Para o sistema $\{x^2 - 2y = 0, 6x + y = -18\}$, escrevemos "`sp.nonlinsolve([x**2-2*y, sp.Eq(6*x+y,-18)], (x,y))`".

Obs: se as variáveis do sistema de equações não tiverem expoentes (elevado a 1) usamos o comando **linsolve()**, para sistemas onde qualquer uma das variáveis tenha expoente diferente de 1 usamos o comando **nonlinsolve()**, tanto para casos com raiz quadrada ($\sqrt{x} - 3 = 0$), quanto para frações ($\frac{1}{x} + 2 = 4$). Então, na optica geométrica com espelhos esféricos, que usamos a equação de Gauss $\frac{1}{f} = \frac{1}{p} + \frac{1}{p'}$, se precisarmos de outra equação como a do aumento de imagem

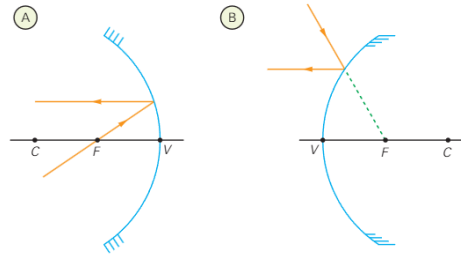
$\frac{i}{o} = -\frac{p'}{p}$, devemos usar o **nonlinsolve()**.

6º momento (30 min): Revise com os alunos os conceitos óptica de espelhos esféricos.

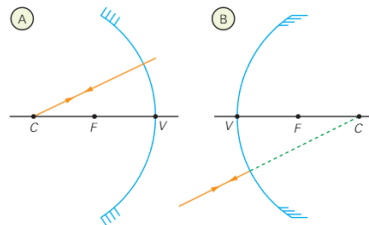
Todo raio de luz que incide paralelamente ao eixo principal é refletido em direção ao foco principal.



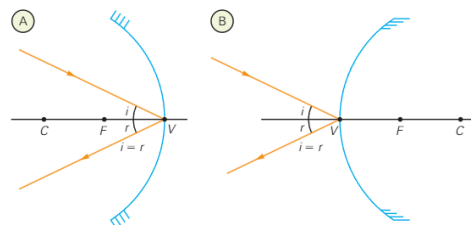
Todo raio de luz que incide em direção ao foco principal é refletido paralelamente ao eixo principal.



Todo raio de luz que incide em direção ao centro de curvatura é refletido sobre si mesmo.



Todo raio de luz que incide sobre o vértice é refletido simetricamente em relação ao eixo principal.



Com o uso de pelo menos dois desses raios, independente de onde colocarmos o objeto, conseguiremos construir a imagem e verificar suas características:

1. Ela é real (está do mesmo lado que o objeto) ou virtual (o espelho está entre o objeto e a imagem);
2. Direita (mesma direção e sentido do objeto) ou invertida;
3. Maior ou menor que o objeto;
4. Imprópria (caso os raios nunca se cruzem). Ocorre quando o objeto está no foco do espelho.

Equações importantes:

1. **Equação de Gauss:** A igualdade que relaciona a posição do objeto (p), da imagem (p') e a distância focal (f) é dada por $\frac{1}{f} = \frac{1}{p} + \frac{1}{p'}$;
2. O aumento linear transversal (A) é a relação entre o tamanho da imagem (i) e o tamanho do objeto (o), dado por $A = \frac{i}{o} = -\frac{p'}{p}$;

7º momento (30 min): Apresente a atividade, deixe os alunos fazendo e circule pela sala tirando dúvidas.

Crterios de avaliao:

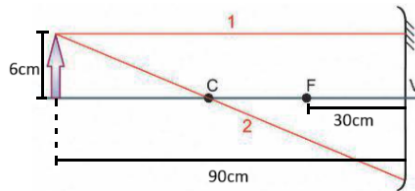
1. Atingiu o objetivo da atividade: solucionou o problema usando o Python 3 como principal ferramenta – independente das estruturas e mtodos usados, desde que o programa escrito seja funcional **(7,0pt)**;
2. Estrutura e mtodos:
 - a. O programa est organizado e fcil de entender **(1,0pt)**;
 - b. Foi utilizado os mtodos explanados em cada encontro **(1,0pt)**;
 - c. O programa est bem comentado, explicando o raciocnio lgico **(1,0pt)**.

AVALIAO

Atividade recomendada:

Lembrete:

1. a Equao de Gauss(Espelhos Esfricos), $\frac{1}{f} = \frac{1}{p} + \frac{1}{p'}$; onde f é a distncia focal, p é a distncia entre o objeto e o vrtese e p' é a distncia entre a imagem e o vrtese.
2. E a relao de aumento linear: $\frac{i}{o} = -\frac{p'}{p}$; onde i é o tamanho da imagem e o é o tamanho do objeto.
- Considere um objeto luminoso colocado sobre o eixo principal de um espelho esférico cncavo. Sabendo que a distncia focal do espelho é 30 cm, que a distncia do objeto ao espelho é 90 cm e que a altura do objeto é 6,0 cm, calcule a distncia da imagem ao espelho e a altura da imagem, ambas em centmetros.



Resolva o problema utilizando os comandos do **sympy** e entregue o código usado.

REFERÊNCIAS

1. G. Van Rossum, F. L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace; 2009.
2. A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F.

Pedregosa, M. J. Curry, A. R. Terrel, Š. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, A. Scopatz. *SymPy: symbolic computing in Python*. PeerJ Computer Science. v. 3, p. e103, 2017. Disponível em: <https://doi.org/10.7717/peerj-cs.103>.

ANEXOS

Será disponibilizado ao alunos via Google Classroom o código a seguir:

https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/2Ano/Aula_5.py

MODELO DE PLANEJAMENTO/PLANO DE AULA PARA CADA ENCONTRO

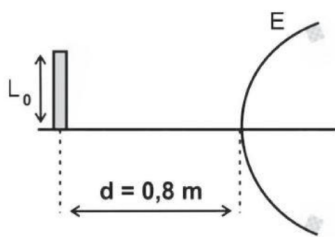
IDENTIFICAÇÃO		
Instituição Concedente: Colégio Impacto Centro de Ensino LTDA – ME Professor: Bruno Camargo Braga Componente Curricular: Física Unidade temática: Investigação Científica e Processos Criativos. Tema de estudo: Resolvendo equações e/ou sistemas de equações no Python – Parte 2		
Grupos 3 e 4: 2º ano do Ensino Médio	Tempo: 2 horas	Data: 23/05/2025
COMPETÊNCIAS E HABILIDADES		
(EMIFCG03) Utilizar informações, conhecimentos e ideias resultantes de investigações científicas para criar ou propor soluções para problemas diversos; (EMIFCG05) Questionar, modificar e adaptar ideias existentes e criar propostas obras ou soluções criativas, originais ou inovadoras, avaliando e assumindo riscos para lidar com as incertezas e colocá-las em prática.		
OBJETIVOS DE APRENDIZAGEM		
Utilizando a linguagem de programação Python (versão 3.11), ser capaz de: <ol style="list-style-type: none"> 1. Definir variáveis simbólicas e usa-las para montar equações; 2. Resolver equações e sistemas de equações. 		
RECURSOS/MATERIAIS DIDÁTICOS		
Quadro, smartphone, Google Classroom, Google Colab, Pydroid 3.		
PROCEDIMENTOS METODOLÓGICOS		
1º momento (10 min): Saudações iniciais, chamada e anotação da ata de sala 2º momento (20 min): Revise os comandos vistos no encontro anterior. Como continuação da aula 5, os alunos devem lembrar de adicionar no começo do código o comando: "import sympy as sp"		
Lembre os alunos dos usos do comando solve : <ul style="list-style-type: none"> • Para $x^2 - 2 = 0$, escrevemos "sp.solve(x**2-2, x)" que gera "[-sqrt(2), sqrt(2)]"; • Para $x^2 - 16 = 9$, escrevemos "sp.solve(sp.Eq(x**2-16,9), x)" que gera "[-5, 		

5];

- Para o sistema $\{x - 2y = 0, x + y = 18\}$, escrevemos "sp.linsolve([x-2*y, sp.Eq(x+y,18)], (x,y))".
- Para o sistema $\{x^2 - 2y = 0, 6x + y = -18\}$, escrevemos "sp.nonlinsolve([x**2-2*y, sp.Eq(6*x+y,-18)], (x,y))".

3º momento (45 min): Aqui resolveremos junto com os alunos um exemplo mais complexo que mistura óptica geométrica com dilatação térmica.

Problema: "Uma barra delgada está em uma temperatura na qual o seu comprimento é igual $L_0=100$ cm. A barra, de coeficiente de dilatação linear $8,0 \cdot 10^{-5} \text{ } ^\circ\text{C}^{-1}$ é, então, colocada a uma distância $d=80$ cm do vértice de um espelho curvo. O espelho possui um raio de curvatura de 160 cm.



Para se fazer a imagem crescer 0,5 cm, pode-se:

- aproximar a barra 10 cm;
- afastar a barra 5 cm;
- aquecer a barra em $125 \text{ } ^\circ\text{C}$

Para resolver o problema precisamos escrever as equações necessárias, que serão a equação de Gauss para espelhos esféricos, a relação de aumento objeto/imagem e a equação de dilatação linear. Teremos assim 7 variáveis e 3 equações base, então, use os comandos:

"f, p, pl, i, o, dL, dT = symbols('f p pl i o dL dT')

gauss = Eq(1/f, 1/p + 1/pl)

aumento = Eq(i/o, -pl/p)

dilatacao = Eq(dL, L0*alfa*dT)"

Nesse exemplo, a imagem é virtual ($pl < 0$) e direita ($i > 0$). Tanto o foco quanto o raio são negativos ($f < 0$, $CV < 0$). Para espelhos esféricos de Gauss, a distância focal f é a metade do raio de curvatura CV , e portanto, será -80cm .

Substituindo f e p que sabemos na equação "gauss", usamos o comando *solve* para determinar a distância da imagem pl . Substituindo então os valores p , pl e $o=L_0$ na equação "aumento" e usando *solve*, obtemos a altura da imagem i .

Podemos agora realizar a verificação pedida no problema, aumentar i e escolher a resposta correta. Se for o caso das distâncias p e pl permanecerem iguais, então quanto que o objeto L_0 precisa dilatar?

Nesse caso, teremos a equação: "novo_o = (i+0.5)*(-p/pl)". Substituindo os valores de i , p e pl e usando o *solve* para o tamanho do objeto, conseguiremos a variação de tamanho dL .

Substituindo agora $alfa$, L_0 e " $dL=\text{novo_o} - L_0$ " na equação "dilatacao" e usando *solve* para dT , obtemos a resposta (c).

4º momento (45 min): Apresente a atividade, deixe os alunos fazendo e circule

pela sala tirando dúvidas.

Crítérios de avaliação:

1. *Atingiu o objetivo da atividade: solucionou o problema usando o Python 3 como principal ferramenta – independente das estruturas e métodos usados, desde que o programa escrito seja funcional (7,0pt);*
2. *Estrutura e métodos:*
 - a. *O programa está organizado e fácil de entender (1,0pt);*
 - b. *Foi utilizado os métodos explanados em cada encontro (1,0pt);*
 - c. *O programa é bem comentado, explicando o raciocínio lógico (1,0pt).*

AVALIAÇÃO

Atividade recomendada:

- Resolva o problema feito em sala na Aula 6, porém, considerando desta vez que a temperatura não mude. Nesse caso quanto que as distâncias do objeto (p) e imagem (pi) devem mudar?

Resolva o problema utilizando os comandos do **sympy** e entregue o código usado.

REFERÊNCIAS

1. G. Van Rossum, F. L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace; 2009.
2. A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. J. Curry, A. R. Terrel, Š. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, A. Scopatz. *SymPy: symbolic computing in Python*. PeerJ Computer Science. v. 3, p. e103, 2017. Disponível em: <https://doi.org/10.7717/peerj-cs.103>.

ANEXOS

Será disponibilizado ao alunos via Google Classroom o código a seguir:

https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/2Ano/Aula_6.py

MODELO DE PLANEJAMENTO/PLANO DE AULA PARA CADA ENCONTRO

IDENTIFICAÇÃO		
Instituição Concedente: Colégio Impacto Centro de Ensino LTDA – ME Professor: Bruno Camargo Braga Componente Curricular: Física Unidade temática: Investigação Científica e Processos Criativos. Tema de estudo: Solução e Explicação das Atividades		
Grupos 3 e 4: 2º ano do Ensino Médio	Tempo: 2 horas	Data: 30/05/2025
COMPETÊNCIAS E HABILIDADES		
(EMIFCG03) Utilizar informações, conhecimentos e ideias resultantes de investigações científicas para criar ou propor soluções para problemas diversos; (EMIFCG05) Questionar, modificar e adaptar ideias existentes e criar propostas obras ou soluções criativas, originais ou inovadoras, avaliando e assumindo riscos para lidar com as incertezas e colocá-las em prática.		
OBJETIVOS DE APRENDIZAGEM		
Utilizando a linguagem de programação Python (versão 3.11), ser capaz de: <ol style="list-style-type: none"> 1. Compreender os possíveis erros quanto a solução de problemas de Física; 2. Encontrar soluções semelhantes a apresentada pelo professor; 		
RECURSOS/MATERIAIS DIDÁTICOS		
Quadro, smartphone, Google Classroom, Google Colab, Pydroid 3.		
PROCEDIMENTOS METODOLÓGICOS		
<p>1º momento (15 min): Saudações iniciais, chamada e anotação da ata de sala</p> <p>2º momento (5 min): Nesta aula será apresentado as correções das atividades 1 a 4 feitas pelos alunos durante o projeto. Mencione aos estudantes que será apresentado sua solução como professor e que a aula será para tirar dúvidas.</p> <p>OBS: os tempos dos momentos a seguir estão exagerados, pois é para usa-los não apenas para apresentar a solução do professor, mas para tirar dúvidas dos alunos e falar de problemas semelhantes caso nenhum aluno queira se pronunciar.</p> <p>Reapresente as atividades propostas aos alunos uma a uma e discuta com eles</p>		

uma solução (e sua implementação em Python). As atividades foram:

1. **3º momento (25 min): Faça um código que peça uma sequência de valores e calcule a média deles.**

Explicação: Esta atividade cobra o uso de comandos comuns de interação usuário-código e manipulação de textos. Uma solução é pedir que seja informado a sequência de valores em um formato específico, pois assim o programa conseguirá facilmente separar os valores para realizar a operação. O aluno deve lembrar que a resposta vem no formato texto e deve ser transformada para um tipo numérico. Utilizando um ciclo conseguimos somar todos os valores e apresentá-los usando o comando “print”.

2. **4º momento (25 min): Faça uma código que peça um número inteiro e nos diz se o número é par ou ímpar.**

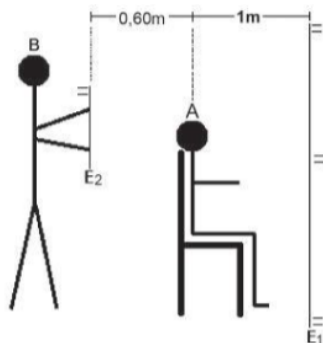
Explicação: Esta atividade cobra o uso de comandos comuns de interação usuário-código e manipulação de textos. Como o número informado deve ser par ou ímpar – ele será um número natural. Com isso basta converter para inteiro a resposta e utilizar o comando para resto de divisão “%”. Se ao dividir por 2, der resto zero o número informado é par, caso contrário será ímpar.

3. **5º momento (25 min): Quando jogamos uma pedra em um lago de águas calmas, são produzidas ondas periódicas que percorrem 5 m em 10 s. Sendo a distância entre duas cristas sucessivas igual a 40 cm, teremos que a frequência e a velocidade de propagação dessas ondas são, respectivamente, iguais a?**

Entregue um código que faça a solução do problema e imprime a resposta.

Explicação: Uma solução simples envolve utilizar as definições de período e frequência como apresentado em aula e imprimir os valores usando ambas funções: “Frequencia(Periodo(0.40))” que devolve a frequência calculada com o período, que em si é calculado pelo comprimento de onda (distância entre duas cristas sucessivas) – note que para unidade ficar correta (como Hz) precisamos converter “cm” para a unidade do SI (metro). A velocidade de propagação foi dada no próprio exercício $v = \frac{5m}{10s} = 0,5 \text{ m/s}$.

4. **6º momento (25 min): Considerando que ao usar espelhos planos, a distância da imagem ao espelho é igual a distância do espelho ao objeto. Um barbeiro B segura um espelho plano E2, de espessura desprezível, paralelamente a outro espelho plano E1, também de espessura desprezível, permitindo que seu cliente A observe, no espelho E1, o seu corte de cabelo na parte posterior da cabeça.**



Determine a menor distância entre o cliente e a imagem que ele observa da sua nuca no espelho E1, considerando que a cabeça do cliente também possui dimensões desprezíveis.

Entregue um código que faça a solução do problema e imprime a resposta.

Dica: use o método "def".

Explicação: Para resolver esta atividade devemos considerar as distâncias corretas entre imagem e objeto. Para espelhos planos, a distância objeto-espelho será igual a distância espelho-imagem. Considerando a imagem de um espelho como objeto para determinar a imagem do segundo espelho, e dividindo o problema como:

1. O objeto será a nuca de A e a imagem A' será localizada pelo espelho E_2 .
2. O objeto será a imagem A' e a imagem A'' será localizada pelo espelho E_1 .
3. A imagem A'' da nuca estará a distância de A como a soma entre a distância A- E_1 e a distância calculada em (2) E_1 -A''.

Conseguimos determinar a distância com uma definição simples de distância objeto-imagem, e aplicando ela com as três considerações acima.

AVALIAÇÃO

- Tira dúvidas.

REFERÊNCIAS

1. G. Van Rossum, F. L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace; 2009.
2. J. D. Hunter. *Matplotlib: A 2D Graphics Environment*. Computing in Science & Engineering. v. 9, n. 3, p. 90-95, 2007.
3. A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. J. Curry, A. R. Terrel, Š. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, A. Scopatz. *SymPy: symbolic computing in Python*. PeerJ Computer Science. v. 3, p. e103, 2017. Disponível em: <https://doi.org/10.7717/peerj-cs.103>.

ANEXOS

Será disponibilizado ao alunos via Google Classroom os códigos discutidos em aula.

https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/Atividade_1.py

https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/Atividade_2.py

https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/2Ano/Atividade_3.py

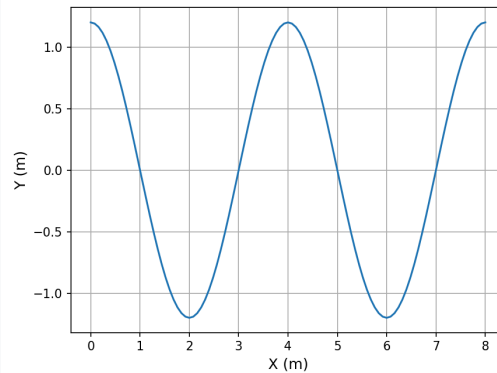
https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/2Ano/Atividade_4.py

MODELO DE PLANEJAMENTO/PLANO DE AULA PARA CADA ENCONTRO

IDENTIFICAÇÃO		
Instituição Concedente: Colégio Impacto Centro de Ensino LTDA – ME Professor: Bruno Camargo Braga Componente Curricular: Física Unidade temática: Investigação Científica e Processos Criativos. Tema de estudo: Solução e Explicação das Atividades – Parte 2		
Grupos 3 e 4: 2º ano do Ensino Médio	Tempo: 2 horas	Data: 02/06/2025
COMPETÊNCIAS E HABILIDADES		
(EMIFCG03) Utilizar informações, conhecimentos e ideias resultantes de investigações científicas para criar ou propor soluções para problemas diversos; (EMIFCG05) Questionar, modificar e adaptar ideias existentes e criar propostas obras ou soluções criativas, originais ou inovadoras, avaliando e assumindo riscos para lidar com as incertezas e colocá-las em prática.		
OBJETIVOS DE APRENDIZAGEM		
Utilizando a linguagem de programação Python (versão 3.11), ser capaz de: <ol style="list-style-type: none"> 1. Compreender os possíveis erros quanto a solução de problemas de Física; 2. Encontrar soluções semelhantes a apresentada pelo professor; 		
RECURSOS/MATERIAIS DIDÁTICOS		
Quadro, smartphone, Google Classroom, Google Colab, Pydroid 3.		
PROCEDIMENTOS METODOLÓGICOS		
<p>1º momento (15 min): Saudações iniciais, chamada e anotação da ata de sala. Nesta aula será apresentado as correções das atividades 5 a 7 feitas pelos alunos durante o projeto.</p> <p>Reapresente as atividades propostas aos alunos uma a uma e discuta com eles uma solução (e sua implementação em Python). As atividades foram:</p> <ol style="list-style-type: none"> 1. 2º momento (30 min): A partir das definições da Aula 4 e dada a onda $o = \text{Onda}(1.2, 0.25, np.pi/2)$, qual é o seu comprimento de onda e sua velocidade de propagação? <u>Entregue um código que faça a solução do problema e gere o gráfico da onda. Adicione o seu código e a imagem do gráfico.</u> 		

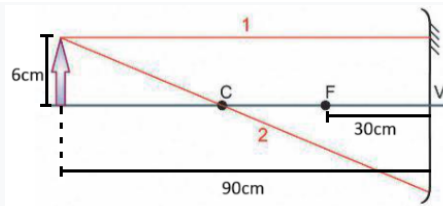
Dica: Pode-se reutilizar o código visto em aula.

Explicação: A velocidade de propagação obtemos ao multiplicar a frequência pelo comprimento de onda $v = f * \lambda$. Pela forma que foi definido o método de onda na aula 4 “Onda(amplitude,frequencia,fase)”, sabemos que a frequência deve ser $f = 0,25 \text{ Hz}$. Quanto ao comprimento de onda λ , basta construir o gráfico da onda e medir a distância de duas cristas (ou vales) sucessivos.



Como vemos no gráfico, a onda começa o eixo x exatamente em uma crista, assim, a posição em x da segunda crista nos diz o comprimento de onda $\lambda = 4 \text{ m}$. Portanto, $v = f * \lambda = 0,25 \text{ Hz} * 4 \text{ m} = 1 \text{ m/s}$.

2. **3º momento (30 min):** Considere um objeto luminoso colocado sobre o eixo principal de um espelho esférico côncavo. Sabendo que a distância focal do espelho é 30 cm, que a distância do objeto ao espelho é 90 cm e que a altura do objeto é 6,0 cm, calcule a distância da imagem ao espelho e a altura da imagem, ambas em centímetros.



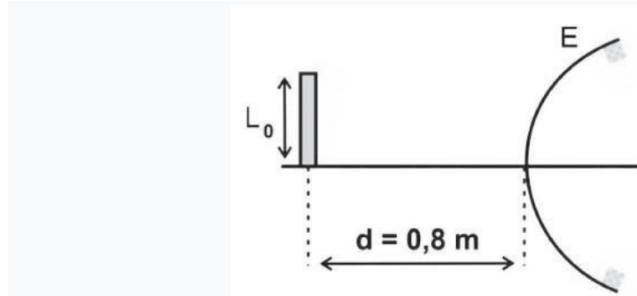
Explicação: O enunciado fornece a distância focal ($f = 30 \text{ cm}$), do objeto ($p = 90 \text{ cm}$) e o tamanho do objeto ($o = 6 \text{ cm}$). Aplicando a equação de Gauss $\frac{1}{f} = \frac{1}{p} + \frac{1}{pl}$ obtemos a distância da imagem que chamaremos aqui de “ pl ”, pois o sympy daria erro com p' devido a definição de textos no Python. Com pl , poderemos então utilizar a relação de aumento $\frac{i}{o} = -\frac{pl}{p}$ e resolver o problema com o uso do “solve”. Lembrando que o sinal negativo significa apenas imagem invertida.

3. **4º momento (45 min):** Resolva o problema feito em sala na Aula 6, porém, considerando desta vez que a temperatura não mude. Nesse caso quanto que as distâncias do objeto (p) e imagem (pl) devem

mudar?

Explicação: O problema feito em sala foi:

- Uma barra delgada está em uma temperatura na qual o seu comprimento é igual $L_0 = 100$ cm. A barra, de coeficiente de dilatação linear $8,0 \cdot 10^{-5} \text{ } ^\circ\text{C}^{-1}$, é, então, colocada a uma distância $d = 80$ cm do vértice de um espelho curvo.



O espelho possui um raio de curvatura de 160 cm. **Para se fazer a imagem crescer 0,5 cm, pode-se:**

- aproximar a barra 10 cm;
- afastar a barra 5 cm;
- aquecer a barra em $125 \text{ } ^\circ\text{C}$

Para resolver o problema precisamos escrever as equações necessárias, que serão a equação de Gauss para espelhos esféricos, a relação de aumento objeto/imagem e a equação de dilatação linear. Teremos assim 7 variáveis e 3 equações base, então, use os comandos:

"f, p, pl, i, o, dL, dT = symbols('f p pl i o dL dT')"

gauss = Eq(1/f, 1/p + 1/pl)

aumento = Eq(i/o, -pl/p)

dilatacao = Eq(dL, L0*alfa*dT)"

Nesse exemplo, a imagem é virtual ($pl < 0$) e direita ($i > 0$). Tanto o foco quanto o raio são negativos ($f < 0$, $CV < 0$). Para espelhos esféricos de Gauss, a distância focal f é a metade do raio de curvatura CV , e portanto, será -80 cm.

Substituindo f e p que sabemos na equação "gauss", usamos o comando *solve* para determinar a distância da imagem pl . Substituindo então os valores p , pl e $o = L_0$ na equação "aumento" e usando *solve*, obtemos a altura da imagem i . Podemos agora realizar a verificação pedida no problema, aumentar i e verificar a resposta.

Diferentemente da solução em sala, queremos que a temperatura permaneça constante. Então, escrevemos uma nova equação de aumento usando a altura calculada da imagem e usamos o comando "nonlinsolve" para resolver o sistema de equações formado pela nova equação de aumento e a equação de Gauss, considerando como variáveis as distâncias do objeto e da imagem.

AVALIAÇÃO
<ul style="list-style-type: none">• Tira dúvidas.
REFERÊNCIAS
<ol style="list-style-type: none">1. G. Van Rossum, F. L. Drake. <i>Python 3 Reference Manual</i>. Scotts Valley, CA: CreateSpace; 2009.2. J. D. Hunter. <i>Matplotlib: A 2D Graphics Environment</i>. Computing in Science & Engineering. v. 9, n. 3, p. 90-95, 2007.3. A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. J. Curry, A. R. Terrel, Š. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, A. Scopatz. <i>SymPy: symbolic computing in Python</i>. PeerJ Computer Science. v. 3, p. e103, 2017. Disponível em: https://doi.org/10.7717/peerj-cs.103.
ANEXOS
<p>Será disponibilizado ao alunos via Google Classroom os códigos discutidos em aula.</p> <p>https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/2Ano/Atividade_5.py</p> <p>https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/2Ano/Atividade_6.py</p> <p>https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/2Ano/Atividade_7.py</p>

MODELO DE PLANEJAMENTO/PLANO DE AULA PARA CADA ENCONTRO

IDENTIFICAÇÃO		
Instituição Concedente: Colégio Impacto Centro de Ensino LTDA – ME Professor: Bruno Camargo Braga Componente Curricular: Física Unidade temática: Investigação Científica e Processos Criativos. Tema de estudo: Avaliação e depoimento		
Grupos 3 e 4: 2º ano do Ensino Médio	Tempo: 2 horas	Data: 06/06/2025
COMPETÊNCIAS E HABILIDADES		
(EMIFCG03) Utilizar informações, conhecimentos e ideias resultantes de investigações científicas para criar ou propor soluções para problemas diversos; (EMIFCG05) Questionar, modificar e adaptar ideias existentes e criar propostas obras ou soluções criativas, originais ou inovadoras, avaliando e assumindo riscos para lidar com as incertezas e colocá-las em prática.		
OBJETIVOS DE APRENDIZAGEM		
Utilizando a linguagem de programação Python (versão 3.11), ser capaz de: <ol style="list-style-type: none"> 1. Conseguir construir e ler corretamente gráficos; 2. Resolver sistemas de equações. 		
RECURSOS/MATERIAIS DIDÁTICOS		
Quadro, smartphone, Google Classroom, Google Colab, Pydroid 3.		
PROCEDIMENTOS METODOLÓGICOS		
1º momento (10 min): Saudações iniciais, chamada e anotação da ata de sala. 2º momento (55 min): Reaplicação da avaliação diagnóstica 3º momento (55 min): Deixe os alunos que não terminaram a avaliação fazerem a avaliação. Indique aos que terminaram para preencher o depoimento de participação via Google Forms.		
AVALIAÇÃO		

Atividades recomendadas:

- Refazer a avaliação diagnóstica;
- Responder o depoimento de participação.

REFERÊNCIAS

1. G. Van Rossum, F. L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace; 2009.
2. J. D. Hunter. *Matplotlib: A 2D Graphics Environment*. Computing in Science & Engineering. v. 9, n. 3, p. 90-95, 2007.
3. A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. J. Curry, A. R. Terrel, Š. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, A. Scopatz. *SymPy: symbolic computing in Python*. PeerJ Computer Science. v. 3, p. e103, 2017. Disponível em: <https://doi.org/10.7717/peerj-cs.103>.

ANEXOS

A avaliação diagnóstica será a mesma descrita no encontro 1.

Descrição do formulário do depoimento de participação:

1. Nome do aluno(a):
2. Como você descreveria sua experiência geral neste projeto?
3. Quais foram os aspectos mais positivos do projeto?
4. Houve alguma dificuldade que você enfrentou antes de começar o projeto (ex: dificuldade em resolver equações ou utilizar gráficos)? Como o programa ajudou a supera-la?
5. Como você se sente em relação ao seu desempenho escolar após participar do projeto?
6. Alguma sugestão ou comentário adicional que gostaria de compartilhar?
7. **Permissão para Uso de Depoimento:**
 Você autoriza o uso do seu depoimento em materiais de divulgação da universidade Uniasselvi?
 - a. Sim
 - b. Não