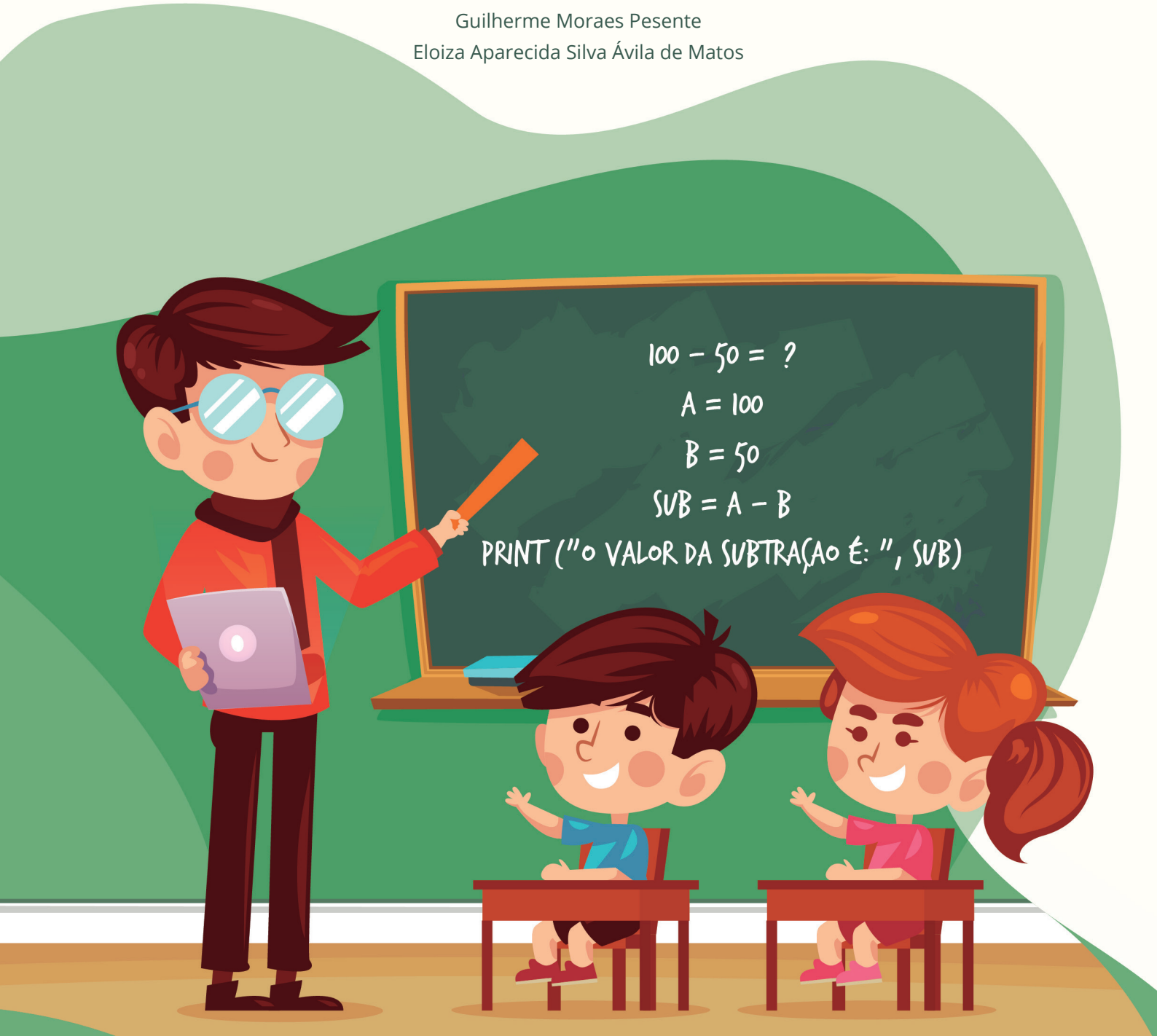
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENSINO DE CIÊNCIA E TECNOLOGIA
MESTRADO PROFISSIONAL EM ENSINO DE CIÊNCIA E TECNOLOGIA

ENSINE MATEMÁTICA POR MEIO DA LINGUAGEM PYTHON

CADERNO PEDAGÓGICO

Guilherme Moraes Pesente
Eloiza Aparecida Silva Ávila de Matos



Ponta Grossa / 2019

ORIENTAÇÕES AOS PROFESSORES

O avanço tecnológico proporcionou a elaboração de estudos a respeito dos impactos e contribuições das tecnologias no meio acadêmico, permitindo assim, o desenvolvimento de softwares voltados a aprendizagem de cada indivíduo. No princípio com softwares como o LOGO que foi desenvolvido pelo MIT e utilizado para o ensino de álgebra, posteriormente, sua evolução para o Scratch e App Inventor.

Por meio desta evolução, o século XXI serviu como ponte para o avanço tecnológico, permitindo assim, a sua aquisição por parte de usuários domésticos e instituições de ensino diversas. Desta forma, se tornou importante que estas instituições, se possível, passe a utilizar as diversas formas de tecnologias em prol da educação, seja por meio de smartphones, tablets, computadores, projetores interativos, lousas digitas, entre outros equipamentos tecnológicos, afim de criar maior interação dos alunos em sala de aula, buscando a investigação de conteúdos e suas inúmeras soluções (PRIMO, 2003).

As instituições de ensino devem analisar a importância da utilização de tecnologias no meio acadêmico, pois em uma geração que constantemente recebe atualizações tecnológicas, seja por meio de softwares ou hardwares, esta se tornou fundamental para a interação em sala de aula entre alunos e professores, conforme apontado por Machado e Lima (2017, p.2):

> ***A tecnologia é de primordial necessidade, pois promove oportunidades de aprendizagem e interatividade tanto para o professor como para o aluno.***
>
> ***A escola é um local de constante transformação e a tecnologia educacional é uma dessas ferramentas para a transformação.***
>
> *(MACHADO e LIMA, 2017, p.2)*

Mas a tecnologia sozinha não produz resultados, é preciso que técnicas e estudos sejam anexados ao processo de ensino, se tornando importante a utilização de teorias que contribuam para a construção e fortalecimento no processo cognitivo de cada aluno. Teorias como da Aprendizagem Significativa de Ausubel (1968) e Construcionismo de Papert (1985), onde uma pode ser utilizada para complementar a outra.

É importante para o professor que o conhecimento prévio existente no processo cognitivo do aluno, seja fortalecido com a utilização das tecnologias educacionais, pois

um conhecimento prévio, que para Ausubel é um conhecimento específico sobre determinado assunto (AUSUBEL, 2000), pode interagir com a utilização da máquina computacional, dando continuidade a este conhecimento por meio da teoria do construcionismo de Papert, que define o computador como ponte para construção do conhecimento (PAPERT, 2008).

Em disciplinas onde os conteúdos podem se tornar complexos para alunos que não possuem uma base acadêmica fortificada, a utilização de linguagem de programação vem a se tornar um meio para que os alunos investiguem situações para a resolução de cada atividade proposta, os motivando para que os resultados sejam alcançados.

Desta forma, o caderno pedagógico tem como objetivo propor, por meio de códigos computacionais, meio e estratégias que possam contribuir na construção do conhecimento de cada aluno, utilizando os princípios da aprendizagem significativa e teoria do construcionismo em alunos do ensino fundamental, sendo possível sua utilização em outros grupos por meio de adaptação para cada necessidade.

Na sequência, são apresentados exemplos de sua utilização em um público do 6ª ano do Ensino Fundamental II, no conteúdo de matemática, assim como sua metodologia, sugestões de uso e comentários do autor, afim de favorecer ao máximo a interação entre professo, aluno e máquina.

INTRODUÇÃO E INSTALAÇÃO DA LINGUAGEM PYTHON

A linguagem de programação Python foi desenvolvida no ano de 1991 pelo holandês Guido Van Rossun, suas principais características são: linguagem de alto nível; interpretada; orientada a objetos e de tipagem forte e dinâmica, sendo gerida pela Python Software Foundation, uma organização sem fins lucrativos. Por possuir fácil sintaxe, esta linguagem é aconselhada para pessoas que estejam dando início aos estudos em ciência da computação.

Com estas características, estudos estão sendo desenvolvidos e a linguagem sendo avaliada para sua utilização no público infantil, como exemplo, o livro "**Ensine seus filhos a programar – Um guia amigável aos pais para a programação Python**", do autor Bryson Payne, que trata do ensino introdutório a programação para crianças de forma lúdica, o que vem beneficiando a criação e compartilhamento de estudos e sabres sobre o assunto, possibilitando que linguagens como Python possam alcançar um público maior.

Por possuir uma fácil leitura e desenvolvimento dos códigos, python está entre as linguagens de programação mais estudadas e utilizadas, sendo comumente usada por cursos de graduação em computação, engenharias e por meio de materiais disponibilizados na internet. Sua estrutura simples de desenvolvimento se destaca, criando possibilidade de aprendizado em diversos públicos.

INSTALAÇÃO

Comentários do autor:

A linguagem Python será instalada no sistema operacional Windows. No sistema Mac OS X o procedimento se torna similar, apenas no Linux que sua instalação se dá por meio do terminal, ao invés de softwares executáveis. O python está na sua terceira versão, sendo assim importante instalar da versão 3.0 em diante.

A linguagem Python está disponível para os principais Sistemas Operacionais, sendo Windows, Linux e Mac OS X. O processo de instalação Python é simples e rápido, sendo preciso em primeiro lugar fazer o seu download por meio de sua página oficial (python.org) na aba Downloads, portanto, escolha seu sistema operacional e faça o download posteriormente.

Figura 1: Download linguagem Python
Fonte: Autoria própria

Feito isto, execute e siga o procedimento padrão, marcando as duas caixas disponíveis em sua parte inferior. A primeira relacionada a instalação para todos os usuários e a segunda para anexar o Path da versão 3.7 direto no sistema. Feito isto, basta instalar ou personalizar sua instalação (é aconselhado utilizar a instalação padrão).

Figura 2: Instalação linguagem Python
Fonte: Autoria própria

Ao finalizar este procedimento, a linguagem Python estará instalada, podendo ser encontrada ao clicar no menu iniciar e pesquisar por IDLE, ao encontrar, abra-o e já está em seu ambiente programável, denominado Shell. Note que na parte superior será informado a versão da IDLE que foi instalada, neste caso, 3.7.4.

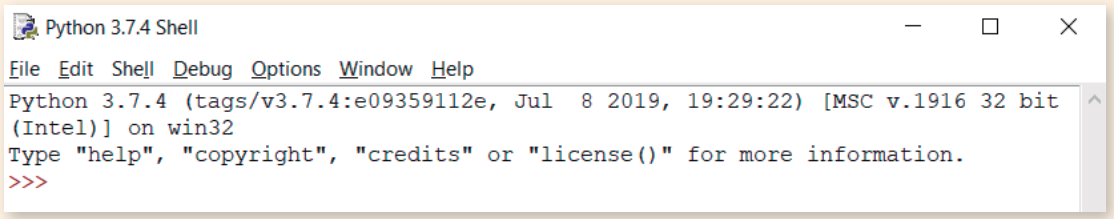


Figura 3: Shell linguagem Python
Fonte: Autoria própria

INTRODUÇÃO A LINGUAGEM

O professor de laboratório deve utilizar aulas de introdução a programação Python, visto que, muitos (senão todos) estão tendo o primeiro contato com uma linguagem de programação, portanto, é importante que este professor explique o ambiente programável, assim como os comandos básicos, decisão, repetição, entre outros.

Estas aulas introdutórias são essenciais, pois servirá como motivação e ambientação de cada aluno, portanto, deve-se existir o incentivo do professor, explicações claras e simples e que este se especialize constantemente, afim de conseguir responder cada questionamento dos alunos. Desta forma, as aulas vão se tornar dinâmica e os alunos vão encarar como desafios possíveis de serem vencidos.

É importante o constante acompanhamento do professor de laboratório no processo de aprendizagem do aluno, pois cada aluno possui um tempo e um método diferente para obter o conhecimento transmitido, principalmente ao se utilizar programação de computadores, pois o não acompanhamento e a não criação de novas explicações e métodos de ensino para estes alunos, podem posteriormente os assustá-los, desmotiva-los e os afastarem deste processo de construção do conhecimento.

Assim, se torna importante a interação entre professor de laboratório e professor de sala de aula, ocorrendo constantemente o diálogo e a criação de técnicas que contribuem no processo de obtenção do conhecimento de cada aluno participante neste processo de aprender.

OPERAÇÕES MATEMÁTICAS

OBJETIVOS

Compreensão sobre o método de resolução dos quatro operações matemáticas, adição, subtração, multiplicação e divisão.

CONTEÚDO

Operações Matemáticas (adição, subtração, multiplicação e divisão).

PROCEDIMENTO METODOLÓGICO

É preciso que o professor de laboratório leve para o quadro exemplos para cada operação e que incentive a participação de cada aluno, de forma individual e grupal, sendo resolvido estes exemplos no quadro e passando posteriormente algumas atividades em caderno. Por meio dos alunos, as atividades devem ser respondidas e corrigidas pelo professor de laboratório junto a estes. Posteriormente, deve-se levar a IDLE da linguagem Python as atividades já feitas e posteriormente novos desafios em forma de atividades, incentivando e ajudando cada aluno.

SUGESTÃO

É aconselhável que sejam desenvolvidos códigos onde o aluno pré-define valores e códigos onde será solicitado cada valor.

ADIÇÃO - EXEMPLO 01

```
a = 2 #variável que recebe o primeiro valor.
b = 3 #variável que recebe o segundo valor.
soma = a+b #variável responsável pela soma dos valores de a e b.
print("O valor da soma é: ", soma) #apresentação do resultado da soma.
```

ADIÇÃO - EXEMPLO 02

```
a = float(input("Informe o primeiro valor da soma: ")) #variável que recebe valores inteiros ou com casas decimais.
b = float(input("Informe o segundo valor da soma: ")) #variável que recebe valores inteiros ou com casas decimais.
c = a + b #variável responsável pela soma dos valores de a e b.
print("O valor da soma é: ", c) #apresentação do resultado da soma.
```

No primeiro exemplo os valores são pré-definidos, o aluno informa os valores desejados, responde a soma no caderno e depois executa o código para verificação do resultado. No segundo exemplo, o aluno deve informar um valor assim que o programa for executado, responder este valor no caderno e depois comparar com a resposta da execução do código.

```
BÔNUS
print("Qual o resultado da adição de 27+45?") #apresentação da pergunta.
print("\n") #pular linha.

resp_esperada = 72 #variável contendo a resposta esperada
resp_usuario = " " #variável responsável por armazenar a resposta do aluno.

while resp_usuario != resp_esperada: #condição responsável por verificar a resposta do aluno com a resposta esperada.
    resp_usuario = float(input("Informe o valor: ")) #variável que recebe a resposta  do aluno em números inteiros ou com casas decimais.
    if resp_usuario == resp_esperada: #condição responsável por verificar se o aluno informou a resposta desejada.
print("Parabéns, você acertou!!!") #apresentação da resposta correta.
    else: #condição contrária a resposta esperada.
print("Tente novamente!!!") #apresentação que informa que o usuário deve tentar novamente.
```

BÔNUS

Neste exemplo é preciso que os alunos já possuam maior familiaridade com a linguagem Python, por trabalhar com condições de repetição e decisão, while e if respectivamente. Portanto, um valor pré-definido deve ser estabelecido, este valor vai ser comparado com a resposta do aluno ao executar o programa, caso esteja correto, o programa é finalizado, caso contrário, um novo valor deverá ser informado até chegar ao resultado correto.

Comentários do autor:

Os demais códigos apenas vão mudar os valores e sinais, adição (+); subtração (-); multiplicação (*); divisão (/). A multiplicação na programação é representada pelo asterisco.

```
SUBTRAÇÃO - EXEMPLO 01

a = 138 #variável que recebe o primeiro valor.
b = 53 #variável que recebe o segundo valor.
sub = a-b #variável responsável pela subtração dos valores de a e b.
print("O valor da subtração é: ", sub) #apresentação do resultado da subtração.
```

SUBTRAÇÃO - EXEMPLO 02

```
a = float(input("Informe o primeiro valor da subtração: ")) #variável que recebe valores inteiros ou com casas decimais.
b = float(input("Informe o segundo valor da subtração: ")) #variável que recebe valores inteiros ou com casas decimais.
c = a – b #variável responsável pela subtração dos valores de a e b.
print("O valor da subtração é: ", c) #apresentação do resultado da subtração.
```

BÔNUS

```
print("Qual o resultado da subtração de 85-45?") #apresentação da pergunta.
print("\n") #pular linha.

resp_esperada = 40 #variável contendo a resposta esperada
resp_usuario = " " #variável responsável por armazenar a resposta do aluno.

while resp_usuario != resp_esperada: #condição responsável por verificar a resposta do aluno com a resposta esperada.
    resp_usuario = float(input("Informe o valor: "))#variável que recebe a resposta do aluno em números inteiros ou com casas decimais.
    if resp_usuario == resp_esperada: #condição responsável por verificar se o aluno informou a resposta desejada.
print("Parabéns, você acertou!!!") #apresentação da resposta correta.
    else: #condição contrária a resposta esperada.
print("Tente novamente!!!") #apresentação que informa que o usuário deve tentar novamente.
```

MULTIPLICAÇÃO - EXEMPLO 01

```
a = 25 #variável que recebe o primeiro valor.
b = 25 #variável que recebe o segundo valor.
mult = a*b #variável responsável pela multiplicação dos valores de a e b.
print("O valor da multiplicação é: ", mult) #apresentação do resultado da multiplicação.
```

MULTIPLICAÇÃO - EXEMPLO 02

```
a = float(input("Informe o primeiro valor da multiplicação: "))#variável que recebe valores inteiros ou com casas decimais.
b = float(input("Informe o segundo valor da multiplicação: "))#variável que recebe valores inteiros ou com casas decimais.
c = a * b #variável responsável pela multiplicação dos valores de a e b.
print("O valor da multiplicação é: ", c) #apresentação do resultado da multiplicação.
```

BÔNUS

```
print("Qual o resultado da multiplicação de 100x25?") #apresentação da pergunta.
print("\n")#pular linha.

resp_esperada = 2500 #variável contendo a resposta esperada
resp_usuario = " " #variável responsável por armazenar a resposta do aluno.


while resp_usuario != resp_esperada: #condição responsável por verificar a resposta do aluno com a resposta esperada.
    resp_usuario = float(input("Informe o valor: "))#variável que recebe a resposta  do aluno em números inteiros ou com casas decimais.
    if resp_usuario == resp_esperada: #condição responsável por verificar se o aluno informou a resposta desejada.
print("Parabéns, você acertou!!!") #apresentação da resposta correta.
    else: #condição contrária a resposta esperada.
print("Tente novamente!!!") #apresentação que informa que o usuário deve tentar novamente.
```

DIVISÃO - EXEMPLO 01

```
a = 25 #variável que recebe o primeiro valor.
b = 25 #variável que recebe o segundo valor.
div = a/b #variável responsável pela divisão dos valores de a e b.
print("O valor da divisão é: ", div) #apresentação do resultado da multiplicação.
```

DIVISÃO - EXEMPLO 02

```
a = float(input("Informe o primeiro valor da divisão: ")) #variável que recebe valores inteiros ou com casas decimais.
b = float(input("Informe o segundo valor da divisão: ")) #variável que recebe valores inteiros ou com casas decimais.
div = a / b #variável responsável pela divisão dos valores de a e b.
print("O valor da divisão é: ", c) #apresentação do resultado da divisão.
```

BÔNUS

```
print("Qual o resultado da divisão de 60/3?") #apresentação da pergunta.
print("\n")#pular linha.

resp_esperada = 20 #variável contendo a resposta esperada
resp_usuario = " " #variável responsável por armazenar a resposta do aluno.

while resp_usuario != resp_esperada: #condição responsável por verificar a resposta do aluno com a resposta esperada.
    resp_usuario = float(input("Informe o valor: "))#variável que recebe a resposta do aluno em números inteiros ou com casas decimais.
    if resp_usuario == resp_esperada: #condição responsável por verificar se o aluno informou a resposta desejada.
print("Parabéns, você acertou!!!") #apresentação da resposta correta.
    else: #condição contrária a resposta esperada.
print("Tente novamente!!!") #apresentação que informa que o usuário deve tentar novamente.
```

SEQUÊNCIA NUMÉRICA

OBJETIVOS

Exemplificar sequências numéricas diretas e com intervalos.

CONTEÚDO

Sequência numérica simples e com intervalos; sequências de valores pares e ímpares; listas.

PROCEDIMENTO METODOLÓGICO

É preciso que o professor de laboratório leve para o quadro exemplos de cada sequência numérica, sejam elas diretas, com intervalos, números pares ou ímpares, sempre ocorrendo o incentivo da participação de cada aluno, entre outras, sendo resolvido estes exemplos no quadro e passando posteriormente algumas atividades em caderno. Por meio dos alunos, as atividades devem ser respondidas e corrigidas pelo professor de laboratório junto a estes. Posteriormente, deve-se levar a IDLE da linguagem Python as atividades já feitas e posteriormente novos desafios em forma de atividades, incentivando e ajudando cada aluno.

SUGESTÃO

É importante que o professor explique que o computador identifica o número zero como o primeiro valor a ser apresentado, desta forma, em uma sequência de dez números, os valores apresentados vão ser: [0,1,2,3,4,5,6,7,8,9], totalizando dez valores.

1ª casa	2ª casa	3ª casa	4ª casa	5ª casa	6ª casa	7ª casa	8ª casa	9ª casa	10ª casa
0	1	2	3	4	5	6	7	8	9

Tabela 1: Casas em list range
Fonte: Autoria própria

EXEMPLO SEQUÊNCIA DE VALORES DIRETOS

```
list(range(5)) #listar uma sequência de cinco valores
[0, 1, 2, 3, 4]
```

Comentários do autor:

Conforme demonstrado no exemplo acima, os valores apresentados vão do primeiro número identificado pelo computador (zero), dando sequência com os próximos valores. Para que o valor final seja exatamente cinco, o valor pedido no range deverá ser um numero a mais do solicitado, neste caso, seis.

```
list(range(1,11)) #listar uma sequência de onze valores com início no número um
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Comentários do autor:

Para que o valor zero não seja apresentado, é preciso informar que o primeiro valor da lista será o número um, conforme apresentado no exemplo acima. Isto vale para situações onde o primeiro valor apresentado deverá ser o número um.

EXEMPLO SEQUÊNCIA DE VALORES PARES

```
list(range(2,20,2)) #listar uma sequência de valores com intervalos de dois
[2, 4, 6, 8, 10, 12, 14, 16, 18]
```

Comentários do autor:

O exemplo acima demonstra uma sequência numérica onde o primeiro valor a ser apresentado é o número dois, sendo seu último valor o número vinte, com um intervalo de duas casas.

EXEMPLO SEQUÊNCIA TABUADA DE NOVE

```
list(range(0,90,9)) #listar uma sequência de valores com intervalos de nove
[0, 9, 18, 27, 36, 45, 54, 63, 72, 81]
```

Comentários do autor:

Por meio dos comandos list e range é possível exemplificar diversas funções. No exemplo acima, foi possível apresentar a tabuada de nove, com valores apresentados entre zero e oitenta e um (9x9), visto que, o primeiro valor reconhecido pelo computador sempre será o zero.

EXEMPLO SEQUÊNCIA DE VALORES NEGATIVOS

```
list(range(0,-10,-1)) #listar uma sequência de valores negativos.
[0, -1, -2, -3, -4, -5, -6, -7, -8, -9]
```

Comentários do autor:

Para que o resultado apresente valores negativos, basta informar o sinal de negativo (-) antes de cada valor, como o zero é neutro, ele se apresentará de forma diferente dos demais.

BÔNUS

```
for item in range(10): #usar condição for para listar dez valores
    print(item) #apresentar valores
[0,1,2,3,4,5,6,7,8,9]
```

DICA BÔNUS

Uma forma diferente de apresentar os valores são por meio do laço de repetição for, neste caso, é pedido que o laço de petição verifique dez valores (range(10)) e posteriormente os apresentem por meio da variável item. Sendo uma forma adicional de elaboração de sequência numérica.

EXPRESSÃO NUMÉRICA

OBJETIVOS

Dividir a expressão numérica em partes, visando sua melhor compreensão

CONTEÚDO

Expressão numérica.

PROCEDIMENTO METODOLÓGICO

É preciso que o professor de laboratório leve para o quadro exemplos de expressões numéricas contendo os operadores matemáticos, sendo resolvido estes exemplos no quadro e passando posteriormente algumas atividades em caderno. Por meio dos alunos, as atividades devem ser respondidas e corrigidas pelo professor de laboratório junto a estes. Posteriormente, deve-se levar a IDLE da linguagem Python as atividades já feitas e posteriormente novos desafios em forma de atividades, incentivando e ajudando cada aluno.

SUGESTÃO

Separe cada momento da resolução da expressão. Ao iniciar pelos parentes, mostre aos alunos o resultado e posteriormente leve ao desenvolvimento, continue somente ao finalizar este momento.

```
import time #biblioteca utilizada para definir tempo das mensagens.

print("Analise a seguinte expressão 90+{85-[28+(10-2)]}") #apresentação da expressão numérica.
print("\n") #pular linha.

exp1 = " " #variável responsável por armazenar a resposta do aluno.
resp_esp1 = "parenteses" #variável contendo a resposta esperada
while exp1 != resp_esp1: #condição responsável por verificar a resposta do aluno com a resposta esperada.
   exp1 = str(input("Qual o primeiro sinal a ser resolvido? ")) #variável utilizada para perguntar e armazenar a resposta do aluno.
   print("\n") #pular linha.

  if exp1 == resp_esp1: #condição responsável por verificar se o aluno informou a resposta desejada.
print("Você acertou a primeira expressão") #mensagem ao usuário.
print("Com isso a subtração (10-2) é: 8") #mensagem ao usuário.
print("\n") #pular linha.
time.sleep(1.5) #delay para mostrar as próximas mensagens.
   else: #condição contrária a resposta esperada.
print("Sua resposta não está correta, tente novamente") #apresentação que informa que o usuário deve tentar novamente.
print("\n") #pular linha.
```

```
print("Nossa expressão agora é 90+{85-[28+8]}") #apresentação da expressão reduzida.
print("\n") #pular linha.

exp2 = " " #variável responsável por armazenar a resposta do aluno.
resp_esp2 = "colchetes" #variável contendo a resposta esperada
while exp2 != resp_esp2: #condição responsável por verificar a resposta do aluno com a resposta esperada.
    exp2 = str(input("Qual o segundo sinal a ser resolvido? ")) #variável utilizada para perguntar e armazenar a resposta do aluno.
    print("\n") #pular linha.
    if exp2 == resp_esp2: #condição responsável por verificar se o aluno informou a resposta desejada.
        print("Você acertou a segunda expressão") #mensagem ao usuário.
        print("Com isso a subtração [28+8] é: 36") #mensagem ao usuário.
        print("\n") #pular linha.
        time.sleep(1.5) #delay para mostrar as próximas mensagens.
    else: #condição contrária a resposta esperada.
        print("Sua resposta não está correta, tente novamente") #apresentação que informa que o usuário deve tentar novamente.
        print("\n") #pular linha

print("Nossa expressão agora é 90+{85-36}") #apresentação da expressão reduzida.
print("\n") #pular linha.

exp3 = " " #variável responsável por armazenar a resposta do aluno.
resp_esp3 = "chaves"  #variável contendo a resposta esperada
while exp3 != resp_esp3: #condição responsável por verificar a resposta do aluno com a resposta esperada.
    exp3 = str(input("Qual o terceiro sinal a ser resolvido? ")) #variável utilizada para perguntar e armazenar a resposta do aluno.
    print("\n") #pular linha.
    if exp3 == resp_esp3: #condição responsável por verificar se o aluno informou a resposta desejada.
        print("Você acertou a terceira expressão") #mensagem ao usuário.
        print("Com isso a subtração {85-36} é: 49") #mensagem ao usuário.
        print("\n") #pular linha.
        time.sleep(1.5) #delay para mostrar as próximas mensagens.
    else: #condição contrária a resposta esperada.
        print("Sua resposta não está correta, tente novamente") #apresentação que informa que o usuário deve tentar novamente.
        print("\n") #pular linha.
```

```
print("Nossa expressão agora é 90+49") #apresentação da expressão reduzida.
print("\n") #pular linha

resultado = " " #variável responsável por armazenar a resposta do aluno.
resp_esp4 = 139 #variável contendo a resposta esperada
while resultado != resp_esp4: #condição responsável por verificar a resposta do aluno com a resposta esperada.
    resultado = int(input("Qual o nosso resultado final? ")) #variável utilizada para perguntar e armazenar a resposta do aluno.
    print("\n") #pular linha.
    if resultado == resp_esp4: #condição responsável por verificar se o aluno informou a resposta desejada.
        print("Você acertou o resultado final") #mensagem ao usuário
        print("\n") #pular linha.
        time.sleep(1.5) #delay para mostrar as próximas mensagens.
    else: #condição contrária a resposta esperada.
        print("Sua resposta não está correta, tente novamente") #apresentação que informa que o usuário deve tentar novamente.
        print("\n") #pular linha.

print("Nossa expressão está completa") #mensagem ao usuário
print("\n") #pular linha.
```

Comentários do autor:

Ao professor de laboratório, este conteúdo se torna o mais completo e complexo para os alunos, pois diversas funções da programação são utilizadas, portanto, é importante o constante diálogo com os alunos, e detalhada explicação. Ao finalizar a construção da primeira parte a ser resolvida, neste caso os parênteses, se torna importante que o professor incentive seus alunos para que estes desenvolvam o restante do código, sugerindo o procedimento e os motivando para que o resultado final seja alcançado, pois basicamente o código se repete, apenas trocando o nome das variáveis e valores. Ao finalizar, estes alunos se sentem motivados e os princípios da aprendizagem significativa e construcionismo são utilizadas mais uma vez.

CONTEÚDO BÔNUS

OBJETIVOS

Afim de habituar os alunos a linguagem Python, se torna interessante a utilização de exercícios simples, como o antecessor e o sucessor de um valor informado, os habituando e demonstrado como o compilador processa as informações.

CONTEÚDO

Antecessor e sucessor.

PROCEDIMENTO METODOLÓGICO

Utilização de atividades de fixação, posteriormente a correção e o desenvolvimento dos códigos por meio da linguagem Python.

```
valor = int(input("Entre com um valor: ")) #variável que armazenará o valor informado pelo aluno
print("O valor informado foi: ",valor) #apresentação do valor informado
print("O seu antecessor é: ",valor-1) #apresentação do antecessor
print("O seu sucessor é: ",valor+1) #apresentação do sucessor
```

Comentários do autor:

Este exemplo pode ser utilizado com diversos valores para seus antecessores e sucessores, sendo preciso apenas a troca do valor referente as casas, por exemplo, para o sucessor da variável "valor" de duas casas maiores que o informado, basta apenas escrever o seguinte código: valor+2, desta forma, o sucessor do valor um será três.

EXEMPLOS DE ATIVIDADES

ANTECESSOR E SUCESSOR

1) Elaborar um código que exibirá o valor informado, antecessor e sucessor.

2) Elaborar um código que exibirá o valor informado, 3 valores do antecessor e 5 valores do sucessor.

SEQUÊNCIA NUMÉRICA

1) Elaborar uma sequência numérica que vá do valor 0 ao valor 50.

2) Elaborar uma sequência numérica que vá do valor 20 ao valor 25.

3) Elaborar uma sequência numérica que vá do valor -1 ao valor -25.

4) Elaborar uma sequência numérica que vá do valor -1 ao -15.

5) Elaborar uma sequência numérica que vá do valor 0 ao valor 30 de 3 em 3.

6) Elaborar uma sequência numérica de números pares.

7) Elaborar uma sequência numérica de números impares.

OPERAÇÕES MATEMÁTICAS

1) Desenvolva códigos de adição.

2) Desenvolva códigos de subtração.

3) Desenvolva códigos de multiplicação.

4) Desenvolva códigos de divisão.

OPERAÇÕES MATEMÁTICAS

1) Desenvolva códigos envolvendo os operações matemáticas.

E AGORA?

A proposta de se utilizar a programação de computadores por meio da linguagem Python vem de encontro com a necessidade de se criar técnicas e meios tecnológicos que incentive e estimule o aprendizado de cada aluno, tornando desafiante e possível de se aprender.

As atividades propostas neste caderno vão de encontro com os conteúdos da disciplina de matemática do 6º ano do ensino fundamental II, e visam contribuir com os professores e estudantes envolvidos, sendo permitido seu uso, baseado em adaptação, nas demais disciplinas e séries.

Se torna essencial a utilização de outras ferramentas de aprendizagem em programação, como Scratch e Python, para complementar o processo de aprendizagem, visto que, o desenvolvimento de exemplos geométricos se torna mais simples de se desenvolver nestas ferramentas.

Por fim, é estendido o convite para a leitura da dissertação vinculada a este caderno pedagógico, intitulada: "O ensino de matemática por meio da linguagem de programação Python", para a compreensão mais ampla desta proposta.

REFERÊNCIAS

AUSUBEL, D.P. **Aquisição e retenção de conhecimentos: Uma perspectiva cognitiva.** New York, 2000.

Ausubel, D.P. & Fitzgerald, D. **The role of discriminability in meaningful verbal learning and retention.** Journal of Educational Psychology, 52(5); 266-74, 1968.

MACHADO, Flávia Cristina; LIMA, Maria de Fátima Webber Prado. **O uso da tecnologia educacional: Um fazer pedagógico no cotidiano escola.** Caxias do Sul, 2017.

PAPERT, Seymour. **A máquina das crianças: Repensando a escola na era da informática.** Nova York, 2008.

PAPERT, Seymou. **LOGO: Computadores e educação.** São Paulo, 1985.

PRIMO, Alex Fernando Teixeira. **INTERAÇÃO MEDIADA POR COMPUTADOR: a comunicação e a educação a distância segundo uma perspectiva sistêmico-relacional.** Porto Alegre, 2003.