

Semana de Nivelamento em LP





Lucas Migliorin

Instrutor

5º período de Engenharia de
Computação

Instagram: @lucas.migliorin

Email: lmdr.eng19@uea.edu.br



Leticia Magalhães

Monitora

7º período de Engenharia de
Computação

Instagram: @ledizia_

Email: Imm.eng18@uea.edu.br

Instruções para o decorrer do WorkShop

Perguntas durante a apresentação:

- ⬡ Escreva no chat sua dúvida que a monitora estará respondendo.
- ⬡ Caso a dúvida seja mais complexa de responder por texto, será dado uma pausa e então respondida pelo instrutor.
- ⬡ Mantenham os microfones e câmeras desligadas
- ⬡ Podem nos procurar para tirar dúvidas

Material do Workshop disponível online:

- ⬡ Link: <https://github.com/Migliorin/Semana-LP>

Siga as redes sociais do DAETEC e CAECOMP!



@daetec_uea



@caecomp.uea

AGENDA DO DIA

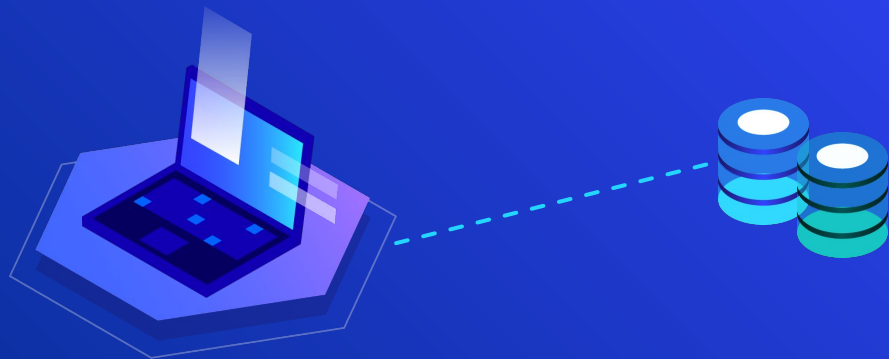
Assunto complementar

Funções

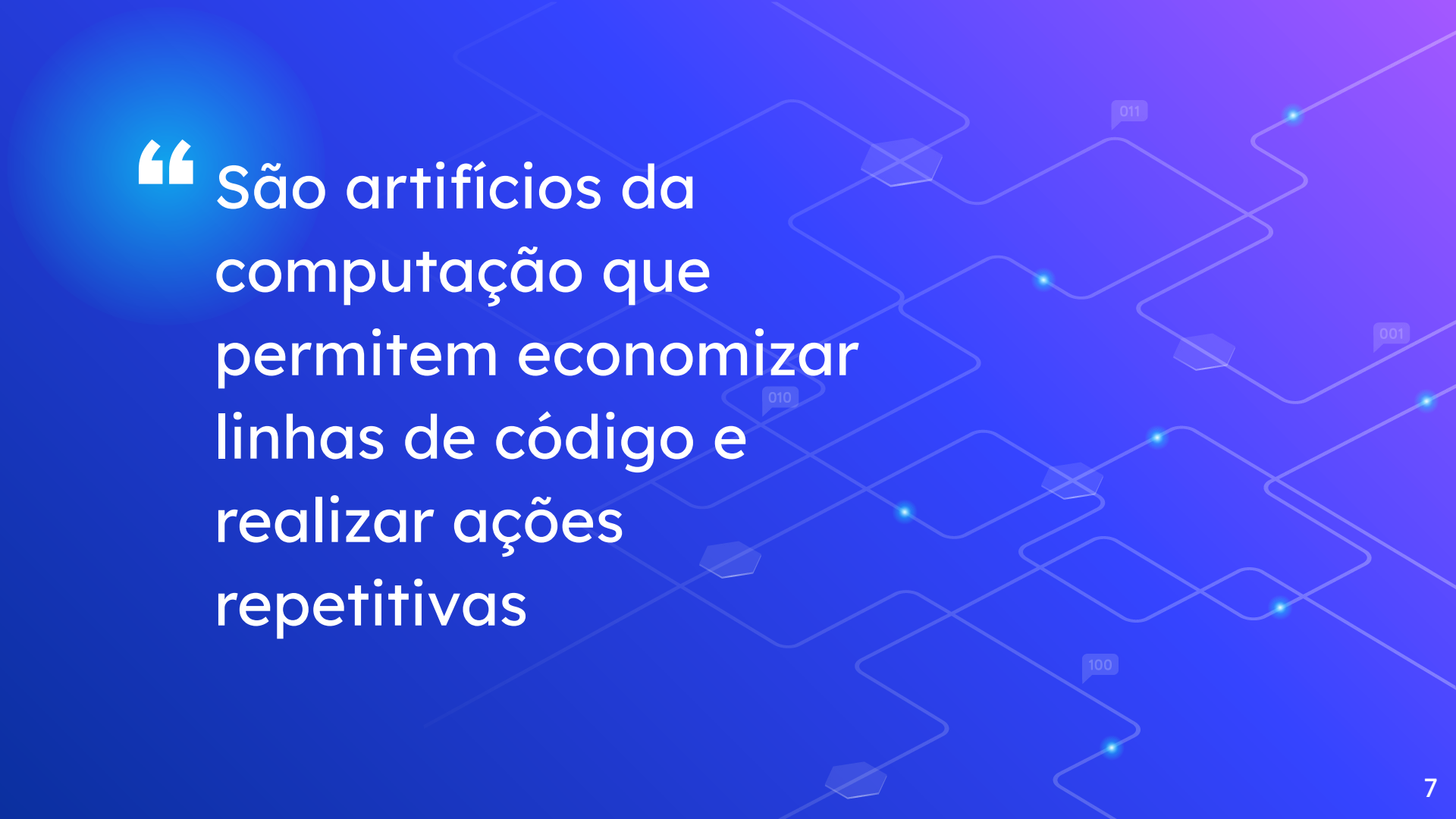
Estudando Python



1. Funções em Python



```
for (var i = 0; i < response.length; i++)  
    var layer = L.marker(  
        [response[i].latitude, response[i].  
        // , {icon: myIcon}  
    );  
    layer.addTo(group);  
  
    layer.bindPopup(  
        "<p>" + "Species: " + response[i].species + "</p>" +  
        "<p>" + "Description: " + response[i].description + "</p>" +  
        "<p>" + "Seen at: " + response[i].latitude + "</p>" +  
        "<p>" + "On: " + response[i].sighted_at + "</p>" +  
    );  
    $('#select').change(function() {  
        species = this.value;  
    });  
    });  
    });  
    $.ajax({  
        url: queryURL,  
        method: "GET"  
    }).done(function(response) {  
        for (var i = 0; i < response.length; i++) {  
            var layer = L.marker(  
                [response[i].latitude, response[i].  
                // , {icon: myIcon}  
            );  
            layer.addTo(group);
```



“ São artifícios da
computação que
permitem economizar
linhas de código e
realizar ações
repetitivas

Somando valores



Somando valores

Sem funções faríamos da seguinte forma



Somando valores

Sem funções faríamos da seguinte forma

Exemplo

```
valor_um = 10
valor_dois = 33

resultado = valor_um + valor_dois

print("A soma e:", resultado)
```

Somando valores

Com função teríamos o seguinte



Somando valores

Com função teríamos o seguinte

⬡ Exemplo



Somando valores

Com função teríamos o seguinte

Exemplo

```
def soma_valores(valor_um,valor_dois):  
    resultado = valor_um + valor_dois  
    return resultado
```



Somando valores

Com função teríamos o seguinte

Exemplo

```
def soma_valores(valor_um, valor_dois):  
    resultado = valor_um + valor_dois  
    return resultado
```

↑
Prefixo
obrigatório para
definir uma
função



Somando valores

Com função teríamos o seguinte

Exemplo

```
def soma_valores(valor_um,valor_dois):  
    resultado = valor_um + valor_dois  
    return resultado
```

Chamando a função



Somando valores

Com função teríamos o seguinte

Exemplo

```
def soma_valores(valor_um,valor_dois):  
    resultado = valor_um + valor_dois  
    return resultado
```

Chamando a função

Exemplo

```
def soma_valores(valor_um,valor_dois):  
    resultado = valor_um + valor_dois  
    return resultado  
  
print(soma_valores(10,5))
```



Fórmula de Bhaskara

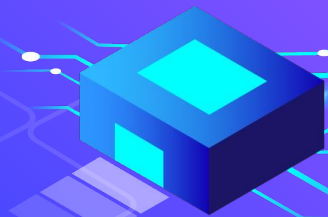


Fórmula de Bhaskara

Com função teríamos o seguinte

Lembrete

$$x = \frac{-b \pm \sqrt{b^2 - 4.a.c}}{2.a}$$



Fórmula de Bhaskara

Com função teríamos o seguinte

Exemplo

```
def bhaskara(a,b,c):  
    discriminante = (b**2) - (4*a*c)  
  
    if(discriminante < 0):  
        print("Nao ha raizes reais")  
        return False  
  
    elif(discriminante == 0):  
        resultado = (-b)/(2*a)  
        return resultado  
  
    else:  
        x1 = (-b) + (discriminante**(1/2))/(2*a)  
        x2 = (-b) - (discriminante**(1/2))/(2*a)  
        return [x1,x2]  
  
# 4x2 + x + 1  
print(bhaskara(4,1,1))
```

Lembrete

$$x = \frac{-b \pm \sqrt{b^2 - 4.a.c}}{2.a}$$

Fórmula de Bhaskara

Com função teríamos o seguinte

Exemplo

```
def bhaskara(a,b,c):  
    discriminante = (b**2) - (4*a*c)  
  
    if(discriminante < 0):  
        print("Nao ha raizes reais")  
        return False  
  
    elif(discriminante == 0):  
        resultado = (-b)/(2*a)  
        return resultado  
  
    else:  
        x1 = (-b) + (discriminante**(1/2))/(2*a)  
        x2 = (-b) - (discriminante**(1/2))/(2*a)  
        return [x1,x2]  
  
# 4x² + x + 1  
print(bhaskara(4,1,1))
```

Tratamento
para raiz
negativa

Lembrete

$$x = \frac{-b \pm \sqrt{b^2 - 4.a.c}}{2.a}$$

Fórmula de Bhaskara

Com função teríamos o seguinte

Exemplo

```
def bhaskara(a,b,c):  
    discriminante = (b**2) - (4*a*c)  
  
    if(discriminante < 0):  
        print("Nao ha raizes reais")  
        return False  
  
    elif(discriminante == 0):  
        resultado = (-b)/(2*a)  
        return resultado  
  
    else:  
        x1 = (-b) + (discriminante**(1/2))/(2*a)  
        x2 = (-b) - (discriminante**(1/2))/(2*a)  
        return [x1,x2]  
  
# 4x2 + x + 1  
print(bhaskara(4,1,1))
```

Tratamento
para raiz igual a
zero


Lembrete

$$x = \frac{-b \pm \sqrt{b^2 - 4.a.c}}{2.a}$$

Fórmula de Bhaskara

Com função teríamos o seguinte

Exemplo

```
def bhaskara(a,b,c):  
    discriminante = (b**2) - (4*a*c)  
  
    if(discriminante < 0):  
        print("Nao ha raizes reais")  
        return False  
  
    elif(discriminante == 0):  
        resultado = (-b)/(2*a)  
        return resultado  
  
    else:   
        x1 = (-b) + (discriminante**(1/2))/(2*a)  
        x2 = (-b) - (discriminante**(1/2))/(2*a)  
        return [x1,x2]  
  
# 4x² + x + 1  
print(bhaskara(4,1,1))
```

Lembrete

$$x = \frac{-b \pm \sqrt{b^2 - 4.a.c}}{2.a}$$

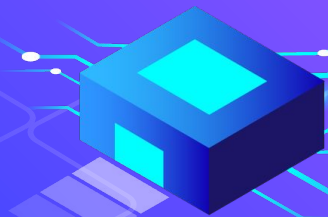
Se não calcular
as raízes
normalmente

Fórmula de Bhaskara

Com outra função

Lembrete

$$x = \frac{-b \pm \sqrt{b^2 - 4.a.c}}{2.a}$$



Fórmula de Bhaskara

Com outra função

Exemplo

```
def bhaskara(a,b,c):  
    discriminante = (b**2) - (4*a*c)  
  
    if(discriminante < 0):  
        return False  
    else:  
        x1 = (-b) + (discriminante**(1/2))/(2*a)  
        x2 = (-b) - (discriminante**(1/2))/(2*a)  
        return [x1,x2]  
  
# 4x2 + x + 1  
  
resul = bhaskara(4,1,1)  
  
if(resultado == False):  
    print("Nao ha raizes reais")  
else:  
    print('Nossas raizes são: ' + str(resul[0]) + ' e ' + str(resul[1]))
```

Lembrete

$$x = \frac{-b \pm \sqrt{b^2 - 4.a.c}}{2.a}$$

Vamos fazer juntos o seguinte programa utilizando funções

1. Escreva um programa em Python para calcular se você foi ou não aprovado na disciplina de Introdução à Linguagem de Programação supondo que foram feitas 3 provas e que a média para passar sem final é 8, se não passou calcule quanto precisa tirar na final para passar sabendo que a média é calculada da seguinte forma:

$$\text{Media_final} = (\text{nota_prova_final} - (2 \times \text{media_nomrila}))/3$$





2. Conteúdo extra: lista, dicionário e string



```
for (var i = 0; i < response.length; i++)  
    var layer = L.marker(  
        [response[i].latitude, response[i].longitude],  
        {icon: myIcon}  
    );  
    layer.addTo(group);  
  
    layer.bindPopup(  
        "<p>" + "Species: " + response[i].species + "</p>" +  
        "<p>" + "Description: " + response[i].description + "</p>" +  
        "<p>" + "Seen at: " + response[i].latitude + " " + response[i].longitude + "</p>" +  
        "<p>" + "On: " + response[i].sighted_at + "</p>"  
    );  
  
$('#select').change(function() {  
    species = this.value;  
});  
  
$.ajax({  
    url: queryURL,  
    method: "GET"  
}).done(function(response) {  
    for (var i = 0; i < response.length; i++) {  
        var layer = L.marker(  
            [response[i].latitude, response[i].longitude],  
            {icon: myIcon}  
        );  
        layer.addTo(group);  
    }  
});
```

Listas e dicionários

Conseguimos tanto adicionar

- ⬡ Uma lista dentro de uma lista
- ⬡ Um dicionário dentro de um dicionário
- ⬡ Uma lista dentro de um dicionário
- ⬡ Um dicionário dentro de uma lista



Listas e dicionários

Conseguimos tanto adicionar

- Uma lista dentro de uma lista



Listas e dicionários

Conseguimos tanto adicionar

- Uma lista dentro de uma lista
 - Exemplo

```
nossa_lista = ['Lucas',20,1.75,False]
minha_notas = [10,6,9]
nossa_lista.append(minha_notas)
# ['Lucas',20,1.75,False,[10,6,9]]
```

Nosso novo
formato de lista

Listas e dicionários

Conseguimos tanto adicionar

- Um dicionário dentro de um dicionário



Listas e dicionários

Conseguimos tanto adicionar

- Um dicionário dentro de uma dicionário
- Exemplo

```
dicionario = {'Nome' : 'Lucas Migliorin',  
             'Idade' : 20,  
             'Altura' : 1.75,  
             'Estuda' : False}  
  
dicionario['Minhas Notas'] = {'av1' : 10, 'av2' : 6, 'av3' : 9}  
  
# {'Nome' : 'Lucas Migliorin',  
#   'Idade' : 20,  
#   'Altura' : 1.75,  
#   'Estuda' : False,  
#   'Minhas Notas' : {'av1' : 10,  
#                     'av2' : 6,  
#                     'av3' : 9}}
```

Formato do
novo dicionário

Listas e dicionários

Conseguimos tanto adicionar

- Uma lista dentro de um dicionário



Listas e dicionários

Conseguimos tanto adicionar

- Uma lista dentro de um dicionário
- Exemplo

```
dicionario = {'Nome' : 'Lucas Migliorin',  
              'Idade' : 20,  
              'Altura' : 1.75,  
              'Estuda' : False}
```

```
dicionario['Minhas Notas'] = [10, 6, 9]
```

```
# {'Nome' : 'Lucas Migliorin',  
#   'Idade' : 20,  
#   'Altura' : 1.75,  
#   'Estuda' : False,  
#   'Minhas Notas' : [10, 6, 9] }
```

Formato do
novo dicionário

Listas e dicionários

Conseguimos tanto adicionar

- Um dicionário dentro de uma lista



Listas e dicionários

Conseguimos tanto adicionar

- Um dicionário dentro de uma lista
 - Exemplo

```
lista = ['Lucas',20,1.75,False]

lista.append({'av1' : 10, 'av2' : 6, 'av3' : 9})

# ['Lucas',20,1.75,False, {'av1' : 10,
#                           'av2' : 6,
#                           'av3' : 9}]
```

Formato da nova
lista

Manipulando strings

Conseguimos pegar valores específicos ou até mesmo uma certa sequência

◡ Exemplo



Manipulando strings

Conseguimos pegar valores específicos ou até mesmo uma certa sequência

Exemplo

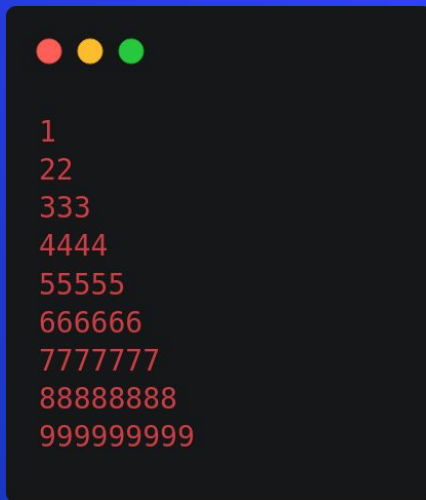
```
frase = 'Obrigado por comparecerem ao WorkShop'

print(frase[0]) # O
print(frase[1]) # b
print(frase[2]) # r
print(frase[3]) # i
print(frase[4]) # g
print(frase[5]) # a
print(frase[6]) # d
print(frase[7]) # o

print(frase[0:7]) # Obrigado
```

Vamos fazer juntos o seguinte programa

1. Escreva um programa em Python que consiga fazer o que é mostrado na figura abaixo com qualquer número fornecido pelo usuário:





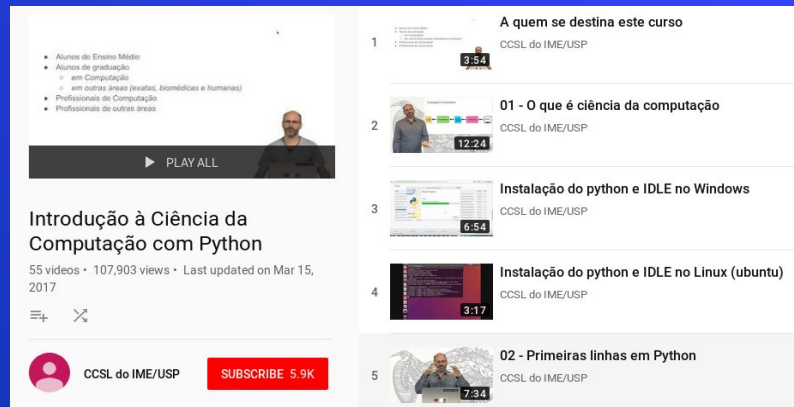
3. Outras formas de estudar Python



```
for (var i = 0; i < response.length; i++)  
    var layer = L.marker(  
        [response[i].latitude, response[i].  
        // , {icon: myIcon}  
    );  
    layer.addTo(group);  
  
    layer.bindPopup(  
        "<p>" + "Species: " + response[i].species + "</p>" +  
        "<p>" + "Description: " + response[i].description + "</p>" +  
        "<p>" + "Seen at: " + response[i].latitude + "</p>" +  
        "<p>" + "On: " + response[i].sighted_at + "</p>" +  
    );  
    $('#select').change(function() {  
        species = this.value;  
    });  
    $.ajax({  
        url: queryURL,  
        method: "GET"  
    }).done(function(response) {  
        for (var i = 0; i < response.length; i++) {  
            var layer = L.marker(  
                [response[i].latitude, response[i].  
                // , {icon: myIcon}  
            );  
            layer.addTo(group);
```

Links úteis

No Youtube temos



A quem se destina este curso
CCSL do IME/USP

1 3:54

2 12:24

3 6:54

4 3:17

5 7:34

01 - O que é ciência da computação
CCSL do IME/USP

Instalação do python e IDLE no Windows
CCSL do IME/USP

Instalação do python e IDLE no Linux (ubuntu)
CCSL do IME/USP

02 - Primeiras linhas em Python
CCSL do IME/USP

Introdução à Ciência da Computação com Python

55 videos • 107,903 views • Last updated on Mar 15, 2017

CCSL do IME/USP

SUBSCRIBE 5.9K

PYCUBATOR

Olé e bem-vindo ao curso de Python para iniciantes. Aonde deseja começar?

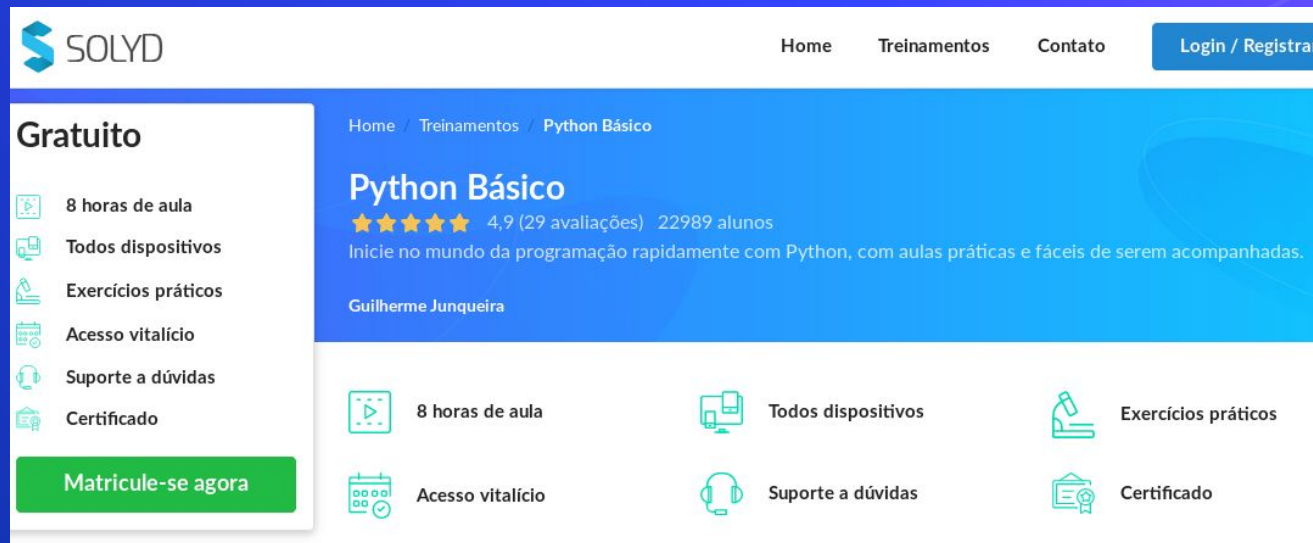
1. **Introdução**
2. Variáveis e Tipos de dados simples
3. Estruturas de controle
4. Tipo de dados compostos
5. Arquivos e Módulos
6. Orientação a Objetos
7. Exceções
8. Funções e Programação Funcional
9. A Biblioteca Padrão
10. Testes unitários

Criado por Noam Efranbaum (@noameli) e Udi Oron (@nonZero).

Traduzido Por Gilson Filho (@gilsonfilho)

Links úteis

Curso com certificado



SOLYD Home Treinamentos Contato Login / Registrar

Gratuito

- 8 horas de aula
- Todos dispositivos
- Exercícios práticos
- Acesso vitalício
- Suporte a dúvidas
- Certificado

Matricule-se agora

Home / Treinamentos / Python Básico

Python Básico

★★★★★ 4,9 (29 avaliações) 22989 alunos

Inicie no mundo da programação rapidamente com Python, com aulas práticas e fáceis de serem acompanhadas.

Guilherme Junqueira

- 8 horas de aula
- Todos dispositivos
- Exercícios práticos
- Acesso vitalício
- Suporte a dúvidas
- Certificado